The Dissertation Committee for David Iseri Inouye
certifies that this is the approved version of the following dissertation:

Appropriate, Accessible and Appealing Probabilistic Graphical Models

Committee:

Inderjit S. Dhillon, Supervisor

Pradeep Ravikumar, Co-Supervisor

Raymond J. Mooney

Qixing Huang

Byron C. Wallace

Appropriate, Accessible and Appealing Probabilistic Graphical Models

by

David Iseri Inouye, B.A., B.S.E.E., M.S.C.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2017

Dedicated to my beloved wife Harriett for her persistent warmth.

Acknowledgments

Unless the Lord *builds the house,*
those who build it labor in vain.
Unless the Lord *watches over the city,*
the watchman stays awake in vain.
It is in vain that you rise up early
and go late to rest,
eating the bread of anxious toil;
for he gives to his beloved sleep.

— Psalm 127:1-2

I thank the Lord God for sustaining me, walking with me and researching with me throughout my work. If the Lord had not worked, my research efforts would have been in vain. Yet, the Lord in his good pleasure has graciously allowed me to complete my Ph.D. Any truth or good you find in this work are from the Lord and any inadequacies are my own.

I am grateful to my supervisor Prof. Inderjit Dhillon for his overall support throughout my research and for giving me great freedom to pursue topics that excited me. I also appreciated his reasonable expectations about the amount and pacing of research—especially when I felt overwhelmed. I thank my co-supervisor Prof. Pradeep Ravikumar for convincing me to submit my first graphical model paper to ICML—a task that initially felt impossible at the time. His constructive guidance and feedback helped form the direction of this research as well as my broader approach to research. I thank

Prof. Byron Wallace for our many fun and fruitful discussions; his enthusiasm and support greatly encouraged me. I thank the other committee members Prof. Raymond Mooney and Prof. Qixing Huang for their time and support.

I gratefully acknowledge the financial support of the National Science Foundation Graduate Research Fellowship Program.

I want to express my gratitude to Prof. Genevera Allen and Prof. Eunho Yang for their work on the Poisson review paper. I thank my former labmate Nagarajan Natarajan (Naga) for his smiles and encouraging words especially in my early years of graduate school. I also greatly appreciated Hsiang-Fu Yu's kind help with optimization, proofs, coding problems and LaTeX. I thank all my other labmates for talking with me and helping me along the way including but not limited to Prof. Cho-Rui Hsieh, Si Si, Kai Zhong, Rashish Tandon, Tianyang Li, Qi Lei, Kai-Yang Chiang and Prof. Joyce Whang.

I am grateful to Prof. Jugal Kalita who sparked my interest in both research and machine learning during my undergraduate studies. Without his influence, I likely would have never pursued research or even known about machine learning. I also thank Prof. William Pottenger who mentored my second summer research internship and supported my application to graduate school.

I am indebted to my parents for their continual love, support and prayers. I thank my brother for our many random research discussions. With the greatest love, I thank my beloved wife Harriett for all her love, affection, perseverance and fortitude amid the challenges of research including the long lost-in-thought stares and late nights.

I am grateful to my many Hampton House friends and roommates who kept me sane in the early years of graduate school. You offered a breath of fresh air away from the pressures of academic life.

Finally, I close with these quotes that encouraged me, challenged me and often

enlivened a right perspective throughout my research:

> *For the LORD gives wisdom;*
> *from his mouth come knowledge and understanding;*
>
> — Proverbs 2:6

> *Come now, you who say, "Today or tomorrow we will go into such and such a town and spend a year there and trade and make a profit" . . . . Instead you ought to say, "If the Lord wills, we will live and do this or that."*
>
> — James 4:13,15

> *If I have prophetic powers, and understand all mysteries and all knowledge, and if I have all faith, so as to remove mountains, but have not love, I am nothing.*
>
> — 1 Corinthians 13:2

> *"Come to me, all who labor and are heavy laden, and I will give you rest. Take my yoke upon you, and learn from me, for I am gentle and lowly in heart, and you will find rest for your souls. For my yoke is easy, and my burden is light."*
>
> — Matthew 11:28-30

> *The words of the wise . . . are given by one Shepherd. My son, beware of anything beyond these. Of making many books there is no end, and much study is a weariness of the flesh.*
>
> — Ecclesiastes 12:11-12

Appropriate, Accessible and Appealing Probabilistic Graphical Models

by

David Iseri Inouye, Ph.D.
The University of Texas at Austin, 2017

Supervisor: Inderjit S. Dhillon
Co-Supervisor: Pradeep Ravikumar

Appropriate - Many multivariate probabilistic models either use independent distributions or dependent Gaussian distributions. Yet, many real-world datasets contain count-valued or non-negative skewed data, e.g. bag-of-words text data and biological sequencing data. Thus, we develop novel probabilistic graphical models for use on count-valued and non-negative data including Poisson graphical models and multinomial graphical models. We develop one generalization that allows for triple-wise or k-wise graphical models going beyond the normal pairwise formulation. Furthermore, we also explore Gaussian-copula graphical models and derive closed-form solutions for the conditional distributions and marginal distributions (both before and after conditioning). Finally, we derive mixture and admixture, or topic model, generalizations of these graphical models to introduce more power and interpretability.

Accessible - Previous multivariate models, especially related to text data, often have complex dependencies without a closed form and require complex inference algorithms that have limited theoretical justification. For example, hierarchical Bayesian models often require marginalizing over many latent variables. We show that our novel graphical models

(even the k-wise interaction models) have simple and intuitive estimation procedures based on node-wise regressions that likely have similar theoretical guarantees as previous work in graphical models. For the copula-based graphical models, we show that simple approximations could still provide useful models; these copula models also come with closed-form conditional *and* marginal distributions, which make them amenable to exploratory inspection and manipulation. The parameters of these models are easy to interpret and thus may be accessible to a wide audience.

Appealing - High-level visualization and interpretation of graphical models with even 100 variables has often been difficult even for a graphical model expert—despite visualization being one of the original motivators for graphical models. This difficulty is likely due to the lack of collaboration between graphical model experts and visualization experts. To begin bridging this gap, we develop a novel "what if?" interaction that manipulates and leverages the probabilistic power of graphical models. Our approach defines: the probabilistic mechanism via conditional probability; the query language to map text input to a conditional probability query; and the formal underlying probabilistic model. We then propose to visualize these query-specific probabilistic graphical models by combining the intuitiveness of force-directed layouts with the beauty and readability of word clouds, which pack many words into valuable screen space while ensuring words do not overlap via pixel-level collision detection. Although both the force-directed layout and the pixel-level packing problems are challenging in their own right, we approximate both simultaneously via adaptive simulated annealing starting from careful initialization. For visualizing mixture distributions, we also design a meaningful mapping from the properties of the mixture distribution to a color in the perceptually uniform CIELUV color space. Finally, we demonstrate our approach via illustrative visualizations of several real-world datasets.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction[1]

1.1 Appropriate

1.1.1 Problem: Inappropriate Probabilistic Models

Gaussian, binary and discrete undirected graphical models—or Markov Random Fields (MRF)—have become popular for compactly modeling and studying the structural dependencies between high-dimensional continuous, binary and categorical data respectively [Friedman et al., 2008, Hsieh et al., 2014, Banerjee et al., 2008, Ravikumar et al., 2010, Jalali et al., 2010]. However, real-world data does not often fit the assumption that variables come from Gaussian or discrete distributions. For example, word counts in documents are nonnegative integers with many zero values and hence are more appropriately modeled by the Poisson distribution. Yet, an independent Poisson distribution would be insufficient because words are often either positively or negatively related to other words—e.g. the words "machine" and "learning" would often co-occur together in ICML papers (positive dependency) whereas the words "deep" and "kernel" would rarely co-occur since they usually refer to different topics (negative dependency). Thus, a Poisson-like model that allows for dependencies between words is desirable. As another example, the delay times at airports are nonnegative continuous values that are more closely modeled by an exponential distribution than a Gaussian distribution but an

[1]Some paragraphs of this chapter are from the drafts of the paper [Inouye et al., 2017] first-authored by David Inouye but co-authored with Eunho Yang, Genevera Allen and Pradeep Ravikumar. See Chapter 5 for more information on David Inouye's contributions.

independent exponential distribution is insufficient because delays at different airports are often related (and sometimes causally related)—e.g. if a flight from Los Angeles, CA (LAX) to San Francisco, CA (SFO) is delayed then it is likely that the return flight of the same airplane will also be delayed. Other examples of non-Gaussian and non-discrete data include high-throughput gene sequencing count data, crime statistics, website visits, survival times, call times and delay times.

Multivariate count-valued data has become increasingly prevalent in modern big data settings. Variables in such data are rarely independent and instead exhibit complex positive and negative dependencies. We highlight three examples of multivariate count-valued data that exhibit rich dependencies: text analysis, genomics, and crime statistics. In text analysis, a standard way to represent documents is to merely count the number of occurrences for each word in the vocabulary and create a word-count vector for each document. This representation is often known as the bag-of-words representation, in which the word order and syntax are ignored. The vocabulary size—i.e. the number of variables in the data—is usually much greater than 1000 unique words, and thus a high-dimensional multivariate distribution is required. Also, words are clearly not independent. For example, if the word "Poisson" appears in a document, then the word "probability" is *more likely* to also appear signifying a positive dependency. Similarly, if the word "art" appears, then the word "probability" is less likely to also appear signifying a negative dependency. In genomics, RNA-sequencing technologies are used to measure gene and isoform expression levels. These technologies yield counts of reads mapped back to DNA locations, that even after normalization, yield non-negative data that is highly skewed with many exact zeros. This genomics data is both high-dimensional, with the number of genes measuring in the tens-of-thousands, and strongly dependent, as genes work together in pathways and complex systems to produce particular phenotypes. In crime analysis, counts of crimes in different counties are clearly multidimensional, with

dependencies between crime counts. For example, the counts of crime in adjacent counties are likely to be correlated with one another, indicating a positive dependency. While positive dependencies are probably more prevalent in crime statistics, negative dependencies might be very interesting. For example, a negative dependency between adjacent counties may suggest that a criminal gang has moved from one county to the other.

These examples motivate the need for a high-dimensional count-valued distribution that permits rich dependencies between variables. In general, a good class of probabilistic models is a fundamental building block for many tasks in data analysis. Estimating such models from data could help answer exploratory questions such as: Which genomic pathways are altered in a disease e.g. by analyzing genomic networks? Or, which county seems to have the strongest effect, with respect to crime, on other counties?

One line of work assumes that the node conditional distributions—i.e. one variable given the values of all the other variables—are univariate exponential families[2] and determines under what conditions a joint distribution exists that is consistent with these node conditional distributions. Besag [1974] developed this multivariate distribution for pairwise dependencies, and Yang et al. [2015] extended this model to k-wise dependencies. Yang et al. [2015] also developed and analyzed an M-estimator based on ℓ_1 regularized node-wise regressions to recover the graphical model structure with high probability. Unfortunately, these models only allowed *negative* dependencies in the case of the exponential and Poisson distributions. Yang et al. [2013] proposed three modifications to the original Poisson model to allow positive dependencies but these modifications alter the Poisson base distribution or require the specification of unintuitive hyperparameters. Allen and Liu [2013] proposed another Poisson graphical model but the model is inconsistent

[2]See [Wainwright and Jordan, 2008] for an introduction to exponential families.

because it cannot be normalized. Thus, we seek to overcome these issues for standard graphical models.

Topic models for count data, such as the well-known Latent Dirichlet Allocation model (LDA), essentially build a hierarchical generative model for count data. Topic models are generalizations of mixture distributions; in particular, they assume that each observation can come from more than one component. However, the topic distributions are assumed to be independent given the topic assignments. Yet in real-world datasets, even words within a topic may be dependent; for example, the words "machine" and "learning" should be dependent in a topic about computer science. Thus, the fundamental assumption of conditional independence is often violated in many real-world datasets [Mimno and Blei, 2011].

1.1.2 Proposed Solution: General Graphical Models with Positive Dependencies

We propose two graphical models to help provide multivariate models that appropriately fit non-Gaussian data such as count-valued or non-negative data. In particular, we propose the fixed-length Poisson MRF (LPMRF) in Chapter 3 that relaxes the independence assumption of the multinomial distribution, which is merely the sum of independent categorical variables. This allows for a count-valued graphical model which allows for explicit positive and negative dependencies between variables. We also propose a more general class of models called square root graphical models (SQR) in Chapter 4 which provides elegant Poisson and exponential graphical models that permit *positive* dependencies—unlike the previous graphical models in [Besag, 1974, Yang et al., 2015].

In addition, we develop two generalizations of topic models in Chapter 6 to overcome the assumption that topics or components are independent distributions. We then propose two novel topic models using undirected graphical models based on these generalizations in

Chapters 7 and 8. These novel topic models fundamentally extend the standard notions of topic models by allowing dependencies even within each topic.

1.2 Accessible

1.2.1 Problem: Complex Estimation Algorithms or Difficult to Interpret Parameters

General multivariate probabilistic models—except for the well-known discrete or Gaussian graphical model—have struggled to become mainstream because they are either difficult to interpret or have complex estimation algorithms. This makes them inaccessible to the general science community. One thread that has become prevalent is the hierarchical Bayesian approach including topic modeling [Blei et al., 2010]. The main idea is to manually construct a directed acyclic graphical model—often with many latent variables—and then learn these latent variables and corresponding parameters. However, these hierarchical models require manual construction and usually manual development of the learning algorithm (though there has been some progress in general Bayesian learning, see `http://probabilistic-programming.org/wiki/Home`). In addition, to compute the likelihood, the latent variables must be marginalized out, which is often quite difficult.

Previous graphical models represented by [Yang et al., 2015] were restricted to negative dependencies in the case of the Poisson model. While [Yang et al., 2013] proposed modifications of the original Poisson graphical model to allow for positive dependencies, the modifications significantly altered the distribution and required the user to specify truncation parameters.

1.2.2 Proposed Solution: Explicitly Model Dependencies and Estimate Parameters with Fast Convex Optimization

As in [Yang et al., 2015], we explicitly model dependencies in our models by associating a parameter with each dependency under a general graphical model formulation. Unlike the models in [Yang et al., 2015], however, our models permit *positive* dependencies, which are much easier to interpret than negative dependencies. This enables simpler interpretation of the models. We propose Newton-like algorithms based on the work in [Hsieh et al., 2011] to learn these models much more quickly than the proximal gradient algorithms in [Yang et al., 2013, 2015]. As with the previous graphical models, our estimation algorithms are also easier to interpret than the inference algorithms in hierarchical Bayesian or copula models because they are merely node-wise regressions with a particular loss function based on a univariate exponential family.

1.3 Appealing

1.3.1 Problem: Difficult to Visualize, Interpret or Provide Feedback

General high-dimensional graphical models—despite their name—have often been difficult to visualize or interpret. In particular, graphical models involving thousands of words are often difficult to understand even for graphical model experts. One naïve visualization would be to merely list the top edges. However, this would not easily show the global structure because it would be difficult to understand even twenty edges at one time. Another idea is to use a matrix to show the edge weights. While this may be clearer than a list of edge weights, the ordering of matrix columns is very important to the interpretation and the global structure may be difficult to understand if there are more than twenty or thirty variables. As another idea, people have visualized graphical models using network visualization programs

such as Gephi[3] or Cytoscape[4]. While network visualization can definitely provide a higher-level view of the model, the software often suffers from overlapping labels so that it is difficult to see many variables at once. In addition, these network visualization tools are only post-processing tools after the model has been fit and do not enable the user to interact with the model. All of these visualizations only provide one view of the model, and thus they cannot show multiple perspectives of the high-dimensional model. Intuitive and powerful user interaction with the models is required to better interpret and understand these high-dimensional and complex graphical models.

1.3.2 Proposed Solution: Intuitive and Interactive Visualization Framework

We propose an intuitive and interactive visualization framework that seeks to combine the mathematical modeling power of probabilistic graphical models, the aesthetics of word clouds such as Wordle[5], and the high-level intuitive idea of network visualization. In Chapter 11, we first describe a novel and intuitive user interaction by converting 'what if?' queries into concrete probabilistic operations on the model. This provides a simple and transparent interface to the user without requiring significant knowledge about the variable values or underlying probabilistic model; this makes the visualization accessible to a wider audience than graphical model experts. In Chapter 12, we propose our novel model visualization which visualizes a graphical model via a beautiful visualization that is semantically meaningful, easy to understand and aesthetically pleasing. By combining the power of graphical models and the beauty of visualization, we develop a novel interaction and visualization that allows exploratory analysis of multiple types of datasets including text data and non-text data such as airport delay times and daily stock returns.

[3] `gephi.org`
[4] `cytoscape.org`
[5] `wordle.net`

1.4 Preliminaries

1.4.1 Notation

Let p and n denote the number of dimensions and instances respectively. We will usually use s or t to denote indices of the dimension and i as an index of the instance. Unless otherwise indicated, we will let $X \in \mathbb{R}^{p \times n}$ denote the data matrix. Parameters of distributions will usually be denoted with Greek letters such as $\boldsymbol{\theta}$, Φ, or Ψ. We will generally use uppercase letters for matrices (e.g. Φ, X), boldface lowercase letters or indices of matrices for column vectors (i.e $\boldsymbol{x}_i, \boldsymbol{\theta}, \Phi_s$) and lowercase letters for scalar values (i.e. x_{si}, θ_s). Let $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_+$ denotes the *nonnegative* real numbers, and $\mathbb{R}_{++}$ denotes the strictly *positive* real numbers. Similarly, $\mathbb{Z}$ denotes the set of integers, and $\mathbb{Z}_+$ and $\mathbb{Z}_{++}$ are defined similarly. We denote the standard basis vectors as $\boldsymbol{e}_s = [0, \cdots, 0, 1, 0, \cdots, 0]^T$ and the ones vector as $\boldsymbol{e} = [1, 1, \cdots, 1]^T$. Let $\boldsymbol{x}^p$ and $\sqrt[j]{\boldsymbol{x}}$ to be the entry-wise power and j-th root of the vector $\boldsymbol{x}$. For an exponential family, let $T(\cdot)$, $B(\cdot)$ and $A(\cdot)$ be the sufficient statistic, log base measure and log partition function (i.e. log normalizing constant) respectively.

1.4.2 Exponential Families

We briefly describe exponential family distributions which form the basis fore the graphical models developed throughout this work. Many commonly used distributions fall into this family, including Gaussian, Bernoulli, exponential, gamma, and Poisson, among others. The exponential family is specified by a vector of sufficient statistics denoted by $T(\boldsymbol{x}) \equiv [T_1(\boldsymbol{x}), T_2(\boldsymbol{x}), \cdots, T_m(\boldsymbol{x})]$, the log base measure $B(\boldsymbol{x})$ and the domain of the random variable $\mathcal{D}$. With this notation, the generic exponential family is defined as:

$$\mathbb{P}_{\text{ExpFam}}(\boldsymbol{x} \,|\, \boldsymbol{\eta}) = \exp\left(\sum_{i=1}^{m} \eta_i T_i(\boldsymbol{x}) + B(\boldsymbol{x}) - A(\boldsymbol{\eta})\right)$$

$$A(\boldsymbol{\eta}) = \log \int_{\mathcal{D}} \exp\left(\sum_{i=1}^{m} \eta_i T_i(\boldsymbol{x}) + B(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \,,$$

where $\boldsymbol{\eta}$ are called the *natural or canonical parameters* of the distribution, μ is the Lebesgue or counting measure depending on whether $\mathcal{D}$ is continuous or discrete respectively, and $A(\boldsymbol{\eta})$ is called the *log partition function* or *log normalization constant* because it normalizes the distribution over the domain $\mathcal{D}$. Note that the sufficient statistics $\{T_i(\boldsymbol{x})\}_{i=1}^m$ can be any arbitrary function of $\boldsymbol{x}$; for example, $T_i(\boldsymbol{x}) = x_1 x_2$ could be used to model interaction between x_1 and x_2. The log partition function $A(\boldsymbol{\eta})$ will be a key quantity when discussing the following models: $A(\boldsymbol{\eta})$ must be finite for the distribution to be valid, so that the realizable domain of parameters is given by $\{\boldsymbol{\eta} \in \mathcal{D} : A(\boldsymbol{\eta}) < \infty\}$. Thus, for instance, if the realizable domain only allows positive or negative interaction terms for instance, that would severely restrict the set of allowed dependencies in the model.

Let us now consider the exponential family form of the univariate Poisson as an example:

$$\begin{aligned}
\mathbb{P}_{\text{Poiss}}(x \,|\, \lambda) &= \lambda^x / x! \exp(-\lambda) \\
&= \exp(\log(\lambda^x) - \log(x!) - \lambda) \\
&= \exp(\underbrace{\log(\lambda)}_{\eta} \underbrace{x}_{T(x)} + \underbrace{(-\log(x!))}_{B(x)} - \lambda), \quad \text{and therefore} \\
\mathbb{P}_{\text{Poiss}}(x \,|\, \eta) &= \exp(\eta x - \log(x!) - \exp(\eta)), \qquad (1.1)
\end{aligned}$$

where $\eta \equiv \log(\lambda)$ is the natural parameter of the Poisson, $T(x) = x$ is the Poisson sufficient statistic, $-\log(x!)$ is the Poisson log base measure and $A(\eta) = \exp(\eta)$ is the Poisson log partition function. Note that for the general exponential family distribution, the log partition function may not have a closed form.

Part I

Novel Graphical Models with Positive Dependencies

Summary of Part I

In the following chapters, we first present previous graphical models—focusing on ones that can handle count-valued data based on the univariate Poisson. One of the key challenges in these previous graphical models is allowing positive dependencies in an elegant way. Previous attempts either required unintuitive hyperparameters such as truncation value [Yang et al., 2013] or ignored the consistency of the joint distribution [Allen and Liu, 2012, 2013]. These models provide the motivation and basis for the new graphical models that we introduce in Chapters 3 and 4. Chapter 3 introduces the novel fixed-length Poisson MRF (LPMRF) for count data by assuming the length of the vector (i.e. it's ℓ_1 norm) is known or fixed; the difference between the LPMRF model and previous Poisson graphical models is similar to the difference between a independent Poisson distribution and a multinomial distribution, which has a fixed number of trials. In Chapter 4, we introduce the novel square-root graphical model (SQR) that elegantly allows both *positive* and *negative* dependencies by taking the square root of the sufficient statistics; no hyperparameters are required and the joint distribution is consistent with the conditional distributions. Finally, in Chapter 5, we compare our proposed graphical models to other multivariate distributions derived for count data. In particular, we review marginal Poisson models as exemplified by copula-based models and mixtures of independent Poisson distributions. Then, we extensively compare all of these models both qualitatively and quantitatively on real-world datasets.

Chapter 2

Previous Graphical Models[1]

In this chapter, we present a brief background on the graphical model class as in [Besag, 1974, Yang et al., 2015][2] along with derivative works in [Yang et al., 2013, Allen and Liu, 2012, 2013]. As will be evident, these previous graphical models either only allow *negative* dependencies [Yang et al., 2015], require unintuitive hyperparameters such as truncation value [Yang et al., 2013] or ignore joint consistency [Allen and Liu, 2012, 2013]. Thus, we develop new models to overcome these issues in later chapters.

We now more formally define the previous graphical models from [Besag, 1974, Yang et al., 2015]. Let $T(x)$ and $B(x)$ be the sufficient statistics and log base measure respectively of the base univariate exponential family and let $\mathcal{D} \subseteq \mathbb{R}_+^p$ be the domain of the random vector. We will denote $T(\boldsymbol{x})\colon \mathbb{R}^p \to \mathbb{R}^p$ to be the entry-wise application of the sufficient statistic function to each entry in the vector $\boldsymbol{x}$. With this notation, the previous class of graphical models can be defined as [Yang et al., 2015]:

$$\mathbb{P}(\boldsymbol{x}|\theta, \Phi) = \exp\left(\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T \Phi T(\boldsymbol{x}) \sum_{s=1}^{p} B(x_s) - A(\boldsymbol{\theta}, \Phi)\right) \tag{2.1}$$

$$A(\boldsymbol{\theta}, \Phi) = \int_{\mathcal{D}} \exp\left(\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T \Phi T(\boldsymbol{x}) + \sum_{s=1}^{p} B(x_s)\right) \mathrm{d}\mu(\boldsymbol{x})\,, \tag{2.2}$$

[1] Parts of this chapter are from drafts of [Inouye et al., 2017]. See Chapter 5 for more information on David Inouye's contribution.

[2] Besag [1974] originally named these Poisson auto models, focusing on pairwise graphical models, but [Yang et al., 2015] considers the general graphical model setting.

where $A(\boldsymbol{\theta}, \Phi)$ is the log partition function (i.e. log normalization constant) which is required for probability normalization, $\Phi \in \mathbb{R}^{p \times p}$ is symmetric with *zeros along the diagonal* and μ is either the standard Lebesgue measure or the counting measure depending on whether the domain $\mathcal{D}$ is continuous or discrete. The only difference from a fully independent model is the quadratic interaction term $T(\boldsymbol{x})^T \Phi T(\boldsymbol{x})$—i.e. $O(T(x)^2)$—which is why the exponential and Poisson cases do not admit positive dependencies as will be described in later sections.

This graphical model was based on assuming that the node-conditional distributions (i.e. the univariate distribution of one variable given all other variables) is specified by a univariate exponential family. Yang et al. [2015] show that if this assumption is made, the consistent joint distribution restricted to pairwise interactions is of the form in Eqn. 2.1—and in fact is necessarily this form.

Parameter estimation in a this graphical model is naturally suggested by its construction: all of the parameters in (2.1) can be estimated by considering the node conditional distributions for each node separately, and solving an ℓ_1-regularized regression for each variable. This parameter estimation approach is not only simple, but is also guaranteed to be consistent even under high dimensional sampling regimes, under some other mild conditions including a sparse graph structural assumption (see Yang et al. [2012, 2015] for more details on the analysis).

2.1 Poisson Graphical Models (PGM/PMRF)

For the Poisson, $T(x) = x$, $B(x) = -\log(x!)$ and $x \in \mathbb{Z}_+$, and thus the Poisson instantiation is as follows:

$$\mathbb{P}_{\text{PMRF}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \Phi) = \exp\left(\boldsymbol{\theta}^T \boldsymbol{x} + \boldsymbol{x}^T \Phi \boldsymbol{x} - \sum_{s=1}^{p} \log(x_s!) - \mathrm{A}\left(\boldsymbol{\theta}, \Phi\right)\right), \tag{2.3}$$

In spite of its simple parameter estimation method, the major drawback with the Poisson graphical model, or Poisson Markov Random Field (PMRF), instantiation is that it only

permits *negative* conditional dependencies between variables:

Proposition 1 (Besag [1974])**.** *Consider the Poisson graphical model distribution in* (2.1)*. Then, for any parameters* $\boldsymbol{\theta}$ *and* Φ*,* $A_{PGM}(\boldsymbol{\theta}, \Phi) < +\infty$ *only if the pairwise parameters are non-positive:* $\phi_{st} \leq 0$*,* $\forall s \neq t$*.*

Intuitively, if any entry in Φ, say Φ_{st}, is positive, the term $\Phi_{st}\boldsymbol{x}_s\boldsymbol{x}_t$ would grow quadratically, whereas the log base measure terms $-\log(x_s!) - \log(x_t!)$ only decreases as $O(x_s\log(x_s) + x_t\log(x_t))$, so $A(\boldsymbol{\theta}, \Phi) \to \infty$ as $x_s, x_t \to \infty$. Thus, even though the Poisson graphical model is a natural extension of the univariate Poisson distribution (from the node conditional viewpoint), it entails a highly restrictive parameter space, with severely limited applicability. Thus, multiple PGM extensions attempt to relax this negativity restriction to permit positive dependencies as described next.

2.2 Extensions of Poisson Graphical Models

To circumvent the severe limitations of the PGM distribution which in particular only permits negative conditional dependencies, several extensions to PGM that permit a richer dependence structure have been proposed.

2.2.1 Truncated PGM (TPGM/TPMRF)

Because the negativity constraint is due in part to the infinite domain of count variable, a natural solution would be to truncate the domain of variables. It was Kaiser and Cressie [1997] who first introduced an approach to truncate the Poisson distribution in the context of graphical models. Their idea was simply to use a Winsorized Poisson distribution for node conditional distributions: x is a Winsorized Poisson if $z = \mathbb{I}(z' < R)z' + \mathbb{I}(z' \geq R)R$, where z' is Poisson, $\mathbb{I}(\cdot)$ is an indicator function, and R is a fixed positive constant denoting the truncation level. However, Yang et al. [2013] showed

that Winsorized node conditional distributions actually does *not* lead to a consistent joint distribution.

As an alternative way of truncation, Yang et al. [2013] instead keep the same parametric form as PGM but merely truncate the domain to non-negative integers less than or equal to R—i.e. $\mathcal{D}_{\text{TPGM}} = \{0, 1, \cdots, R\}$, so that the joint distribution takes the form [Yang et al., 2015]:

$$\mathbb{P}_{\text{TPGM}}(\boldsymbol{x}) = \exp\{\boldsymbol{\theta}^T \boldsymbol{x} + \boldsymbol{x}^T \Phi \boldsymbol{x} - \textstyle\sum_i \log(x_s!) - A_{\text{TPGM}}(\boldsymbol{\theta}, \Phi)\}\,. \tag{2.4}$$

As they show, the node-conditional distributions of this graphical model distribution belong to an exponential family that is Poisson-like, but with domain bounded by R. Thus, the key difference from the vanilla Poisson graphical model is that the domain is finite, and hence the log partition function $A_{\text{TPGM}}(\cdot)$ only involves a finite number of summations. Thus, no restrictions are imposed on the parameters for the normalizability of the distribution.

Yang et al. [2013] discuss several major drawbacks to TPGM. First, the domain needs to be bounded a priori, so that R should ideally be set larger than any unseen observation. Second, the effective range of parameter space for a non-degenerate distribution is still limited: as the truncation value R increases, the effective values of pairwise parameters become increasingly negative or close to zero—otherwise, the distribution can be degenerate placing most of its probability mass at 0 or R.

2.2.2 Quadratic PGM (QPGM/QPMRF)

Yang et al. [2013] also investigate the possibility of Poisson graphical models that (a) allows both positive and negative dependencies, as well as (b) allow the domain to range over all non-negative integers. As described previously, a key reason for the negative constraint on the pairwise parameters ϕ_{st} is that the log base measure $\sum_s \log(x_s!)$ scales more slowly

than the quadratic pairwise term $\boldsymbol{x}^T\Phi\boldsymbol{x}$ where $\boldsymbol{x} \in \mathbb{Z}_+^p$. Yang et al. [2013] thus propose two possible solutions: increase the base measure or decrease the quadratic pairwise term.

First, if we modify the base measure of Poisson distribution with "Gaussian-esque" quadratic functions (note that for the linear sufficient statistics with positive dependencies, the base measures should be quadratic at the very least [Yang et al., 2013]), then the joint distribution, which they call a quadratic PGM, is normalizable while allowing both positive and negative dependencies [Yang et al., 2013]:

$$\mathbb{P}_{\text{QPGM}}(\boldsymbol{x}) = \exp\{\boldsymbol{\theta}^T\boldsymbol{x} + \boldsymbol{x}^T\Phi\boldsymbol{x} - A_{\text{QPGM}}(\boldsymbol{\theta}, \Phi)\}. \tag{2.5}$$

Essentially, QPGM has the same form as the Gaussian distribution, but where its domain is the set of non-negative integers. The key differences from PGM are that Φ can have negative values along the diagonal, and the Poisson base measure $\sum_s -\log(x_s!)$ is replaced by the quadratic term $\sum_s \phi_{ss}x_s^2$. Note that a sufficient condition for the distribution to be normalizable is given by:

$$\boldsymbol{x}^T\Phi\boldsymbol{x} \;<\; -c\|\boldsymbol{x}\|_2^2 \quad \forall \boldsymbol{x} \in \mathbb{Z}_+^p\,, \tag{2.6}$$

for some constant $c > 0$, which in turn can be satisfied if Φ is negative definite. One significant drawback of QPGM is that the tail is Gaussian-esque and thin rather than Poisson-esque and thicker as in PGM.

2.2.3 Sub-Linear PGM (SPGM/SPMRF)

Another possible modification is to use sub-linear sufficient statistics in order to preserve the Poisson base measure and possibly heavier tails. Consider the following univariate distribution over count-valued variables:

$$\mathbb{P}(z) \propto \exp\{\theta T(z\,;\, R_0, R) - \log z!\}\,, \tag{2.7}$$

which has the same base measure $\log z!$ as the Poisson, but with the following sub-linear sufficient statistics:

$$T(z\,;\,R_0,R) = \begin{cases} z & \text{if } z \le R_0 \\ -\frac{1}{2(R-R_0)}\,z^2 + \frac{R}{R-R_0}\,x - \frac{R_0^2}{2(R-R_0)} & \text{if } R_0 < z \le R \\ \frac{R+R_0}{2} & \text{if } z \ge R\,. \end{cases} \tag{2.8}$$

For values of x up to R_0, $T(x)$ increases linearly, while after R_0 its slope decreases linearly, and finally after R, $T(x)$ becomes constant. The joint graphical model, which they call a sub-linear PGM (SPGM), specified by the node conditional distributions belonging to the family (2.7), has the following form:

$$\mathbb{P}_{\text{SPGM}}(\boldsymbol{x}) = \exp\{\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T\,\Phi\,T(\boldsymbol{x}) - \textstyle\sum_s \log(x_s!) - A_{\text{SPGM}}(\boldsymbol{\theta},\Phi \mid R_0,R)\}\,, \tag{2.9}$$

where

$$A_{\text{SPGM}}(\boldsymbol{\theta},\Phi \mid R_0,R) = \log \sum_{\boldsymbol{x}\in\mathbb{Z}_+} \exp\{\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T\,\Phi\,T(\boldsymbol{x}) - \textstyle\sum_s \log(x_s!)\}\,, \tag{2.10}$$

and $T(\boldsymbol{x})$ is the entry-wise application of the function in (2.8). SPGM is always normalizable for $\phi_{st} \in \mathbb{R}\,\forall\, s \neq t$ [Yang et al., 2013].

The main difficulty in estimating Poisson graphical model variants above with infinite domain is the lack of closed-form expressions for the log partition function, even just for the node-conditional distributions that are needed for parameter estimation. Yang et al. [2013] propose an approximate estimation procedure that uses the univariate Poisson and Gaussian log partition functions as upper bounds for the node-conditional log-partition functions for the QPGM and SPGM models respectively.

2.2.4 Local PGM

Inspired by the neighborhood selection technique of Meinshausen and Bühlmann [2006], Allen and Liu [2012, 2013] propose to learn the network structure of count-valued

data by fitting a series of ℓ_1-regularized Poisson regressions to learn the node-neighborhoods. Such an estimation method may yield interesting network estimates, but as Allen and Liu [2013] note, these estimates do not correspond to a consistent joint density. Instead, the underlying model is defined in terms of a series of local models where each variable is conditionally Poisson given its node-neighbors; this approach is thus termed the local Poisson graphical model (LPGM). Note that LPGM does not impose any restrictions on the parameter space or types of dependencies; if the parameter space of each local model was constrained to be non-positive, then the LPGM reduces to the vanilla Poisson graphical model as previously discussed. Hence, the LPGM is less interesting as a candidate multivariate model for count-valued data, but many may still find its simple and interpretable network estimates appealing. Recently, several have proposed to adopt this estimation strategy for alternative network types [Hadiji et al., 2015, Han and Zhong, 2016].

2.3 Exponential Graphical Models

We also review the exponential instantiation of the graphical model from [Yang et al., 2015] which the domain $\mathcal{D} \in \mathbb{R}_+^p$, $T(x) = x$ and $B(x) = 0$. Suppose there is even one positive entry in Φ denoted ϕ_{st}. Then as $\boldsymbol{x} \to \infty$, the positive quadratic term $x_s \phi_{st} x_t$ will dominate the linear term $\boldsymbol{\theta}^T \boldsymbol{x}$ and thus the log partition function will diverge (i.e. $A(\boldsymbol{\theta}, \Phi) \to \infty$). Thus, as with the Poisson instantiation, $\Phi_{st} \leq 0$ is required for a consistent joint distribution.

2.4 Conclusion

In the next chapters, we seek to overcome the main issue with the previous graphical models [Yang et al., 2015] that only allows negative dependencies for the Poisson and exponential instantiations. We would like to maintain the univariate distribution's

characteristics (such as a sub-quadratic tails for the Poisson and exponential distributions) while allowing both *positive and negative* dependencies. In addition, we would prefer to avoid requiring the specification of hyperparameters. In Chapters 3 and 4, we develop one graphical model distribution for the Poisson case called a fixed-length Poisson MRF (LPMRF) and a more elegant class of graphical models called square root graphical models (SQR) that generalizes both the Poisson and exponential distributions.

Chapter 3

Fixed-Length Poisson MRF[1]

3.1 Abstract

We propose a novel distribution that generalizes the Multinomial distribution to enable dependencies between dimensions. Our novel distribution is based on the parametric form of the Poisson MRF model [Yang et al., 2012] but is fundamentally different because of the domain restriction to a fixed-length vector similar to the multinomial distribution where the number of trials is fixed or known. Thus, we propose the Fixed-Length Poisson MRF (LPMRF) distribution. We develop AIS sampling methods to estimate the likelihood and log partition function (i.e. the log normalizing constant), which was not developed for the Poisson MRF model.

3.2 Introduction & Related Work

The multinomial distribution seems to be a natural distribution for modeling count-valued data such as text documents. Indeed, most topic models such as PLSA [Hofmann, 1999], LDA [Blei et al., 2003] and numerous extensions—see [Blei et al., 2010] for a survey of probabilistic topic models—use the multinomial as the fundamental base distribution while adding complexity using other latent variables. This is most likely due to the extreme simplicity of multinomial parameter estimation—simple frequency counts—that is usually

[1]The majority of this chapter is from [Inouye et al., 2015] with some edits for better integration into this dissertation. [Inouye et al., 2015] was primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

smoothed by the simple Dirichlet conjugate prior. In addition, because the multinomial requires the length of a document to be fixed or pre-specified, usually a Poisson distribution on document length is assumed. This yields a Poisson-multinomial distribution—which by well-known results is merely an independent Poisson model.[2] However, the multinomial assumes independence between the words because the multinomial is merely the sum of independent categorical variables. This restriction does not seem to fit with real-world text. For example, words like "neural" and "network" will tend to co-occur quite frequently together in NIPS papers. Thus, we seek to relax the word independence assumption of the multinomial.

While the Truncated Poisson graphical model (TPGM/TPMRF) [Yang et al., 2013] described in Chapter 2 may provide interesting parameter estimates, a TPMRF with positive dependencies may be almost entirely concentrated at the corners of the joint distribution because of the quadratic term in the log probability (see the bottom left of Fig. 3.1). In addition, the log partition function of the TPMRF is intractable to estimate even for a small number of dimensions because the sum is over an exponential number of terms.

Thus, we develop a different distribution than a TPMRF that allows positive dependencies but is more appropriately normalized. We observe that the multinomial is proportional to an independent Poisson model with the domain restricted to a fixed length L. Thus, in a similar way, we propose a Fixed-Length Poisson MRF (LPMRF) that is proportional to a PMRF but is restricted to a domain with a fixed vector length—i.e. where $\|\boldsymbol{x}\|_1 = L$. This distribution is quite different from previous PMRF variants because the normalization is very different as will be described in later sections. For a motivating example, in Fig. 3.1, we show the marginal distributions of the empirical distribution and fitted models using only three words from the Classic3 dataset that contains documents

[2]The assumption of Poisson document length is not important for most topic models [Blei et al., 2003].

regarding library sciences and aerospace engineering (See Sec. 8.4). Clearly, real-world text has positive dependencies as evidenced by the empirical marginals of "boundary" and "layer" (i.e. referring to the boundary layer in fluid dynamics) and LPMRF does the best at fitting this empirical distribution. In addition, the log partition function—and hence the likelihood—for LPMRF can be approximated using sampling as described in later sections. Under the PMRF or TPMRF models, both the log partition function and likelihood were computationally intractable to compute exactly.[3] Thus, approximating the log partition function of an LPMRF opens up the door for likelihood-based hyperparameter estimation and model evaluation that was not possible with PMRF.

3.3 Fixed-Length Poisson MRF

LPMRF Definition The Fixed-Length Poisson MRF (LPGM/LPMRF) distribution is a simple yet fundamentally different distribution than the (PGM/PMRF). Letting $L \equiv \|\boldsymbol{x}\|_1$ be the length of document, we define the LPMRF distribution as follows:

$$\mathbb{P}_{\text{LPMRF}}(\boldsymbol{x}|\boldsymbol{\theta}, \Phi, L) = \exp(\boldsymbol{\theta}^T\boldsymbol{x} + \boldsymbol{x}^T\Phi\boldsymbol{x} - \textstyle\sum_s \log(x_s!) - A_L(\boldsymbol{\theta}, \Phi)) \tag{3.1}$$

$$A_L(\boldsymbol{\theta}, \Phi) = \log \sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp(\boldsymbol{\theta}^T\boldsymbol{x} + \boldsymbol{x}^T\Phi\boldsymbol{x} - \textstyle\sum_s \log(x_s!)) \tag{3.2}$$

$$\mathcal{X}_L = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{Z}_+^p, \|\boldsymbol{x}\|_1 = L\}. \tag{3.3}$$

The only difference from the PMRF parametric form is the log partition function $A_L(\boldsymbol{\theta}, \Phi)$ which is conditioned on the set $\mathcal{X}_L$ (unlike the unbounded set for PMRF). This domain restriction is critical to formulating a tractable and reasonable distribution. Combined with a Poisson distribution on vector length $L = \|\boldsymbol{x}\|_1$, the LPMRF distribution can be a much more suitable distribution for documents than a multinomial. The LPMRF distribution reduces to the standard multinomial if there are no dependencies. However, if there are dependencies,

[3]The example in Fig. 3.1 was computed by exhaustively computing the log partition function.

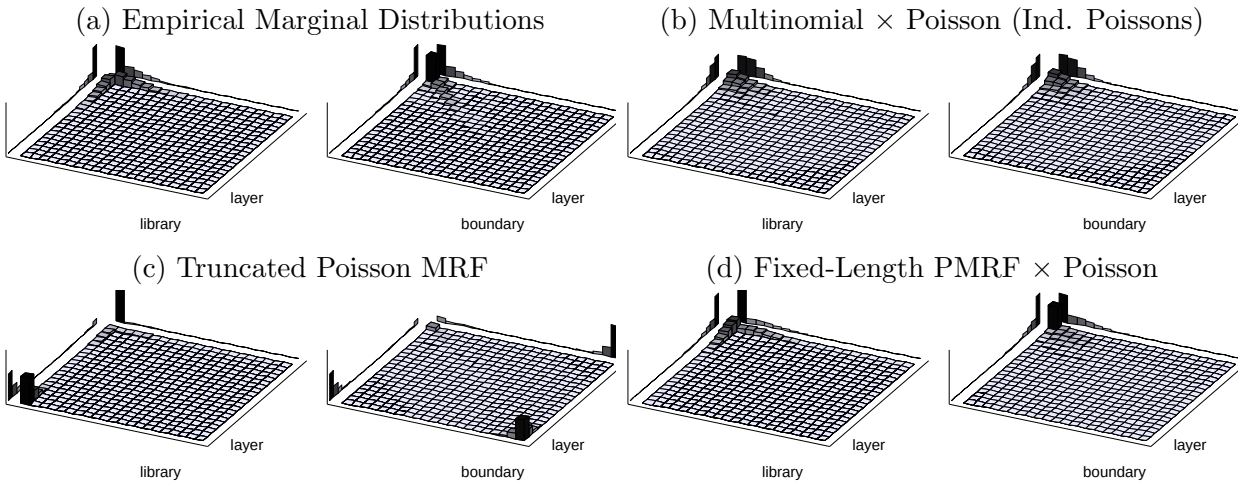


Figure 3.1: **Marginal Distributions from Classic3 Dataset** (Top Left) Empirical Distribution, (Top Right) Estimated multinomial × Poisson joint distribution—i.e. independent Poissons, (Bottom Left) Truncated Poisson MRF, (Bottom Right) Fixed-Length PMRF × Poisson joint distribution. The simple empirical distribution clearly shows a strong dependency between "boundary" and "layer" but strong negative dependency of "boundary" with "library". Clearly, the word-independent multinomial-Poisson distribution underfits the data. While the Truncated PMRF can model dependencies, it obviously has normalization problems because the normalization is dominated by the edge case. The LPMRF-Poisson distribution much more appropriately fits the empirical data.

then the distribution can be quite different than a multinomial as illustrated in Fig. 3.2 for an LPMRF with $p = 2$ and L fixed at either 10 or 20 words. After the original submission, we realized that for $p = 2$ the LPMRF model is the same as the multiplicative binomial generalization in [Altham, 1978]. Thus, the LPMRF model can be seen as a multinomial generalization ($p \geq 2$) of the multiplicative binomial in [Altham, 1978].

LPMRF Parameter Estimation Because the parametric form of the LPMRF model is the same as the form of the PMRF model and we primarily care about finding the correct dependencies, we decide to use the PMRF estimation algorithm described in [Inouye et al., 2014a] to estimate $\boldsymbol{\theta}$ and Φ. The algorithm in [Inouye et al., 2014a] uses an approximation

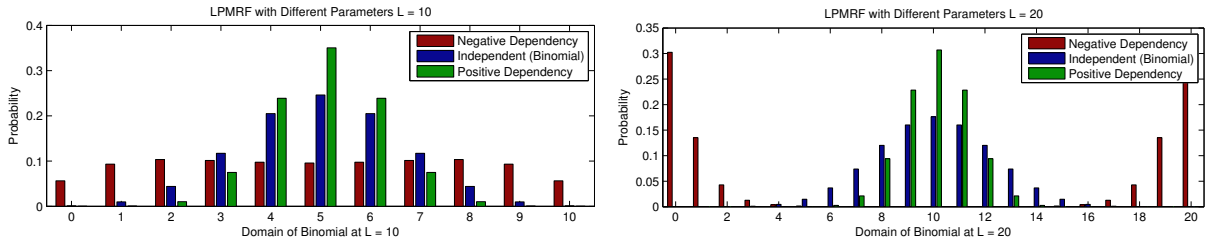


Figure 3.2: LPMRF distribution for $L = 10$ (left) and $L = 20$ (right) with negative, zero and positive dependencies. The distribution of LPMRF can be quite different than a multinomial (zero dependency) and thus provides a much more flexible parametric distribution for count data.

to the likelihood by using the pseudo-likelihood and performing ℓ_1 regularized node-wise Poisson regressions. The ℓ_1 regularization is important both for the sparsity of the dependencies and the computational efficiency of the algorithm. While the PMRF and LPMRF are different distributions, the pseudo-likelihood approximation for estimation provides good results as shown in the results section. We present timing results to show the scalability of this algorithm in Sec. 8.6. Other parameter estimation methods would be an interesting area of future work.

3.3.1 Likelihood and Log Partition Estimation

Unlike previous work on the PMRF or TPMRF distributions, we develop a tractable approximation to the LPMRF log partition function (Eq. 3.2) so that we can compute approximate likelihood values. The likelihood of a model can be fundamentally important for hyperparameter optimization and model evaluation.

LPMRF Annealed Importance Sampling First, we develop an LPMRF Gibbs sampler by considering the most common form of multinomial sampling, namely by taking the sum of a sequence of L Categorical variables. From this intuition, we sample one word at a time

while holding all other words fixed. The probability of one word in the sequence $\boldsymbol{w}_\ell$ given all the other words is proportional to $\exp(\boldsymbol{\theta}_s + 2\Phi_s \boldsymbol{x}_{-\ell})$ where $\boldsymbol{x}_{-\ell}$ is the sum of all other words. See the Appendix for the details of Gibbs sampling. Then, we derive an annealed importance sampler [Neal, 2001] using the Gibbs sampling by scaling the Φ matrix for each successive distribution by the linear sequence starting with 0 and ending with 1 (i.e. $\gamma = 0, \ldots, 1$). Thus, we start with a simple multinomial sample from $\mathbb{P}(x \mid \boldsymbol{\theta}, 0 \cdot \Phi, L) = \mathbb{P}_{\text{Mult}}(x \mid \boldsymbol{\theta}, L)$ and then Gibbs sample from each successive distribution $\mathbb{P}_{\text{LPMRF}}(x \mid \boldsymbol{\theta}, \gamma\Phi, L)$ updating the sample weight as defined in [Neal, 2001] until we reach the final distribution when $\gamma = 1$. From these weighted samples, we can compute an estimate of the log partition function [Neal, 2001].

Upper Bound Using Hölder's inequality, a simple convex relaxation and the partition function of a multinomial, an upper bound for the log partition function can be computed: $A_L(\boldsymbol{\theta}, \Phi) \leq L^2 \lambda_{\Phi,1} + L\log(\sum_s \exp \theta_s) - \log(L!)$, where $\lambda_{\Phi,1}$ is the maximum eigenvalue of Φ. See the Appendix for the full derivation. We simplify this upper bound by subtracting $\log(\sum_s \exp \theta_s)$ from $\boldsymbol{\theta}$ (which does not change the distribution) so that the second term becomes 0. Then, neglecting the constant term $-\log(L!)$ that does not interact with the parameters $(\boldsymbol{\theta}, \Phi)$, the log partition function is upper bounded by a simple quadratic function w.r.t. L.

Weighting Φ for Different L For datasets in which L is observed for every sample but is not uniform—such as document collections, the log partition function will grow quadratically in L if there are any positive dependencies as suggested by the upper bound. This causes long documents to have extremely small likelihood. Thus, we must modify Φ as L gets larger to counteract this effect. We propose a simple modification that scales the Φ for each L: $\tilde{\Phi}^L = \omega(L)\Phi$. In particular, we propose to use the sigmoidal function using the Log Logistic

cumulative distribution function (CDF): $\omega(L) = 1 - \text{LogLogisticCDF}(L \,|\, \alpha_{\text{LL}}, \beta_{\text{LL}})$. We set the β_{LL} parameter to 2 so that the tail is $O(1/L^2)$ which will eventually cause the upper bound to approach a constant. Letting $\bar{L} = \frac{1}{n}\sum_i L_i$ be the mean instance length, we choose $\alpha_{\text{LL}} = c\bar{L}$ for some small constant c. This choice of α_{LL} helps the weighting function to appropriately scale for corpora of different average lengths.

Final Approximation Method for All L For our experiments, we approximate the log partition function value for all L in the range of the corpus. We use 100 AIS samples for 50 different test values of L linearly spaced between the $0.5\bar{L}$ and $3\bar{L}$ so that we cover both small and large values of L. This gives a total of 5,000 annealed importance samples. We use the quadratic form of the upper bound $U_a(L) = \omega(L)L^2a$ (ignoring constants with respect to Φ) and find a constant a that upper bounds all 50 estimates: $a_{\max} = \max_L[\omega(L)L^2]^{-1}(\hat{A}_L(\boldsymbol{\theta}, \Phi) - L\log(\sum_s \exp\theta_s) + \log(L!))$, where $\hat{A}_L$ is an AIS estimate of the log partition function for the 50 test values of L. This gives a smooth approximation for all L that are greater than or equal to all individual estimates. An example of this final approximation can be seen in Fig. 3.3.

3.4 Conclusion

We motivated the need for a more flexible distribution than the multinomial such as the Poisson MRF. However, the PMRF distribution has several complications due to its normalization that hinder it from being a general-purpose model for count data. We overcome these difficulties by restricting a fixed-length domain as in a multinomial while retaining the parametric form of the Poisson MRF. By parameterizing by the length of the document, we can then efficiently compute sampling-based estimates of the log partition function and hence the likelihood—which were not previously developed for the PMRF model. In general, we suggest that the LPMRF model could open up new avenues of research

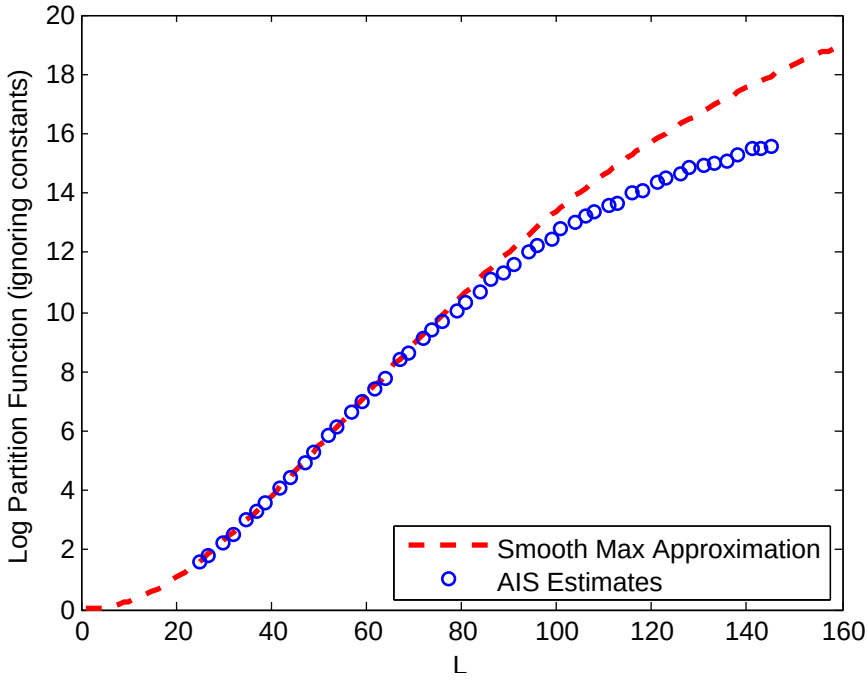


Figure 3.3: Example of log partition estimation for all values of L.

where the multinomial distribution is currently used.

Chapter 4

Square Root Graphical Model[1]

4.1 Abstract

We develop Square Root Graphical Models (SQR), a novel class of parametric graphical models that provides multivariate generalizations of univariate exponential family distributions. Previous multivariate graphical models [Yang et al., 2015] did not allow positive dependencies for the exponential and Poisson generalizations. However, in many real-world datasets, variables clearly have positive dependencies. For example, the airport delay time in New York—modeled as an exponential distribution—is positively related to the delay time in Boston. With this motivation, we give an example of our model class derived from the univariate exponential distribution that allows for almost arbitrary positive and negative dependencies with only a mild condition on the parameter matrix—a condition akin to the positive definiteness of the Gaussian covariance matrix. Our Poisson generalization allows for both positive and negative dependencies without any constraints on the parameter values. We also develop parameter estimation methods using node-wise regressions with ℓ_1 regularization and likelihood approximation methods using sampling. Finally, we demonstrate our exponential generalization on a synthetic dataset and a real-world dataset of airport delay times.

[1]The majority of this chapter is from [Inouye et al., 2016a] with some edits for better integration into this dissertation. [Inouye et al., 2016a] was primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

4.2 Square Root Graphical Model

We remind the reader of the form of the previous graphical model class described in Chapter 2 [Yang et al., 2015]:

$$\begin{aligned}\mathbb{P}(\boldsymbol{x}|\theta,\Phi) = \exp\Big(&\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T \Phi T(\boldsymbol{x}) \\ &+ \textstyle\sum_{s=1}^{p} B(x_s) - A(\boldsymbol{\theta},\Phi)\Big)\end{aligned} \tag{4.1}$$

$$\begin{aligned}A(\boldsymbol{\theta},\Phi) = \textstyle\int_{\mathcal{D}} \exp\Big(&\boldsymbol{\theta}^T T(\boldsymbol{x}) + T(\boldsymbol{x})^T \Phi T(\boldsymbol{x}) \\ &+ \textstyle\sum_{s=1}^{p} B(x_s)\Big)\mathrm{d}\mu(\boldsymbol{x})\,,\end{aligned} \tag{4.2}$$

where $A(\boldsymbol{\theta},\Phi)$ is the log partition function (i.e. log normalization constant) which is required for probability normalization, $\Phi \in \mathbb{R}^{p\times p}$ is symmetric with zeros along the diagonal and μ is either the standard Lebesgue measure or the counting measure depending on whether the domain $\mathcal{D}$ is continuous or discrete.

The amazingly simple yet helpful change from the previous graphical model class [Yang et al., 2015] is that we take the square root of the sufficient statistics in the interaction term. Essentially, this makes the interaction term linear in the sufficient statistics $O(T(x))$ rather than quadratic $O(T(x)^2)$ as in Eqn. 4.1. This change avoids the problem of the quadratic term overcoming the other terms while allowing both positive and negative dependencies. More formally, given any univariate exponential family with nonnegative sufficient statistics $T(x) \geq 0$, we can define the Square Root Graphical Model (SQR) class as follows:

$$\begin{aligned}\mathbb{P}(\boldsymbol{x}\,|\,\boldsymbol{\theta},\Phi) = \exp\Big(&\boldsymbol{\theta}^T\sqrt{T(\boldsymbol{x})} + \sqrt{T(\boldsymbol{x})}^T \Phi \sqrt{T(\boldsymbol{x})} \\ &+ \textstyle\sum_s B(x_s) - A(\Phi)\Big)\end{aligned} \tag{4.3}$$

$$\begin{aligned}A(\boldsymbol{\theta},\Phi) = \textstyle\int_{\mathcal{D}} \exp\Big(&\boldsymbol{\theta}^T\sqrt{T(\boldsymbol{x})} + \sqrt{T(\boldsymbol{x})}^T \Phi \sqrt{T(\boldsymbol{x})} \\ &+ \textstyle\sum_s B(x_s)\Big)\mathrm{d}\mu(\boldsymbol{x})\,,\end{aligned} \tag{4.4}$$

where $\sqrt{T(\boldsymbol{x})}$ is an entry-wise square root except when $T(x) = x^2$ in which case $\sqrt{T(x)} \equiv x$.[2] Figure 4.1 shows examples of the exponential and Poisson SQR distributions for no dependency, positive dependency and negative dependency. If $\boldsymbol{\theta} = 0$ and Φ is a diagonal matrix, then we recover an independent joint distribution so the SQR class of models can be seen as a direct relaxation of the independence assumption, similar to previous graphical models. In the next sections, we analyze some of the properties of SQR models including their conditional distributions.

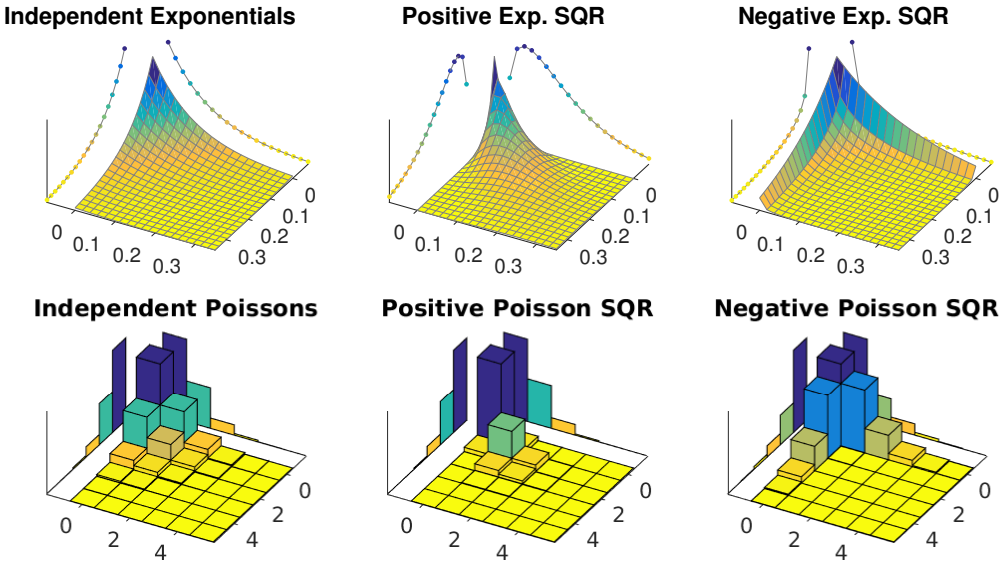


Figure 4.1: These examples of 2D exponential SQR and Poisson SQR distributions with no dependency (i.e. independent), positive dependency and negative dependency show the amazing flexibility of the SQR model class that can intuitively model *positive and negative* dependencies while having a simple parametric form. The approximate 1D marginals are shown along the edges of the plots.

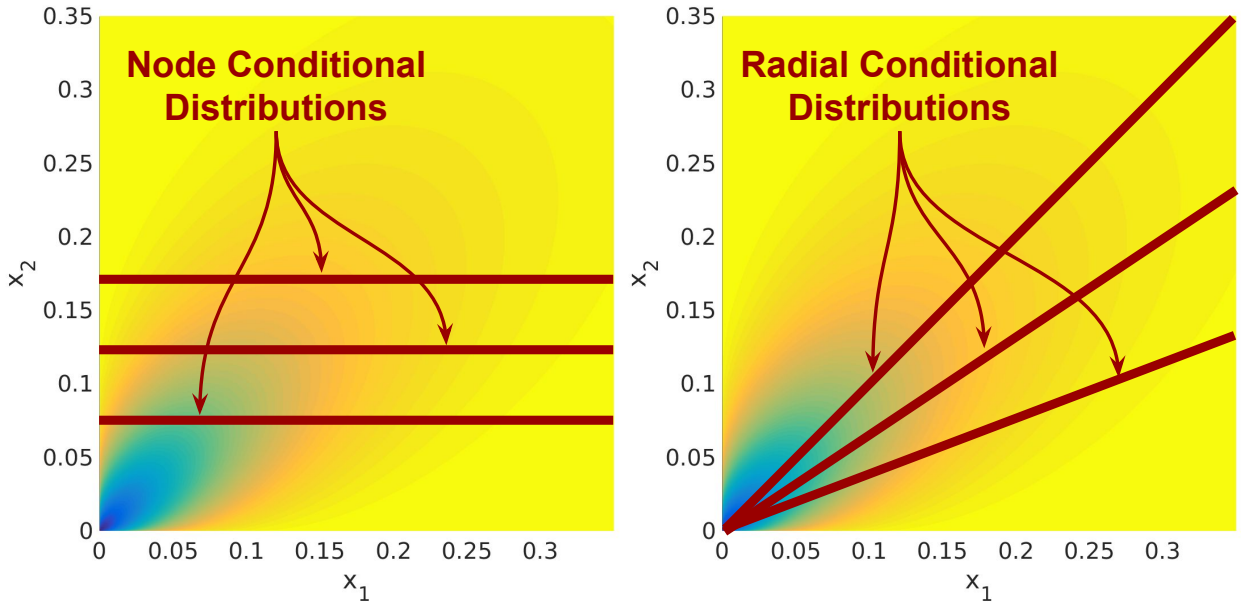


Figure 4.2: *Node* conditional distributions (left) are univariate probability distributions of one variable assuming the other variables are given while *radial* conditional distributions are univariate probability distributions of vector scaling assuming the vector direction is given. Both conditional distributions are helpful in understanding SQR graphical models.

4.2.1 SQR Conditional Distributions

We analyze two types of univariate conditional distributions of the SQR graphical models. The first is the standard *node* conditional distribution, i.e. the conditional distribution of one variable given the values for all other variables (see Fig. 4.2). The second is what we will call the *radial* conditional distribution in which the *unit direction* is fixed but the length of the vector is unknown (see Fig. 4.2). The node conditional distribution is helpful for parameter estimation as described more fully in Sec. 4.2.3. The radial conditional distribution is important for understanding the form of the SQR distribution as well as providing a means to succinctly prove that the normalization constant is finite (i.e. the distribution is valid) as described in Sec. 4.2.2.

[2]This nuance is important for the Gaussian SQR in Sec. 4.3.

Node Conditional Distribution The probability distribution of one variable x_s given all other variables $\boldsymbol{x}_{-s} = [x_1, x_2, \ldots, x_{s-1}, x_{s+1}, \ldots, x_p]$ is as follows:

$$\mathbb{P}(x_s \,|\, \boldsymbol{x}_{-s}, \boldsymbol{\theta}, \Phi) \propto \\ \exp\Big\{\phi_{ss} T(x_s) + \Big(\theta_s + 2\boldsymbol{\phi}_{-s}^T \sqrt{T(\boldsymbol{x}_{-s})}\Big)\sqrt{T(x_s)} + B(x_s)\Big\},$$

where $\boldsymbol{\phi}_{-s} \in \mathbb{R}^{p-1}$ is the s-th column of Φ with the s-th entry removed. This conditional distribution can be reformulated as a new two parameter exponential family:

$$\mathbb{P}(x_s \,|\, \boldsymbol{x}_{-s}, \boldsymbol{\theta}, \Phi) = \\ \exp\Big(\eta_1 \tilde{T}_1(x_s) + \eta_2 \tilde{T}_2(x_s) + B(x_s) - A_{\text{node}}(\boldsymbol{\eta})\Big) \tag{4.5}$$

$$A_{\text{node}}(\boldsymbol{\eta}) = \\ \int_{\mathcal{D}} \exp\Big(\eta_1 \tilde{T}_1(x_s) + \eta_2 \tilde{T}_2(x_s) + B(x_s)\Big) \mathrm{d}\mu(x_s)\,, \tag{4.6}$$

where $\eta_1 = \phi_{ss}$, $\eta_2 = \theta_s + 2\boldsymbol{\phi}_{-s}^T \sqrt{T(\boldsymbol{x}_{-s})}$, $\tilde{T}_1(x) = T(x)$, and $\tilde{T}_2(x) = \sqrt{T(x)}$. Note that this reduces to the base exponential family only if $\eta_2 = 0$ unlike the model in Eqn. 4.1 which, by construction, has node conditionals in the base exponential family. Examples of node conditional distributions for the exponential and Poisson SQR can be seen in Fig. 4.3. While these node conditionals are different from the base exponential family and hence slightly more difficult to use for parameter estimation as described later in Sec. 4.2.3, the benefit of almost arbitrary positive and negative dependencies significantly outweighs the cost of using SQR over previous graphical models.

Radial Conditional Distribution For simplicity, let us assume w.l.o.g. that $T(\boldsymbol{x}) = \boldsymbol{x}$.[3] Suppose we condition on the unit direction $\mathbf{v} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1}$ of the sufficient statistics but the scaling

[3] If T is not linear than we can merely reparameterize the distribution so that this is the case.

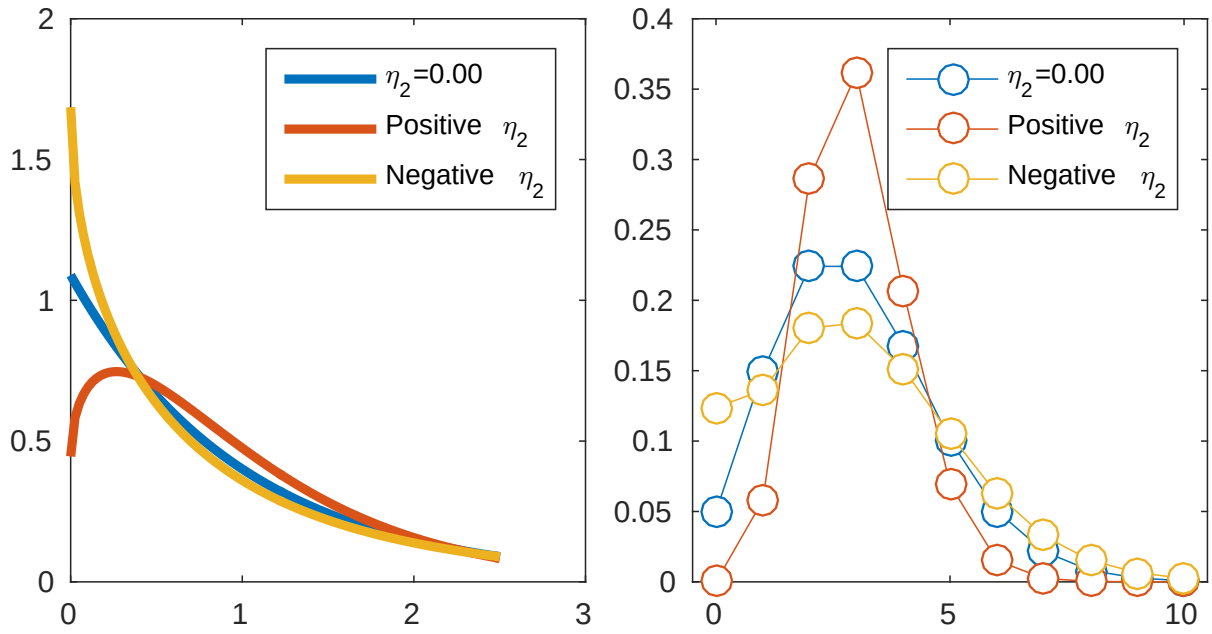


Figure 4.3: Examples of the node conditional distributions of exponential (left) and Poisson (right) SQR models for $\eta_2 = 0$, $\eta_2 > 0$ and $\eta_2 < 0$.

of this unit direction $z = \|\boldsymbol{x}\|_1$ is unknown. We call this the *radial* conditional distribution:

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x} = z\mathbf{v} \,|\, \mathbf{v}, \boldsymbol{\theta}, \Phi) \\
&\propto \exp\left(\boldsymbol{\theta}^T\sqrt{z\mathbf{v}} + \sqrt{z\mathbf{v}}^T\,\Phi\,\sqrt{z\mathbf{v}} + \textstyle\sum_s B(zv_s)\right) \\
&\propto \exp\left((\boldsymbol{\theta}^T\sqrt{\mathbf{v}})\sqrt{z} + \left(\sqrt{\mathbf{v}}^T\,\Phi\,\sqrt{\mathbf{v}}\right)z + \textstyle\sum_s B(zv_s)\right).
\end{aligned}$$

The radial conditional distribution can be rewritten as a univariate exponential family:

$$\mathbb{P}(z \,|\, \mathbf{v}, \boldsymbol{\theta}, \Phi) = \exp\left(\underbrace{\bar{\eta}_1 z + \bar{\eta}_2\sqrt{z}}_{O(z)} + \underbrace{\tilde{B}_{\mathbf{v}}(z)}_{O(B(z))} - A_{\text{rad}}(\bar{\boldsymbol{\eta}})\right) \tag{4.7}$$

$$A_{\text{rad}}(\bar{\boldsymbol{\eta}}) = \int_{\mathcal{D}} \exp\left(\underbrace{\bar{\eta}_1 z + \bar{\eta}_2\sqrt{z}}_{O(z)} + \underbrace{\tilde{B}_{\mathbf{v}}(z)}_{O(B(z))}\right)\mathrm{d}\mu(z)\,, \tag{4.8}$$

where $\bar{\eta}_1 = \sqrt{\mathbf{v}}^T\Phi\sqrt{\mathbf{v}}$, $\bar{\eta}_2 = \boldsymbol{\theta}^T\sqrt{\mathbf{v}}$ and $\tilde{B}_{\mathbf{v}}(z) = \sum_s B(zv_s)$. Note that if the log base measure of the base exponential family is zero $B(x) = 0$, then the radial conditional is the

same as the node conditional distribution because the modified base measure is also zero $\tilde{B}_{\mathbf{v}}(z) = 0$. If both $\boldsymbol{\theta} = 0$ and $B(x) = 0$, this actually reduces to the base exponential family. For example, the exponential distribution has $B(x) = 0$, and thus if we set $\boldsymbol{\theta} = 0$, the radial conditional of an exponential SQR is merely the exponential distribution. Other examples with a log base measure of zero include the Beta distribution and the gamma distribution with a known shape. For distributions in which the log base measure is not zero, the distribution will deviate from the node conditional distribution based on the relative difference between $B(x)$ and $\tilde{B}_{\mathbf{v}}(x)$. However, the important point even for distributions with non-zero log base measures is that the terms in the exponent grow at the same rate as the base exponential family—i.e. $O(z) + O(B(z))$. This helps to ensure that the radial conditional distribution is normalizable even as $z \to \infty$ since the base exponential family was normalizable. As an example, the Poisson distribution has the log base measure $B(x) = -\log(x!)$ and thus $\tilde{B}_{\mathbf{v}}(x)$ is $O(-x\log x)$ whereas the other terms $\bar{\eta}_1 z + \bar{\eta}_2\sqrt{z}$ are only $O(z)$. This provides the intuition of why the Poisson SQR radial distribution is normalizable as will be explained in Sec. 4.3.2.

4.2.2 Normalization

Normalization of the distribution was the reason for the negative-only parameter restrictions of the exponential and Poisson distributions in the previous graphical models [Besag, 1974, Yang et al., 2015] as defined in Eqn. 4.1. However, we show that in the case of SQR models, normalization is much simpler to achieve and generally puts little to no restriction on the value of the parameters—thus allowing both positive and negative dependencies. For our derivations, let $\mathcal{V} = \{\mathbf{v} : \|\mathbf{v}\|_1 = 1, \mathbf{v} \in \mathbb{R}^p_+\}$ be the set of unit vectors in the positive orthant. The SQR log partition function $A(\Phi)$ can be decomposed into nested

integrals over the unit direction and the one dimensional integral over scaling, denoted z:

$$A(\boldsymbol{\theta}, \Phi) = \log \int_{\mathcal{V}} \int_{\mathcal{Z}(\mathbf{v})} \exp\Big(\boldsymbol{\theta}^T \sqrt{z\mathbf{v}} + \sqrt{z\mathbf{v}}^T \Phi \sqrt{z\mathbf{v}} + \sum_s B(z v_s) \Big) \mathrm{d}\mu(z)\, \mathrm{d}\mathbf{v} \tag{4.9}$$

$$= \log \int_{\mathcal{V}} \int_{\mathcal{Z}(\mathbf{v})} \exp(\bar{\eta}_1(\mathbf{v}) z + \bar{\eta}_2(\mathbf{v}) \sqrt{z} + \sum_s B(z v_s)) \mathrm{d}\mu(z)\, \mathrm{d}\mathbf{v}, \tag{4.10}$$

where $\mathcal{Z}(\mathbf{v}) = \{z \in \mathbb{R}_+ : z\mathbf{v} \in \mathcal{D}\}$, and μ and $\mathcal{D}$ are defined as in Eqn. 4.2. Because $\mathcal{V}$ is bounded, we merely need that the radial conditional distribution is normalizable (i.e. $A_{\text{rad}}(\bar{\boldsymbol{\eta}}) < \infty$ from Eqn. 4.8) for the joint distribution to be normalizable. As suggested in Sec. 4.2.1, the radial conditional distribution is similar to the base exponential family and thus likely only has similar restrictions on parameter values as the base exponential family. In Sec. 4.3, we give examples for the exponential SQR and Poisson SQR distributions showing that this condition can be achieved with little or no restriction on the parameter values.

4.2.3 Parameter Estimation

For estimating the parameters Φ and $\boldsymbol{\theta}$, we follow the basic approach of [Ravikumar et al., 2010, Yang et al., 2015, 2013] and fit p ℓ_1-regularized node-wise regressions using the node conditional distributions described in Sec. 4.2.1. Thus, given a data matrix $X \in \mathbb{R}^{p \times n}$ we attempt to optimize the following convex function:

$$\arg\min_{\Phi} -\tfrac{1}{n} \textstyle\sum_s \sum_i \Big(\eta_{1si} x_{si} + \eta_{2si} \sqrt{x_{si}} + B(x_{si}) - A_{\text{node}}(\eta_{1si}, \eta_{2si}) \Big) + \lambda \|\Phi\|_{1,\text{off}}\,, \tag{4.11}$$

where $\eta_{1si} = \phi_{s,s}$, $\eta_{2si} = \theta + 2\boldsymbol{\phi}_{-s}^T \sqrt{T(\boldsymbol{x}_{-si})}$, $\|\Phi\|_{1,\text{off}} = \sum_{s \neq t} |\phi_{st}|$ is the ℓ_1-norm on the off diagonal elements and λ is a regularization parameter. Note that this can be trivially parallelized into p independent sub problems which allows for significantly faster computation

as in [Inouye et al., 2015]. Unlike previous graphical models [Yang et al., 2015] that were known to have closed-form solutions to the node conditional log partition function, the main difficulty for SQR graphical models is that the node conditional log partition function $A_{\text{node}}(\boldsymbol{\eta})$ is not known to have a closed form in general.

For the particular case of exponential SQR models, there is a closed-form solution for A_{node} using the error function as will be seen in Sec. 4.3.1 on exponential SQR models. More generally, because A_{node} is merely a one dimensional summation or integral, standard numerical approximations such as Gaussian quadrature could be used. Similarly, the gradient of ∇A_{node} could be numerically approximated by:

$$\nabla A_{\text{node}} = \tfrac{1}{\epsilon}\Big[\big(\hat{A}(\eta_1+\epsilon,\eta_2)-\hat{A}(\eta_1,\eta_2)\big), \\ \big(\hat{A}(\eta_1,\eta_2+\epsilon)-\hat{A}(\eta_1,\eta_2)\big)\Big], \tag{4.12}$$

where ϵ is a small step such as 0.001. Notice that to compute the function value and the gradient, only three 1D numerical integrations are needed. Another significant speedup that could be explored in future work would be to use a Newton-like method as in [Hsieh et al., 2014, Inouye et al., 2015], which optimize a quadratic approximation around the current iterate. Because these Newton-like methods only need a small number of Newton iterations to converge, the number of numerical integrations could be reduced significantly compared to gradient descent which often require thousands of iterations to converge.

4.2.4 Likelihood Approximation

We use Annealed Importance Sampling (AIS) [Neal, 2001] similar to the sampling used in [Inouye et al., 2015] for likelihood approximation. In particular, we need to approximate the SQR log partition function $A(\boldsymbol{\theta}, \Phi)$ as in Eqn. 4.4. First, we derive a slice sample for the node conditionals in which the bounds for the slice can be computed in closed form. Second, we use the slice sampler to develop a Gibbs sampler for SQR models. Finally, we derive an annealed importance sampler [Neal, 2001] using the Gibbs sampler as

the intermediate sampler by linearly combining the off-diagonal part of the parameter matrix Φ_{off} with the diagonal part Φ_{diag}—i.e. $\tilde{\Phi} = \gamma\Phi_{\text{off}} + \Phi_{\text{diag}}$. We also modify $\tilde{\boldsymbol{\theta}} = \gamma\boldsymbol{\theta}$ similarly. For each successive distribution, we linearly change γ from 0 to 1. Thus, we start by sampling from the base exponential family independent distribution $\prod_{s=1}^{p} \mathbb{P}(\boldsymbol{x} \,|\, \eta_{1s} = \phi_{ss}, \eta_{2s} = 0)$ and slowly move towards the final SQR distribution $\mathbb{P}(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \Phi)$. We maintain the sample weights as defined in [Neal, 2001] and from these weights, we can compute an approximation to the log partition function [Neal, 2001].

4.3 Examples from Various Exponential Families

We give several examples of SQR graphical models in the following sections (however, it should be noted that we have been developing a *class* of graphical models for *any* univariate exponential family with nonnegative sufficient statistics). The main analysis for each case is determining what conditions on the parameter matrix Φ allow the joint distribution to be normalized. As described in Sec. 4.2.2, for SQR models, this merely reduces to determining when the radial conditional distribution is normalizable. We analyze the exponential and Poisson cases in later sections but first we give examples of the discrete and Gaussian SQR graphical models.

The discrete SQR graphical model—including the binary Ising model—is equivalent to the standard discrete graphical model because the sufficient statistics are indicator functions $T_s(x) = \boldsymbol{I}(x = s), \forall s \neq p$ and the square root of an indicator function is merely the indicator function. Thus, in the discrete case, the discrete graphical model in [Ravikumar et al., 2010, Yang et al., 2015] is equivalent to the discrete SQR graphical model. For the Gaussian distribution, we can use the nonnegative Gaussian sufficient statistic $T(x) = x^2$. Thus, the Gaussian SQR graphical model is merely $\mathbb{P}(\boldsymbol{x}|\Phi) \propto \exp(\boldsymbol{\theta}^T\boldsymbol{x} + \boldsymbol{x}^T\Phi\boldsymbol{x})$, which by inspection is clearly the standard Gaussian distribution where $\boldsymbol{\theta} = \Sigma^{-1}\mu$ and $\Phi = -\frac{1}{2}\Sigma^{-1}$ is required

to be negative definite.[4] Thus, the Gaussian graphical model can be seen as a special case of SQR graphical models.

4.3.1 Exponential SQR Graphical Model

We consider what are the required conditions on the parameters $\boldsymbol{\theta}$ and Φ for the exponential SQR graphical model. If $\bar{\eta}_1$ is positive, the log partition function will diverge because even the end point $\lim_{z\to\infty} \exp(\bar{\eta}_1 z) \to \infty$. On the other hand, if $\bar{\eta}_1$ is negative, then the radial conditional distribution is similar in form to the exponential distribution and thus the log partition function will be finite because the negative linear term $\bar{\eta}_1 z$ dominates in the exponent as $z \to \infty$.[5] See appendix for proof. Thus, the basic condition on Φ is:

$$\Phi_{\text{Exp}} \in \{\Phi : \sqrt{\mathbf{v}}^T \Phi \sqrt{\mathbf{v}} < 0, \forall \mathbf{v} \in \mathcal{V}\}. \tag{4.13}$$

Note that this allows both positive and negative dependencies. A sufficient condition is that Φ be negative definite—as is the case for Gaussian graphical models. However, negative definiteness is far from necessary because we only need negativity of the interaction term for vectors in the positive orthant. It may even be possible for Φ to positive definite but Eqn. 4.13 be satisfied; however, we have not explored this idea.

For fitting the SQR model, the node conditional log partition function $A_{\text{Exp}}(\boldsymbol{\eta})$ has a closed-form solution:

$$A_{\text{Exp}}(\boldsymbol{\eta}) = \log\left(\frac{\sqrt{\pi}\eta_1 \exp\left(\frac{-\eta_2^2}{4\eta_1}\right)\left(1-\text{erf}\left(\frac{-\eta_2}{2\sqrt{-\eta_1}}\right)\right)}{-2(-\eta_1)^{\frac{3}{2}}} - \frac{1}{\eta_1}\right),$$

[4] This is by the slightly nuanced definition of the square root operator in Eqn. 4.3 and 4.4 such that $\sqrt{x^2} \equiv x$ rather than $|x|$.

[5] On the edge case when $\bar{\eta}_1 = 0$, the log partition function will diverge if $\bar{\eta}_2 \geq 0$ and will converge if $\bar{\eta}_1 < 0$ by simple arguments. The normalizability condition when $\eta_2 = 0$ could slightly loosen the condition on Φ in Eqn. 4.13 but for simplicity we did not include this edge case.

where $\text{erf}(\cdot)$ is the error function. The erf function shows up because of an initial substitution of $u = \sqrt{x}$ to transform the exponent into a quadratic form. Note that $\eta_1 < 0$ by the condition on Φ_{Exp} in Eqn. 4.13 above. The derivatives of A_{Exp} can also be computed in closed form for use in the parameter estimation algorithm.

4.3.2 Poisson SQR Graphical Model

The normalization analysis for Poisson SQR graphical model is also relatively simple but requires a more careful analysis than the exponential SQR graphical model. Let us consider the form of the Poisson radial conditional: $\mathbb{P}_{\text{rad}}(z \,|\, \mathbf{v}) \propto \exp(\bar{\eta}_1 z + \bar{\eta}_2 \sqrt{z} - \sum_s \log((zv_s)!))$. Note that the domain of z, denoted $\mathcal{D}_z = \{z \in \mathbb{Z}_+ : z\mathbf{v} \in \mathbb{Z}_+^p\}$, is discrete. We can simplify the analysis by taking a larger domain $\tilde{\mathcal{D}}_z = \{z \in \mathbb{Z}_+\}$ of all non-negative integers and changing the log factorial to the smooth gamma function, i.e. $\sum_s \log((zv_s)!)) \rightarrow \sum_s \log(\Gamma(zv_s + 1))$. Thus, the radial conditional log partition function is upper bounded by:

$$\sum_{z \in \mathbb{Z}_+} \exp\Big(\underbrace{\bar{\eta}_1 z + \bar{\eta}_2 \sqrt{z}}_{O(z)} - \underbrace{\textstyle\sum_s \log(\Gamma(zv_s + 1))}_{O(z \log z)} \Big) < \infty. \tag{4.14}$$

The basic intuition is that the exponent has a linear $O(z)$ term minus an $O(z\log z)$ term, which will eventually overcome the linear term and hence the summation will converge. Note that we did not assume any restrictions on Φ except that all the entries are finite. Thus, for the Poisson distribution, Φ can have arbitrary positive and negative dependencies. A formal proof for Eqn. 4.14 is given in the appendix.

4.4 Experiments and Results

4.4.1 Synthetic Experiment

In order to show that our parameter estimation algorithm has the ability to find the correct dependencies, we develop a synthetic experiment on chain-like graphs. We construct

Φ to be a k-dependent circular chain-like graph by first setting the diagonal of Φ to be 1. Then, we add an edge between each node and its k neighbors with a value of $\frac{0.9}{2*k}$, i.e. the s-th node is connected to the $(s+1)$-th, $(s+2)$-th, ..., $(s+k)$-th nodes where the indices are modulo p (e.g. $k = 1$ is the standard chain graph). This ensures that Φ is negative definite by the Gershgorin disc theorem. We generate samples using Gibbs sampling with 1000 Gibbs iterations per sample and 10 slice samples for each node conditional sample. For this experiment, we set $p = 30, \lambda = 10^{-5}, k \in \{1, 2, 3, 4\}$, and $n \in \{100, 200, 400, 800, 1600\}$. We calculate the edge precision for the fitted model by computing the precision for the top kp edges—i.e. the number of true edges in the top kp estimated edges over the total number of true edges. The results in Fig. 4.4 demonstrate that our parameter estimation algorithm is able to easily find the edges for small k and is even able to identify the edges for large k, though the problem becomes more difficult when k is large (because there are more parameters, which are also smaller), and thus more samples are needed. With 1,600 samples, our parameter estimation algorithm is able to recover at least 95% of the edges even when $k = 4$.

4.4.2 Airport Delay Times Experiment

In order to demonstrate that the SQR graphical model class is more suitable for real-world data than the graphical models in [Yang et al., 2015] (which can only model *negative* dependencies), we fit an exponential SQR model to a dataset of airport delay times at the top 30 commercial USA airports—also known as Large Hub airports. We gathered flight data from the US Department of Transportation public "On-Time: On-Time Performance" database[6] for the year 2014. We calculated the average delay time per day at each of the top 30 airports (excluding cancellations).

For our implementation, we set $\lambda \in \{0.05, 0.005, 0.0005\}$ and set a maximum of 5000

[6] http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID =236&DB_Short_Name=On-Time

iterations for our proximal gradient descent algorithm. For approximating the log partition function using the AIS sampling defined in Sec. 4.2.4, we sampled 1000 AIS samples with 100 annealing distributions—i.e. γ took 100 values between 0 and 1—, 10 Gibbs steps per annealed distribution and 10 slice samples for every node conditional sampling. Generally, our algorithm with these parameter settings took roughly 35 seconds to train the model and about 25 seconds to compute the likelihood (i.e. AIS sampling) using MATLAB prototype code on the TACC Maverick cluster.[7]

We computed the geometric mean of the relative log likelihood compared to the independent exponential model, i.e. $\exp((\mathcal{L}_{\text{SQR}} - \mathcal{L}_{\text{Ind}})/n)$, where $\mathcal{L}$ is the log likelihood. These values can be seen in Fig. 4.4 (higher is better). Clearly, the exponential SQR model provides a major improvement in relative likelihood over the independent model suggesting that the delay times of airports are clearly related to one another. In Fig. 4.5, we visualize the non-zeros of Φ—which correspond to the edges in the graphical model—to show that our model is capturing intuitive positive dependencies.

First, it should be noted that all the dependencies are positive yet positive dependencies were not allowed by previous graphical models [Yang et al., 2015]! Second, as would be expected because of weather delays, the airports in the Chicago area seems to affect the delays of many other airports. Similarly, a weather effect seems to be evident for the airports near New York City. Third, as would be expected, some dependencies seem to be geographic in nature as seen by the west coast dependencies, Texas dependency (i.e. DFW-IAH), and east coast dependencies. Note that the geographic dependencies were found even though no location data was given to the algorithm. Fourth, the busiest airport in Atlanta, GA (ATL) is not strongly dependent on other airports. This seems reasonable because Atlanta rarely has snow and there are few major airports geographically close to

[7] https://portal.tacc.utexas.edu/user-guides/maverick

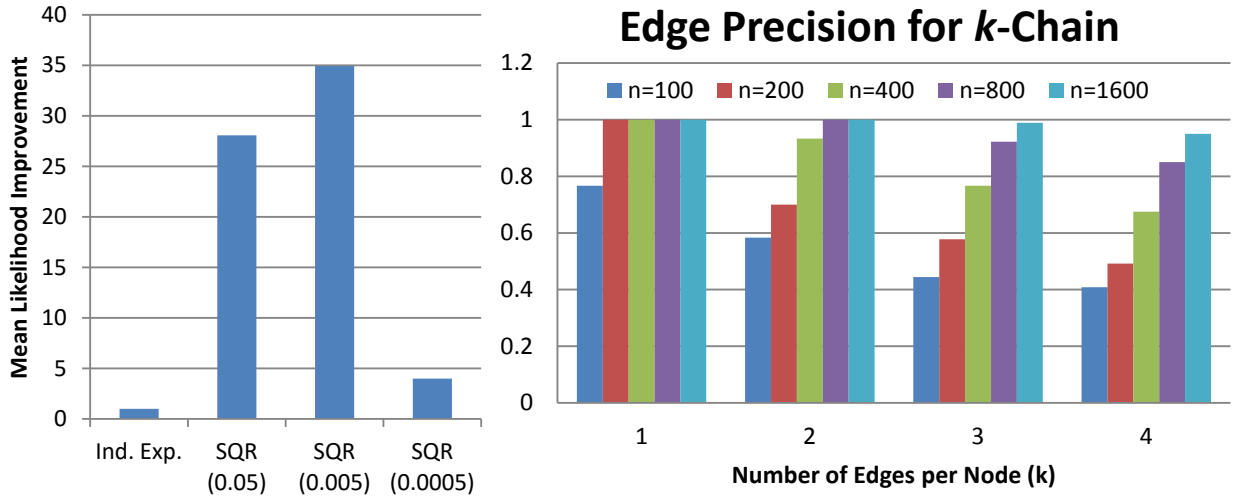


Figure 4.4: (Left) The fitted exponential SQR model improves significantly over the independent exponential model in terms of relative likelihood suggesting that a model with positive dependencies is more appropriate. (Right) The edge precision for the circular chain graph described in Sec. 4.4.1 demonstrate that our parameter estimation algorithm is able to effectively identify edges for small k, and if given enough samples, can also identify edges for larger k.

Atlanta. These qualitative results suggest that the exponential SQR model is able to capture multiple interesting and intuitive dependencies.

4.5 Discussion

As full probability models, SQR graphical models could be used in any situation where a multivariate distribution is required. For example, SQR models could be used in Bayesian classification by modeling the probability of each class distribution instead of the classical Naive Bayes assumption of independence. As another example, SQR models could be used as the base distribution in mixtures or admixture composite distributions as in [Inouye et al., 2014b,a]—similar to multivariate Gaussian mixture models. Another extension would be to consider mixed SQR graphical models in which the joint distribution has variables using

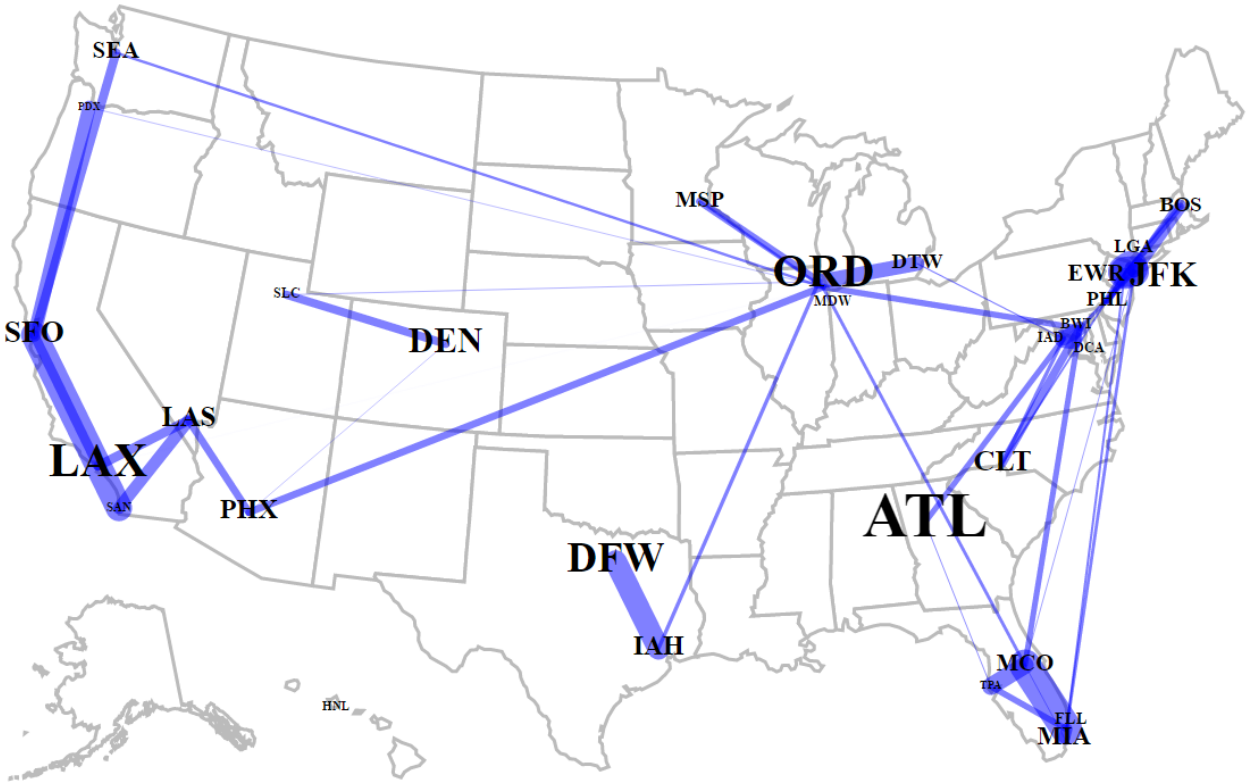


Figure 4.5: Visualizing the edges between airports shows that SQR models can capture interesting and intuitive *positive* dependencies even though previous exponential graphical models [Yang et al., 2015] were restricted to negative dependencies. The delays at the Chicago airports seem to greatly affect other airports as would be expected because of Chicago weather delays. Other dependencies are likely related to weather or geography. (For this visualization, we set $\lambda = 0.0005$. Width of lines is proportional to the value of the edge weight, i.e. a non-zero in Φ, and the size of airport abbreviation is proportional to the average number of passengers.)

different exponential families as base distributions as explored for previous graphical models in [Yang et al., 2014a, Tansey et al., 2015].

Comparison to Fixed-Length Poisson MRF In Chapter 3 Inouye et al. [2015], we proposed a graphical model variant called Fixed-Length Poisson MRF (LPMRF) that

modifies the domain of the distribution assuming the length of the vector $L = \|\boldsymbol{x}\|_1$ is fixed, i.e. $\mathcal{D} = \{\boldsymbol{x} \in \mathbb{Z}_+^p : \|\boldsymbol{x}\|_1 = L\}$. Because the domain is finite as in TPGM, the distribution is normalizable even with positive dependencies. However, as with TPGM, the quadratic term in the parametric form dominates the distribution if L is large, and thus Inouye et al. [2015] modify the distribution by introducing a weighting function that decreases the quadratic term as L increases. The previous Poisson graphical models (LPMRF, PGM, TPGM, SPGM, QPGM) attempt to deal with the quadratic interaction term in different ways but all of them significantly change the distribution/domain and often require the specification of new unintuitive hyperparameters to allow for positive dependencies. Also, according to the authors' best knowledge, no variants of the exponential graphical model have been proposed to allow for positive dependencies. Therefore, we propose a novel graphical model class that alleviates the problem with the quadratic interaction term and provides both exponential and Poisson graphical models that allow *positive and negative* dependencies.

4.6 Conclusion

We introduce a novel class of graphical models that creates multivariate generalizations for *any* univariate exponential family with nonnegative sufficient statistics—including Gaussian, discrete, exponential and Poisson distributions. We show that SQR graphical models generally have few restrictions on the parameters and thus can model *both positive and negative* dependencies unlike previous generalized graphical models as represented by [Yang et al., 2015]. In particular, for the exponential SQR model, the parameter matrix Φ can have both positive and negative dependencies and is only constrained by a mild condition—akin to the positive-definiteness condition on Gaussian covariance matrices. For the Poisson distribution, there are no restrictions on the parameter values, and thus the Poisson SQR model allows for arbitrary positive and

negative dependencies. We develop parameter estimation and likelihood approximation methods and demonstrate that the SQR model indeed captures interesting and intuitive dependencies by modeling both synthetic datasets and a real-world dataset of airport delays. The general SQR class of distributions opens the way for graphical models to be effectively used with non-Gaussian and non-discrete data without the unintuitive restriction to negative dependencies.

Chapter 5

A Review of Multivariate Distributions for Count Data Derived from the Poisson Distribution[1]

5.1 Abstract

The Poisson distribution has been widely studied and used for modeling univariate count-valued data. Multivariate generalizations of the Poisson distribution that permit dependencies, however, have been far less popular. Yet, real-world high-dimensional count-valued data found in word counts, genomics, and crime statistics, for example, exhibit rich dependencies, and motivate the need for multivariate distributions that can appropriately model this data. We review multivariate distributions derived from the univariate Poisson, categorizing these models into three main classes: 1) where the marginal distributions are Poisson, 2) where the joint distribution is a mixture of independent multivariate Poisson distributions, and 3) where the node-conditional distributions are derived from the Poisson. We discuss the development of multiple instances of these classes and compare the models in terms of interpretability and theory. Then, we empirically compare multiple models from each class on three real-world datasets that have varying data characteristics from different domains, namely traffic

[1]Most of this chapter was published in [Inouye et al., 2017], which was a collaborative effort with Eunho Yang, Genevera I. Allen and Pradeep Ravikumar. David Inouye drafted the experimental and comparison sections, executed all the experiments, rewrote and expanded the copula section, added summaries for major sections, and edited the whole paper for content and continuity. Genevera I. Allen originally drafted the copula section of this paper. Eunho Yang originally drafted the introduction, graphical model section and mixture model section. All authors contributed to editing and revising.

accident data, biological next generation sequencing data, and text data. These empirical experiments develop intuition about the comparative advantages and disadvantages of each class of multivariate distribution that was derived from the Poisson. Finally, we suggest new research directions as explored in the subsequent discussion section.

5.2 Introduction

The classical model for a count-valued random variable is the univariate Poisson distribution, whose probability mass function for $x \in \{0, 1, 2, \dots\}$ is:

$$\mathbb{P}_{\text{Poiss}}(x \mid \lambda) = \lambda^x \exp(-\lambda)/x!\,, \tag{5.1}$$

where λ is the standard mean parameter for the Poisson distribution. A trivial extension of this to a multivariate distribution would be to assume independence between variables, and take the product of node-wise univariate Poisson distributions, but such a model would be ill-suited for many examples of multivariate count-valued data that require rich dependence structures. We review multivariate probability models that are derived from the univariate Poisson distribution and permit non-trivial dependencies between variables. We categorize these models into three main classes based on their primary modeling assumption. The first class assumes that the univariate marginal distributions are derived from the Poisson. The second class is derived as a mixture of independent multivariate Poisson distributions. The third class assumes that the univariate conditional distributions are derived from the Poisson distribution—this last class of models can also be studied in the context of probabilistic graphical models. An illustration of each of these three main model classes can be seen in Fig. 5.1. While these models might have been classified by primary application area or performance on a particular task, a classification based on modeling assumptions helps emphasize the core abstractions for each model class. In addition, this categorization may help practitioners from different disciplines learn from the models that have worked well

in different areas. We discuss multiple instances of these classes in the later sections and highlight the strengths and weaknesses of each class. We then provide a short discussion on the differences between classes in terms of interpretability and theory. Using two different empirical measures, we empirically compare multiple models from each class on three real-world datasets that have varying data characteristics from different domains, namely traffic accident data, biological next generation sequencing data, and text data. These experiments develop intuition about the comparative advantages and disadvantages of the models and suggest new research directions as explored in the subsequent discussion section.

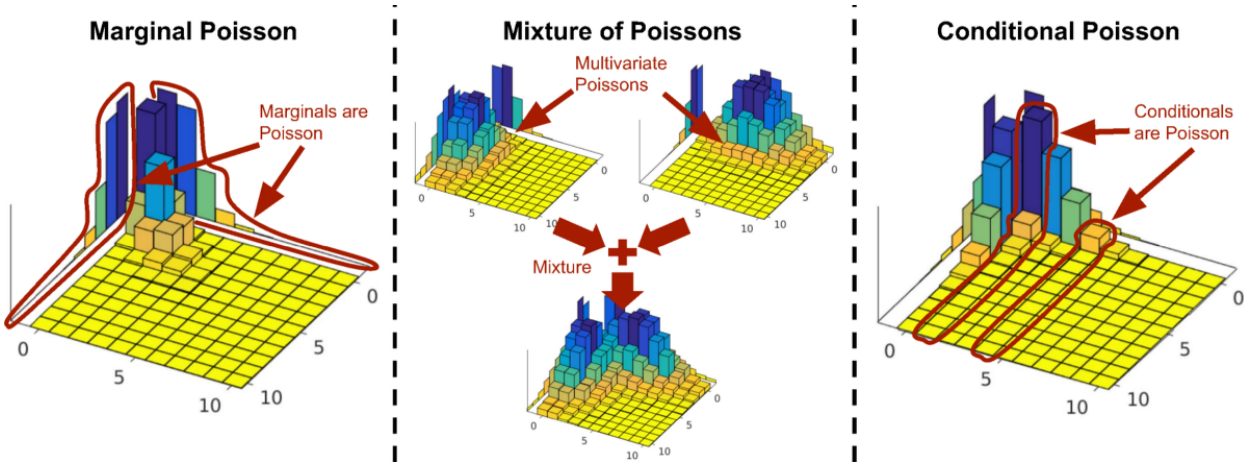


Figure 5.1: (Left) The first class of Poisson generalizations is based on the assumption that the univariate marginals are derived from the Poisson. (Middle) The second class is based on the idea of mixing independent multivariate Poissons into a joint multivariate distribution. (Right) The third class is based on the assumption that the univariate conditional distributions are derived from the Poisson.

5.2.0.1 Notation

$\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_+$ denotes the *nonnegative* real numbers, and $\mathbb{R}_{++}$ denotes the *positive* real numbers. Similarly, $\mathbb{Z}$ denotes the set of integers. Matrices

are denoted as capital letters (e.g. X, Φ), vectors are denoted as boldface lowercase letters (e.g. $\boldsymbol{x}, \boldsymbol{\phi}$) and scalar values are non-bold lowercase letters (e.g. x, ϕ).

5.3 Marginal Poisson Generalizations

The models in this section generalize the univariate Poisson to a multivariate distribution with the property that the marginal distributions of each variable are Poisson. This is analogous to the marginal property of the multivariate Gaussian distribution, since the marginal distributions of a multivariate Gaussian are univariate Gaussian, and thus seems like a natural constraint when extending the univariate Poisson to the multivariate case. Several historical attempts at achieving this marginal property have incidentally developed the same class of models, with different derivations [M'Kendrick, 1925, Campbell, 1934, Wicksell, 1916, Teicher, 1954]. This marginal Poisson property can also be achieved via the more general framework of copulas [Xue-Kun Song, 2000, Nikoloulopoulos and Karlis, 2009, Nikoloulopoulos, 2013a].

5.3.1 Multivariate Poisson Distribution

The formulation of the multivariate Poisson[2] distribution goes back to M'Kendrick [1925] where authors use differential equations to derive the bivariate Poisson process. An equivalent but more readable interpretation to arrive at the bivariate Poisson distribution would be to use the summation of independent Poisson variables, as follows [Campbell, 1934]: Let y_1, y_2 and z be univariate Poisson variables with parameters λ_1, λ_2 and λ_0 respectively. Then by setting $x_1 = y_1 + z$ and $x_2 = y_2 + z$, (x_1, x_2) follows the bivariate

[2]The label "multivariate Poisson" was introduced in the statistics community to refer to the particular model introduced in this section but other generalizations could also be considered multivariate Poisson distributions.

Poisson distribution, and its joint probability mass is defined as:

$$\begin{aligned}&\mathbb{P}_{\text{BiPoi}}(x_1, x_2 \,|\, \lambda_1, \lambda_2, \lambda_0) \\ =\ &\exp(-\lambda_1 - \lambda_2 - \lambda_0)\frac{\lambda_1^{x_1}}{x_1!}\frac{\lambda_2^{x_2}}{{}_2!}\sum_{z=0}^{\min(x_1,x_2)}\binom{x_1}{z}\binom{x_2}{z}z!\left(\frac{\lambda_0}{\lambda_1\lambda_2}\right)^z. \end{aligned} \tag{5.2}$$

Since the sum of independent Poissons is also Poisson (whose parameter is the sum of those of two components), the marginal distribution of x_1 (similarly x_2) is still a Poisson with the rate of $\lambda_1 + \lambda_0$. It can be easily seen that the covariance of x_1 and x_2 is λ_0 and as a result the correlation coefficient is somewhere between 0 and $\min\{\frac{\sqrt{\lambda_1+\lambda_0}}{\sqrt{\lambda_2+\lambda_0}}, \frac{\sqrt{\lambda_2+\lambda_0}}{\sqrt{\lambda_1+\lambda_0}}\}$ [Holgate, 1964]. Independently, Wicksell [1916] derived the bivariate Poisson as the limit of a bivariate binomial distribution. Campbell [1934] show that the models in M'Kendrick [1925] and Wicksell [1916] can identically be derived from the sums of 3 independent Poisson variables.

This approach to directly extend the Poisson distribution can be generalized further to handle the multivariate case $\boldsymbol{x} \in \mathbb{Z}_+^p$, in which each variable x_i is the sum of individual Poisson y_i and the common Poisson x_0 as before. The joint probability for a Multivariate Poisson is developed in Teicher [1954] and further considered by other works [Dwass and Teicher, 1957, Srivastava and Srivastava, 1970, Wang, 1974, Kawamura, 1979]:

$$\mathbb{P}_{\text{MulPoi}}(\boldsymbol{x}; \boldsymbol{\lambda}) = \exp\Big(-\sum_{i=0}^{p}\lambda_i\Big)\Big(\prod_{i=1}^{p}\frac{\lambda_i^{x_i}}{x_i!}\Big)\sum_{z=0}^{\min_i x_i}\Big(\prod_{i=1}^{p}\binom{x_i}{z}\Big)z!\left(\frac{\lambda_0}{\prod_{i=1}^{p}\lambda_i}\right)^z. \tag{5.3}$$

Several have shown that this formulation of the multivariate Poisson can also be derived as a limiting distribution of a multivariate binomial distribution when the success probabilities are small and the number of trials is large [Krishnamoorthy, 1951, Krummenauer, 1998, Johnson et al., 1997]. As in the bivariate case, the marginal distribution of x_i is Poisson with parameter $\lambda_i + \lambda_0$. Since λ_0 controls the covariance between *all* variables, an extremely limited set of correlations between variables is permitted.

Mahamunulu [1967] first proposed a more general extension of the multivariate Poisson distribution that permits a full covariance structure. This distribution has been studied further by many [Loukas and Kemp, 1983, Kano and Kawamura, 1991, Johnson et al., 1997, Karlis, 2003, Tsiamyrtzis and Karlis, 2004]. While the form of this general multivariate Poisson distribution is too complicated to spell out for $p > 3$, its distribution can be specified by a multivariate reduction scheme. Specifically, let y_i for $i = 1, \ldots, (2^p - 1)$ be independently Poisson distributed with parameter λ_i. Now, define $\mathbf{A} = [A_1, A_2, \ldots A_p]$ where A_i is a $d \times \binom{p}{i}$ matrix consisting of ones and zeros where each column of A_i has exactly i ones with no duplicate columns. Hence, A_1 is the $p \times p$ identity matrix and A_p is a column vector of all ones. Then, $\boldsymbol{x} = \mathbf{A}\boldsymbol{y}$ is a p-dimensional multivariate Poisson distributed random vector with a full covariance structure. Note that the simpler multivariate Poisson distribution with constant covariance in Eq. 5.3 is a special case of this general form where $\mathbf{A} = [A_1, A_p]$.

The multivariate Poisson distribution has not been widely used for real data applications. This is likely due to two major limitations of this distribution. First, the multivariate Poisson distribution only permits *positive* dependencies; this can easily be seen as the distribution arises as the sum of independent Poisson random variables and hence covariances are governed by the positive rate parameters λ_i. The assumption of positive dependencies is likely unrealistic for most real count-valued data examples. Second, computation of probabilities and inference of parameters is especially cumbersome for the multivariate Poisson distribution; these are only computationally tractable for small p and hence not readily applicable in high-dimensional settings. Kano and Kawamura [1991] proposed multivariate recursion schemes for computing probabilities, but these schemes are only stable and computationally feasible for small p, thus complicating likelihood-based inference procedures. Karlis [2003] more recently proposed a latent variable based EM algorithm for parameter inference of the general multivariate Poisson

distribution. This approach treats every pairwise interaction as a latent variable and conducts inference over both the observed and hidden parameters. While this method is more tractable than recursion schemes, it still requires inference over $\binom{d}{2}$ latent variables and is hence not feasible in high-dimensional settings. Overall, the multivariate Poisson distribution introduced above is appealing in that its marginal distributions are Poisson; yet, there are many modeling drawbacks including severe restriction on the types of dependencies permitted (e.g. only positive relationships), a complicated and intractable form in high-dimensions, and challenging inference procedures.

5.3.2 Copula Approaches

A much more general way to construct valid multivariate Poisson distributions with Poisson marginals is by pairing a *copula* distribution with Poisson marginal distributions. For continuous multivariate distributions, the use of copula distributions is founded on the celebrated Sklar's theorem: any continuous joint distribution can be decomposed into a copula and the marginal distributions, and conversely, any combination of a copula and marginal distributions gives a valid continuous joint distribution [Sklar, 1959]. The key advantage of such models for continuous distributions is that copulas fully specify the dependence structure hence separating the modeling of marginal distributions from the modeling of dependencies. While copula distributions paired with continuous marginal distributions enjoy wide popularity (see for example [Cherubini et al., 2004a] in finance applications), copula models paired with discrete marginal distributions, such as the Poisson, are more challenging both for theoretical and computational reasons [Genest and Nešlehová, 2007, Nikoloulopoulos, 2013b, 2016]. However, several simplifications and recent advances have attempted to overcome these challenges [Rüschendorf, 2013, Nikoloulopoulos, 2013b, 2016].

5.3.2.1 Copula Definition and Examples

A copula is defined by a joint cumulative distribution function (CDF), $C(\boldsymbol{u})\colon [0,1]^p \rightarrow [0,1]$ with uniform marginal distributions. As a concrete example, the Gaussian copula (see left subfigure of Fig. 5.2 for an example) is derived from the multivariate normal distribution and is one of the most popular multivariate copulas because of its flexibility in the multidimensional case; the Gaussian copula is defined simply as:

$$C_R^{\text{Gauss}}(u_1, u_2, \cdots, u_p) = H_R\left(H^{-1}(u_1), \cdots, H^{-1}(u_p)\right),$$

where $H^{-1}(\cdot)$ denotes the standard normal inverse cumulative distribution function, and $H_R(\cdot)$ denotes the joint cumulative distribution function of a $\mathcal{N}(0, R)$ random vector, where R is a correlation matrix. A similar multivariate copula can be derived from the multivariate Student's t distribution if extreme values are important to model [Demarta and McNeil, 2005].

The Archimedean copulas are another family of copulas which have a *single* parameter that defines the global dependence between all variables [Trivedi and Zimmer, 2005]. One property of Archimedean copulas is that they admit an explicit form unlike the Gaussian copula. Unfortunately, the Archimedean copulas do not directly allow for a rich dependence structure like the Gaussian because they only have one dependence parameter rather than a parameter for each pair of variables.

Pair copula constructions (PCCs) [Aas et al., 2009] for copulas, or vine copulas, allow combinations of different bivariate copulas to form a joint multivariate copula. PCCs define multivariate copulas that have an expressive dependency structure like the Gaussian copula but may also model asymmetric or tail dependencies available in Archimedean and t copulas. Pair copulas only use univariate CDFs, conditional CDFs, and bivariate copulas to construct a multivariate copula distribution and hence can use combinations of the Archimedean

copulas described previously. The multivariate distributions can be factorized in a variety of ways using bivariate copulas to flexibly model dependencies. *Vines*, or graphical tree-like structures, denote the possible factorizations that are feasible for PCCs [Bedford and Cooke, 2002].

5.3.2.2 Copula Models for Discrete Data

As per Sklar's theorem, any copula distribution can be combined with marginal distribution CDFs $\{F_i(x_i)\}_{i=1}^d$ to create a joint distribution:

$$G(x_1, x_2, \cdots, x_p \,|\, \theta, F_1, \cdots, F_p) = C_\theta \left(u_1 = F_1(x_1), \cdots, u_p = F_p(x_p)\right).$$

If sampling from the given copula is possible, this form admits simple direct sampling from the joint distribution (defined by the CDF $G(\cdot)$) by first sampling from the copula $\boldsymbol{u} \sim \text{Copula}(\theta)$ and then transforming $\boldsymbol{u}$ to the target space using the inverse CDFs of the marginal distributions: $\boldsymbol{x} = [F_1^{-1}(u_1), \cdots, F_p^{-1}(u_p)]$.

A valid multivariate discrete joint distribution can be derived by pairing a copula distribution with Poisson marginal distributions. For example, a valid joint CDF with Poisson marginals is given by

$$G(x_1, x_2, \cdots, x_p \,|\, \theta) = C_\theta \left(F_1(x_1 \,|\, \lambda_1), \cdots, F_p(x_p \,|\, \lambda_p)\right),$$

where $F_i(x_i \,|\, \lambda_i)$ is the Poisson cumulative distribution function with mean parameter λ_i, and θ denotes the copula parameters. If we pair a Gaussian copula with Poisson marginal distributions, we create a valid joint distribution that has been widely used for generating samples of multivariate count data [Xue-Kun Song, 2000, Yahav and Shmueli, 2012, Cook et al., 2010]—an example of the Gaussian copula paired with Poisson marginals to form a discrete joint distribution can be seen in Fig. 5.2.

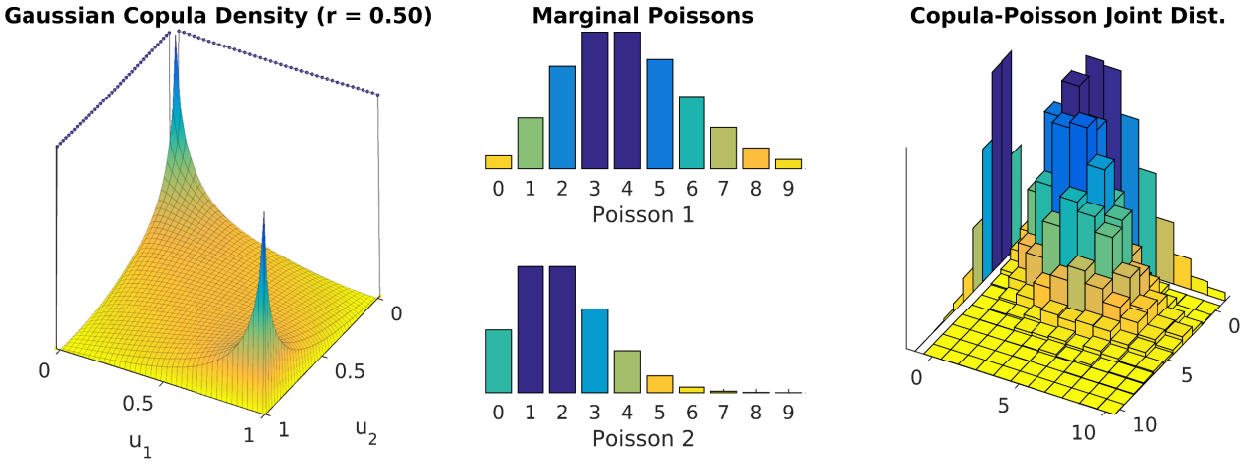


Figure 5.2: A copula distribution (left)—which is defined over the unit hypercube and has uniform marginal distributions—, paired with univariate Poisson marginal distributions for each variable (middle) defines a valid discrete joint distribution with Poisson marginals (right).

Nikoloulopoulos [2013a] present an excellent survey of copulas to be paired with discrete marginals by defining several desired properties of a copula (quoted from [Nikoloulopoulos, 2013a]):

1. Wide range of dependence, allowing both positive and negative dependence.
2. Flexible dependence, meaning that the number of bivariate marginals is (approximately) equal to the number of dependence parameters.
3. Computationally feasible cumulative distribution function (CDF) for likelihood estimation.
4. Closure property under marginalization, meaning that lower-order marginals belong to the same parametric family.
5. No joint constraints for the dependence parameters, meaning that the use of covariate functions for the dependence parameters is straightforward.

Each copula model satisfies some of these properties but not all of them. For example,

Gaussian copulas satisfy properties (1), (2) and (4) but not (3) or (5) because the normal CDF is not known in closed form and the positive definiteness constraint on the correlation matrix. Nikoloulopoulos [2013a] recommend Gaussian copulas for general models and vine copulas if modeling dependence in the tails or asymmetry is needed.

5.3.2.3 Theoretical Properties of Copulas Derived from Discrete Distributions

From a theoretical perspective, a multivariate discrete distribution can be viewed as a continuous copula distribution paired with discrete marginals but the derived copula distributions are not unique and hence, are unidentifiable [Genest and Nešlehová, 2007]. Note that this is in contrast to continuous multivariate distributions where the derived copulas are uniquely defined [Sklar, 1973]. Because of this non-uniqueness property, Genest and Nešlehová [2007] caution against performing inference on and interpreting dependencies of copulas derived from discrete distributions. A further consequence of non-uniqueness is that when copula distributions are paired with discrete marginal distributions, the copulas no longer fully specify the dependence structure as with continuous marginals [Genest and Nešlehová, 2007]. In other words, the dependencies of the joint distribution will depend in part on which marginal distributions are employed. In practice, this often means that the range of dependencies permitted with certain copula and discrete marginal distribution pairs is much more limited than the copula distribution would otherwise model. However, several have suggested that this non-uniqueness property does not have major practical ramifications [Nikoloulopoulos, 2013a, Karlis, 2016].

We discuss a few common approaches used for the estimation of continuous copulas with discrete marginals.

5.3.2.4 Continuous Extension for Parameter Estimation

For estimation of continuous copulas from data, a two-stage procedure called Inference Function for Marginals (IFM) [Joe and Xu, 1996] is commonly used in which the marginal distributions are estimated first and then used to map the data onto the unit hypercube using the CDFs of the inferred marginal distributions. While this is straightforward for continuous marginals, this procedure is less obvious for discrete marginal distributions when using a continuous copula. One idea is to use the continuous extension (CE) of integer variables to the continuous domain [Denuit and Lambert, 2005] by forming a new "jitter" continuous random variable $\tilde{x}$:

$$\tilde{x} = x + (u - 1),$$

where u is a random variable defined on the unit interval. It is straightforward to see that this new random variable is continuous and $\tilde{x} \leq x$. An obvious choice for the distribution of u is the uniform distribution. With this idea, inference can be performed using a surrogate likelihood by randomly projecting each discrete data point into the continuous domain and averaging over the random projections as done in [Heinen and Rengifo, 2007, 2008]. Madsen [2009], Madsen and Fang [2011] use the CE idea as well but generate *multiple* jittered samples $\{\tilde{x}^{(1)}, \tilde{x}^{(1)}, \ldots, \tilde{x}^{(m)}\}$ for each original observation x to estimate the discrete likelihood rather than merely generating one jittered sample $\tilde{x}$ for each original observation x as in [Heinen and Rengifo, 2007, 2008]. Nikoloulopoulos [2013b] find that CE-based methods significantly underestimate the correlation structure because the CE jitter transform operates independently for each variable instead of considering the correlation structure between the variables.

5.3.2.5 Distributional Transform for Parameter Estimation

In a somewhat different direction, Rüschendorf [2013] proposed the use of a generalization of the CDF distribution function $F(\cdot)$ for the case with discrete variables,

which they term a *distributional transform* (DT) denoted by $\tilde{F}(\cdot)$:

$$\tilde{F}(x, v) \equiv F(x) + v\mathbb{P}(x) = \mathbb{P}(X < x) + v\mathbb{P}(X = x)\,,$$

where $v \sim \text{Uniform}(0, 1)$. Note that in the continuous case, $\mathbb{P}(X = x) = 0$ and thus this reduces to the standard CDF for continuous distributions. One way of thinking of this modified CDF is that the random variable v adds a random jump when there are discontinuities in the original CDF. If the distribution is discrete (or more generally if there are discontinuities in the original CDF), this transformation enables a simple proof of a theorem akin to Sklar's theorem for discrete distributions [Rüschendorf, 2013].

Kazianka and Pilz [2010], Kazianka [2013] propose using the distributional transform (DT) from [Rüschendorf, 2013] to develop a simple and intuitive approximation for the likelihood. Essentially, they simply take the expected jump value of $\mathbb{E}(v) = 0.5$ (where $v \sim \text{Uniform}(0, 1)$) and thus transform the discrete data to the continuous domain by the following:

$$u_i \equiv F_i(x_i - 1) + 0.5\mathbb{P}(x_i) = 0.5(F_i(x_i - 1) + F_i(x_i))\,,$$

which can be seen as simply taking the average of the CDF values at $x_i - 1$ and x_i. Then, they use a continuous copula such as the Gaussian copula. Note that this is much simpler to compute than the simulated likelihood (SL) method in [Nikoloulopoulos, 2013b] or the continuous extension (CE) methods in [Heinen and Rengifo, 2007, 2008, Madsen, 2009, Madsen and Fang, 2011], which require averaging over many different random initializations.

5.3.2.6 Simulated Likelihood for Parameter Estimation

Finally, [Nikoloulopoulos, 2013b] propose a method to directly approximate the maximum likelihood estimate by estimating a discretized Gaussian copula. Essentially,

unlike the CE and DT methods which attempt to transform discrete variables to continuous variables, the MLE for a Gaussian copula with discrete marginal distributions $F_1, F_2, \ldots, F_p$ can be formulated as estimating multivariate normal rectangular probabilities:

$$\mathbb{P}(\boldsymbol{x} \,|\, \boldsymbol{\gamma}, R) = \int_{\phi^{-1}[F_1(x_1-1\,|\,\gamma_1)]}^{\phi^{-1}[F_1(x_1\,|\,\gamma_1)]} \cdots \int_{\phi^{-1}[F_1(x_p-1\,|\,\gamma_p)]}^{\phi^{-1}[F_p(x_p\,|\,\gamma_p)]} \Phi_R(z_1, \ldots, z_p) \mathrm{d}z_1 \ldots \mathrm{d}z_p \,, \tag{5.4}$$

where $\boldsymbol{\gamma}$ are the marginal distribution parameters, $\phi^{-1}(\cdot)$ is the univariate standard normal inverse CDF, and $\Phi_R(\cdots)$ is the multivariate normal density with correlation matrix R. Nikoloulopoulos [2013b] propose to approximate the multivariate normal rectangular probabilities via fast simulation algorithms discussed in [Genz and Bretz, 2009]. Because this method directly approximates the MLE via simulated algorithms, this method is called simulated likelihood (SL). Nikoloulopoulos [2016] compare the DT and SL methods for small sample sizes and find that the DT method tends to overestimate the correlation structure. However, because of the computational simplicity, Nikoloulopoulos [2016] give some heuristics of when the DT method might work well compared to the more accurate but more computationally expensive SL method.

5.3.2.7 Vine Copulas for Discrete Distributions

Panagiotelis et al. [2012] provide conditions under which a multivariate discrete distribution can be decomposed as a vine PCC copula paired with discrete marginals. In addition, Panagiotelis et al. [2012] show that likelihood computation for vine PCCs with discrete marginals is quadratic as opposed to exponential as would be the case for general multivariate copulas such as the Gaussian copula with discrete marginals. However, computation in truly high-dimensional settings remains a challenge as $2d(d-1)$ bivariate copula evaluations are required to calculate the PMF or likelihood of a d-variate PCC using the algorithm proposed by Panagiotelis et al. [2012]. These bivariate copula

evaluations, however, can be coupled with some of the previously discussed computational techniques such as continuous extensions, distributional transforms, and simulated likelihoods for further computational improvements. Finally, while vine PCCs offer a very flexible modeling approach, this comes with the added challenge of selecting the vine construction and bivariate copulas [Czado et al., 2013], which has not been well studied for discrete distributions. Overall, Nikoloulopoulos [2013a] recommend using vine PCCs for complex modeling of discrete data with tail dependencies and asymmetric dependencies.

5.3.3 Summary of Marginal Poisson Generalizations

We have reviewed the historical development of the multivariate Poisson which has Poisson marginals and then reviewed many of the recent developments of using the much more general copula framework to derive Poisson generalizations with Poisson marginals. The original multivariate Poisson models based on latent Poisson variables are limited to positive dependencies and require computationally expensive algorithms to fit. However, estimation of copula distributions paired with Poisson marginals—while theoretically has some caveats—can be performed efficiently in practice. Simple approximations such as the expectation under the distributional transformation can provide nearly trivial transformations that move the discrete variables to the continuous domain in which all the tools of continuous copulas can be exploited. More complex transformations such as the simulated likelihood method [Nikoloulopoulos, 2013b] can be used if the sample size is small or high accuracy is needed.

5.4 Poisson Mixture Generalizations

Instead of directly extending univariate Poissons to the multivariate case, a separate line of work proposes to indirectly extend the Poisson based on the mixture of independent Poissons. Mixture models are often considered to provide more flexibility by allowing the

parameter to vary according to a mixing distribution. One important property of mixture models is that they can model *overdispersion*. Overdispersion occurs when the variance of the data is larger than the mean of the data—unlike in a Poisson distribution in which the mean and variance are equal. One way of quantifying dispersion is the dispersion index:

$$\delta = \frac{\sigma^2}{\mu}\,. \tag{5.5}$$

If $\delta > 1$, then the distribution is overdispersed whereas if $\delta < 1$, then the distribution is underdispersed. In real world data as will be seen in the experimental section, overdispersion is more common than underdispersion. Mixture models also enable dependencies between the variables as will be described in the following paragraphs.

Suppose that we are modeling univariate random variable x with a density of $f(x \mid \theta)$. Rather than assuming θ is fixed, we let θ itself to be a random variable following some *mixing* distribution. More formally, a general *mixture* distribution can be defined as [Karlis and Xekalaki, 2005]:

$$\mathbb{P}(x \mid g(\cdot)) = \int_{\Theta} f(x \mid \theta)\, g(\theta)\, d\theta\,, \tag{5.6}$$

where the parameter θ is assumed to come from the mixing distribution $g(\theta)$ and Θ is the domain of θ.

For the Poisson case, let $\boldsymbol{\lambda} \in \mathbb{R}^p_{++}$ be a p-dimensional vector whose i-th element λ_i is the parameter of the Poisson distribution for x_i. Now, given some mixing distribution $g(\boldsymbol{\lambda})$, the family of Poisson mixture distributions is defined as

$$\mathbb{P}_{\text{MixedPoi}}(\boldsymbol{x}) = \int_{\mathbb{R}^d_{++}} g(\boldsymbol{\lambda}) \prod_{i=1}^{d} \mathbb{P}_{\text{Poiss}}(x_i \mid \lambda_i)\, d\boldsymbol{\lambda}\,, \tag{5.7}$$

where the domain of the joint distribution is any count-valued assignment (i.e. $x_i \in \mathbb{Z}_+, \forall i$). While the probability density function (Eq. 5.7) has the complicated form

involving a multidimensional integral (a complex, high-dimensional integral when p is large), the mean and variance are known to be expressed succinctly as

$$\mathbb{E}(\boldsymbol{x}) = \mathbb{E}(\boldsymbol{\lambda})\,, \tag{5.8}$$

$$\mathrm{Var}(\boldsymbol{x}) = \mathbb{E}(\boldsymbol{\lambda}) + \mathrm{Var}(\boldsymbol{\lambda})\,. \tag{5.9}$$

Note that Eq. 5.9 implies that the variance of a mixture is always larger than the variance of a single distribution. The higher order moments of $\boldsymbol{x}$ are also easily represented by those of $\boldsymbol{\lambda}$. Besides the moments, other interesting properties (convolutions, identifiability etc.) of Poisson mixture distributions are extensively reviewed and studied in Karlis and Xekalaki [2005].

One key benefit of Poisson mixtures is that they permit both positive as well as negative dependencies simply by properly defining $g(\boldsymbol{\lambda})$. The intuition behind these dependencies can be more clearly understood when we consider the sample generation process. Suppose that we have the distribution $g(\boldsymbol{\lambda})$ in two dimensions (i.e. $d = 2$) with a strong positive dependency between λ_1 and λ_2. Then, given a sample (λ_1, λ_2) from $g(\boldsymbol{\lambda})$, x_1 and x_2 are likely to also be positively correlated.

In an early application of the model, Arbous and Kerrich [1951] constrain the Poisson parameters as the different scales of common gamma variable λ: for $i = 1, \ldots, p$, the time interval t_i is given and λ_i is set to $t_i\lambda$. Hence, $g(\boldsymbol{\lambda})$ is a univariate gamma distribution specified by $\lambda \in \mathbb{R}_{++}$ —which only allows simple dependency structure. Steyn [1976], as another early attempt, choose the multivariate normal distribution for the mixing distribution $g(\boldsymbol{\lambda})$ to provide more flexibility on the correlation structure. However, the normal distribution poses problems because λ must reside in $\mathbb{R}_{++}$ while the the normal distribution is defined on $\mathbb{R}$.

One of the most popular choice for $g(\boldsymbol{\lambda})$ is the log-normal distribution thanks to its

rich covariance structure and natural positivity constraint[3]:

$$\mathcal{N}_{\log}(\boldsymbol{\lambda} \,|\, \boldsymbol{\mu}, \Sigma) = \frac{1}{\Pi_{i=1}^{d} \lambda_i \sqrt{(2\pi)^d |\Sigma|}} \exp\Big(-\frac{1}{2} (\log\boldsymbol{\lambda} - \boldsymbol{\mu})^\top \Sigma^{-1} (\log\boldsymbol{\lambda} - \boldsymbol{\mu}) \Big) \,. \tag{5.10}$$

The log-normal distribution above is parameterized by $\boldsymbol{\mu}$ and Σ, which are the mean and the covariance of $(\log\lambda_1, \log\lambda_2, \cdots, \log\lambda_d)$, respectively. Setting the random variable x_i to follow the Poisson distribution with parameter λ_i, we have the multivariate Poisson log-normal distribution [Aitchison and Ho, 1989] from Eq. 5.7:

$$\mathbb{P}_{\text{PoiLogN}}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \Sigma) = \int_{\mathbb{R}_+^d} \mathcal{N}_{\log}(\boldsymbol{\lambda} \,|\, \boldsymbol{\mu}, \Sigma) \prod_{i=1}^{d} \mathbb{P}_{\text{Poiss}}(x_i \,|\, \lambda_i) \, d\boldsymbol{\lambda} \,. \tag{5.11}$$

While the joint distribution (Eq. 5.11) does not have a closed-form expression and hence as d increases, it becomes computationally cumbersome to work with, its moments are available in closed-form as a special case of Eq. 5.9:

$$\begin{aligned} \alpha_i \equiv \mathbb{E}(x_i) &= \exp(\mu_i + \frac{1}{2}\sigma_{ii}) \,, \\ \text{Var}(x_i) &= \alpha_i + \alpha_i^2 \big(\exp(\sigma_{ii}) - 1 \big) \,, \\ \text{Cov}(x_i, x_j) &= \alpha_i \alpha_j (\exp(\sigma_{ij}) - 1) \,. \end{aligned} \tag{5.12}$$

The correlation and the degree of overdispersion (defined as the variance divided by the mean) of the marginal distributions are strictly coupled by α and σ. Also, the possible Spearman's ρ correlation values for this distribution are limited if the mean value α_i is small. To briefly explore this phenomena, we simulated a two-dimensional Poisson log-normal model with mean zero and covariance matrix:

$$\Sigma = 2\log(\alpha_i) \begin{bmatrix} 1 & \pm 0.999 \\ \pm 0.999 & 1 \end{bmatrix} ,$$

[3]This is because if $y \in \mathbb{R} \sim$ Normal, then $\exp(y) \in \mathbb{R}_{++} \sim$ LogNormal.

which corresponds to a mean value of α_i per Eq. 5.12 and the strongest positive and negative correlation possible between the two variables. We simulated one million samples from this distribution and found that when fixing $\alpha_i = 2$, the Spearman's ρ values are between -0.53 and 0.58. When fixing $\alpha_i = 10$, the Spearman's ρ values are between -0.73 and 0.81. Thus, for small mean values, the log-normal mixture is limited in modeling strong dependencies but for large mean values the log-normal mixture can model stronger dependencies. Besides the examples provided here, various Poisson mixture models from different mixing distributions are available although limited in the applied statistical literature due to their complexities. See Karlis and Xekalaki [2005] and the references therein for more examples of Poisson mixtures. Karlis and Xekalaki [2005] also provide the general properties of mixtures as well as the specific ones of Poisson mixtures such as moments, convolutions, and the posterior.

While this review focuses on modeling multivariate count-valued responses without any extra information, the several extensions of multivariate Poisson log-normal models have been proposed to provide more general correlation structures when covariates are available [Chib and Winkelmann, 2001, Ma et al., 2008, Park and Lord, 2007, El-Basyouny and Sayed, 2009, Aguero-Valverde and Jovanis, 2009, Zhan et al., 2015]. These works formulate the mean parameter of log-normal mixing distribution, $\log\mu_i$, as a linear model on given covariates in the Bayesian framework.

In order to alleviate the computational burden of using log-normal distributions as an infinite mixing density as above, Karlis and Meligkotsidou [2007] proposed an EM type estimation for a finite mixture of $k > 1$ Poisson distributions, which still preserves similar properties such as both positive and negative dependencies, as well as closed form moments. While [Karlis and Meligkotsidou, 2007] consider mixing multivariate Poissons with positive dependencies, the simplified form where the component distributions are independent Poisson distributions is much simpler to implement using an expectation-maximization (EM) algorithm. This simple finite mixture distribution can be

viewed as a middle ground between a single Poisson and a non-parametric estimation method where a Poisson is located at every training point—i.e. the number of mixtures is equal to the number of training data points ($k = n$).

The gamma distribution is another common mixing distribution for the Poisson because it is the conjugate distribution for the Poisson mean parameter λ. For the univariate case, if the mixing distribution is gamma, then the resulting univariate distribution is the well-known negative binomial distribution. The negative binomial distribution can handle overdispersion in count-valued data when the variance is larger than the mean. Unlike the Poisson log-normal mixture, the univariate gamma-Poisson mixture density—i.e. the negative binomial density—is known in closed form:

$$\mathbb{P}(x \,|\, r, p) = \frac{\Gamma(r+x)}{\Gamma(r)\Gamma(x+1)} p^r (1-p)^x \,.$$

As $r \to \infty$, the negative binomial distribution approaches the Poisson distribution. Thus, this can be seen as a generalization of the Poisson distribution. Note that the variance of this distribution is always larger than the Poisson distribution with the same mean value.

In a similar vein to using the gamma distribution, if instead of putting a prior on the Poisson mean parameter λ, we reparametrize the Poisson distribution by the log Poisson mean parameter $\theta = \log(\lambda)$, then the log-gamma distribution is conjugate to parameter θ. Bradley et al. [2015] recently leveraged the log-gamma conjugacy to the Poisson log-mean parameter θ by introducing the Poisson log-gamma hierarchical mixture distribution. In particular, they discuss the multivariate log-gamma distribution that can have flexible dependency structure similar to the multivariate log-normal distribution and illustrate some modeling advantages over the log-normal mixture model.

5.4.1 Summary of Mixture Model Generalizations

Overall, mixture models are particularly helpful if there is overdispersion in the data—which is often the case for real-world data as seen in the experiments section—while also allowing for variable dependencies to be modeled implicitly through the mixing distribution. If the data exhibits overdispersion, then the log-normal or log-gamma distributions [Bradley et al., 2015] give somewhat flexible dependency structures. The principal caveat with complex mixture of Poisson distributions is computational; exact inference of the parameters is typically computationally difficult due to the presence of latent mixing variables. However, simpler models such as the finite mixture using simple expectation maximization (EM) may provide good results in practice (see comparison section).

5.5 Conditional Poisson Generalizations

While the multivariate Poisson formulation in Eq. 5.3 as well as the distribution formed by pairing a copula with Poisson marginals assume that univariate *marginal distributions* are derived from the Poisson, the models in previous chapters of this dissertation generalizes the univariate Poisson by assuming the univariate *node-conditional distributions* are derived from the Poisson [Besag, 1974, Yang et al., 2012, 2013, 2015, Inouye et al., 2015, 2016a]. Like the assumption of Poisson marginals in previous sections, this conditional Poisson assumption seems a different yet natural extension of the univariate Poisson distribution. The multivariate Gaussian can be seen to satisfy such a conditional property since the node-conditional distributions of a multivariate Gaussian are univariate Gaussian. One benefit of these conditional models is that they can be seen as undirected graphical models or Markov Random Fields, and they have a simple parametric form. In addition, estimating these models generally reduces to estimating simple node-wise regressions, and some of these estimators have theoretical guarantees on

estimating the global graphical model structure even under high-dimensional sampling regimes, where the number of variables (p) is potentially even larger than the number of samples (n). For details about these graphical models, we refer the reader to Chapter 2 for information about previous graphical models including the Poisson graphical model (PGM), the truncated Poisson graphical model (TPGM), the sub-linear Poisson graphical model (SPGM) and the quadratic Poisson graphical model (QPGM). We refer the reader to Chapter 4 for details about the Poisson SQR graphical model and to Chapter 3 for details about the fixed-length Poisson MRF model (FLPGM).

5.5.1 Summary of Conditional Poisson Generalizations

The conditional Poisson models benefit from the rich literature in exponential families and undirected graphical models, or Markov Random Fields. In addition, the conditional Poisson models have a simple parametric form. The historical Poisson graphical model—or the auto-Poisson model [Besag, 1974])—only allowed negative dependencies between variables. Multiple extensions have sought to overcome this severe limitation by altering the Poisson graphical model so that the log partition function is finite even with positive dependencies. One major drawback to the graphical model approach is that computing the likelihood requires approximation of the joint log partition function $A(\boldsymbol{\theta}, \Phi)$; a related problem is that the distribution moments and marginals are not known in closed-form. Despite these drawbacks, parameter estimation using composite likelihood methods via ℓ_1-penalized node-wise regressions (in which the joint likelihood is not computed) has solid theoretical properties under certain conditions.

5.6 Model Comparison

We compare models by first discussing two structural aspects of the models: (a) interpretability and (b) the relative stringency and ease of verifying theoretical assumptions

and guarantees. We then present and discuss an empirical comparison of the models on three real-world datasets.

5.6.1 Comparison of Model Interpretation

Marginal models can be interpreted as weakly decoupling modeling marginal distributions over individual variables, from modeling the dependency structure over the variables. However, in the discrete case, specifically for distributions based on pairing copulas with Poisson marginals, the dependency structure estimation is also dependent on the marginal estimation, unlike for copulas paired with continuous marginals [Genest and Nešlehová, 2007]. **Conditional models** or graphical models, on the other hand, can be interpreted as specifying generative models for each variable given the variable's neighborhood (i.e. the conditional distribution). In addition, dependencies in graphical models can be visualized and interpreted via networks. Here, each variable is a node and the weighted edges in the network structure depict the pair-wise conditional dependencies between variables. The simple network depiction for graphical models may enable domain experts to interpret complex dependency structures more easily compared to other models. Overall, marginal models may be preferred if modeling the statistics of the data, particularly the marginal statistics over individual variables, is of primary importance, while conditional models may be preferred if prediction of some variables given others is of primary importance. **Mixture models** may be more or less difficult to interpret depending on whether there is an application-specific interpretation of the latent mixing variable. For example, a finite mixture of two Poisson distributions may model the crime statistics of a city that contains downtown and suburban areas. On the other hand, a finite mixture of fifty Poisson distributions or a log-normal Poisson mixture when modeling crash severity counts (as seen in the empirical comparison section) seems more difficult to interpret; even the model empirically well fits the data, the hidden mixture variable might

not have an obvious application-specific interpretation.

5.6.2 Comparison of Theoretical Considerations

Estimation of **marginal models** from data has various theoretical problems, as evidenced by the analysis of copulas paired with discrete marginals in [Genest and Nešlehová, 2007]. The extent to which these theoretical problems cause any significant practical issues remains unclear. In particular, the estimators of the marginal distributions themselves typically have easily checked assumptions since the empirical marginal distributions can be inspected directly. On the other hand, the estimation of **conditional models** is both computationally tractable, and comes with strong theoretical guarantees even under high-dimensional regimes where $n < p$ [Yang et al., 2015]. However the assumptions under which the guarantees of the estimators hold are difficult to check in practice, and could cause problems if they are violated (e.g. outliers caused by unobserved factors). Estimation of **mixture models** tend to have limited theoretical guarantees. In particular, finite Poisson mixture models have very weak assumptions on the underlying distribution—eventually becoming a non-parametric distribution if $k = O(n)$—but the estimation problems are likely NP-hard, with very few theoretical guarantees for practical estimators. Yet, empirically as seen in the next section, estimating a finite mixture model using Expectation-Maximization iterations performs well in practice.

5.6.3 Empirical Comparison

In this section, we seek to empirically compare models from the three classes presented to assess how well they fit real-world count data.

5.6.3.1 Comparison Experimental Setup

We empirically compare models on selected datasets from three diverse domains which have different data characteristics in terms of their mean count values and dispersion indices (Eq. 5.5) as can be seen in Table 5.1. The crash severity dataset is a small accident dataset from [Milton et al., 2008] with three different count variables corresponding to crash severity classes: "Property-only", "Possible Injury", and "Injury". The crash severity data exhibits high count values and high overdispersion. We retrieve raw next generation sequencing data for breast cancer (BRCA) using the software TCGA2STAT [Wan et al., 2016] and computed a simple log-count transformation of the raw counts: $\lfloor \log(x+1) \rfloor$, a common preprocessing technique for RNA-Seq data. The BRCA data exhibits medium counts and medium overdispersion. We collect the word count vectors from the Classic3 text corpus which contains abstracts from aerospace engineering, medical and information sciences journals.[4]. The Classic3 dataset exhibits low counts—including many zeros—and medium overdispersion. In the supplementary material, we also give results for a crime statistics dataset and the 20 Newsgroup dataset but they have similar characteristics and perform similarly to the the BRCA and Classic3 datasets respectively; thus, we omit them for simplicity. We select variables (e.g. for $p = 10$ or $p = 100$) by sorting the variables by mean count value—or sorting by variance in the case of the BRCA dataset as highly variable genes are of more interest in biology.

In order to understand how each model might perform under varying data characteristics, we consider the following two questions: (1) How well does the model (i.e. the joint distribution) fit the underlying data distribution? (2) How well does the model capture the dependency structure between variables? To help answer these questions, we evaluate the empirical fit of models using two metrics, which only require samples from the

[4] `http://ir.dcs.gla.ac.uk/resources/test_collections/`

Table 5.1: Dataset Statistics

(Per Variable ⇒)			Means			Dispersion Indices			Spearman's ρ		
Dataset	p	n	Min	Med	Max	Min	Med	Max	Min	Med	Max
Crash Severity	3	275	3.4	3.8	9.7	6	9.3	16	0.61	0.73	0.79
BRCA	10	878	3.2	5	7.7	1.5	2.2	3.8	-0.2	0.25	0.95
	100	878	1.1	4	9	0.63	1.7	4.6	-0.5	0.08	0.95
	1000	878	0.51	3.5	11	0.26	1	4.6	-0.64	0.06	0.97
Classic3	10	3893	0.26	0.33	0.51	1.4	3.4	3.8	-0.17	0.12	0.82
	100	3893	0.09	0.14	0.51	1.1	2.1	8.3	-0.17	0.02	0.82
	1000	3893	0.02	0.03	0.51	0.98	1.7	8.5	-0.17	-0	0.82

model. The first metric is based on a statistic called maximum mean discrepancy (MMD) [Gretton, 2012] which estimates the maximum moment difference over all possible moments. The empirical MMD can be approximated as follows from two sets of samples $X \in \mathbb{R}^{n_1 \times p}$ and $Y \in \mathbb{R}^{n_2 \times p}$:

$$\widehat{\mathrm{MMD}}(\mathcal{G}, X, Y) = \sup_{f \in \mathcal{G}} \frac{1}{n_1} \sum_{i=1}^{n_1} f(\boldsymbol{x}_i) - \frac{1}{n_2} \sum_{j=1}^{n_2} f(\boldsymbol{y}_j)\,, \tag{5.13}$$

where $\mathcal{G}$ is the union of the RKHS spaces based on the Gaussian kernel using twenty one σ values log-spaced between 0.01 and 100. In our experiments, we estimate the MMD between the pairwise marginals of model samples and the pairwise marginals of the original observations:

$$D_{st}^{\mathrm{MMD}} = \begin{cases} \widehat{\mathrm{MMD}}(\mathcal{G}, [\boldsymbol{x}^{(s)}], [\widehat{\boldsymbol{x}}^{(s)}]), & s = t \\ \widehat{\mathrm{MMD}}(\mathcal{G}, [\boldsymbol{x}^{(s)}, \boldsymbol{x}^{(t)}], [\widehat{\boldsymbol{x}}^{(s)}, \widehat{\boldsymbol{x}}^{(t)}]), & \text{otherwise} \end{cases}. \tag{5.14}$$

where $\boldsymbol{x}^{(s)}$ is the vector of data for the s-th variable of the true data and $\widehat{\boldsymbol{x}}^{(s)}$ is the vector of data for the s-th variable of samples from the estimated model—i.e. $\boldsymbol{x}^{(s)}$ are observations from the true underlying distribution and $\widehat{\boldsymbol{x}}^{(s)}$ are samples from the estimated model distribution. In our experiments, we use the fast approximation code for MMD from

[Zhao and Meng, 2015] with 2^6 number of basis vectors for the FastMMD approximation algorithm. The second metric merely computes the absolute difference between the pairwise Spearman's ρ values of model samples and the Spearman's ρ values of the original observations:

$$D^{\rho}_{st} = |\rho(\boldsymbol{x}^{(s)}, \boldsymbol{x}^{(t)}) - \rho(\widehat{\boldsymbol{x}}^{(s)}, \widehat{\boldsymbol{x}}^{(t)})|, \quad \forall s, t\,. \tag{5.15}$$

The MMD metric is of more general interest because it evaluates whether the models actually fit the empirical data distribution while the Spearman metric may be more interesting for practitioners who primarily care about the dependency structure, such as biologists who specifically want to study gene dependencies rather than gene distributions.

We empirically compare the model fits on these real-world data sets for several types of models from the three general classes presented. As a baseline, we estimate an independent Poisson model ("Ind Poisson"). We include Gaussian copulas and vine copulas both paired with Poisson marginals ("Copula Poisson" and "Vine Poisson") to represent the marginal model class. We estimate the copula-based models via the two-stage Inference Functions for Margins (IFM) method [Joe and Xu, 1996] via the distributional transform [Rüschendorf, 2013]. For the mixture class, we include both a simple finite mixture of independent Poissons ("Mixture Poiss") and a log-normal mixture of Poissons ("Log-Normal"). The finite mixture was estimated using a simple expectation-maximization (EM) algorithm; the log-normal mixture model was estimated via MCMC sampling using the code from [Zhan et al., 2015]. For the conditional model class, we estimate the simple Poisson graphical model ("PGM"), which only allows negative dependencies, and three variants that allow for positive dependencies: the truncated Poisson graphical model ("Truncated PGM"), the Fixed-Length Poisson graphical model with a Poisson distribution on the vector length $L = \|x\|_1$ ("FLPGM Poisson") and the Poisson square root graphical model ("Poisson SQR"). Using composite

likelihood methods of penalized ℓ_1 node-wise regressions, we estimate these models via code from [Yang et al., 2015], [Inouye et al., 2014b], [Inouye et al., 2016b] and the `XMRF`[5] R package. After parameter estimation, we generate 1,000 samples for each method using different types of sampling for each of the model classes.

To avoid overfitting to the data, we employ 3-fold cross-validation and report the average over the three folds. Because the conditional models (PGM, TPGM, FLPGM, and Poisson SQR) can be significantly different depending on the regularization parameter—i.e. the weight for the ℓ_1 regularization term in the objective function for these models—, we select the regularization parameter of these models by computing the metrics on a tuning split of the training data. For the mixture model, we similarly tune the number of components k by testing $k = \{10, 20, 30, \cdots, 100\}$. For the very high dimensional datasets where $p = 1000$, we use a regularization parameter near the tuning parameters found when $p = 100$ and fix $k = 50$ in order to avoid the extra computation of selecting a parameter. More sampling and implementation details for each model are available in the supplementary material.

5.6.3.2 Empirical Comparison Results

The full results for both the MMD and Spearman's ρ metrics for the crash severity, breast cancer RNA-Seq and Classic3 text datasets can be seen in Fig. 5.3, Fig. 5.4, and Fig. 5.5 respectively. The low dimensional results ($p \leq 10$) give evidence across all the datasets that three models outperform the others in their classes:[6] The Gaussian copula paired with Poisson marginals model ("Copula Poisson") for the marginal model class, the mixture of Poissons distribution ("Mixture Poiss") for the mixture model class, and the

[5] `https://cran.r-project.org/web/packages/XMRF/index.html`

[6] For the crash-severity dataset, the truncated Poisson graphical model ("Truncated PGM") outperforms the Poisson SQR model under the pairwise MMD metric. After inspection, however, we realized that the Truncated PGM model performed better merely because outlier values were truncated to the 99th percentile as described in the supplementary material. This reduced the overfitting of outlier values caused by the crash severity dataset's high overdispersion.

Poisson SQR distribution ("Poisson SQR") for the conditional model class. Thus, we only include these representative models along with an independent Poisson baseline in the high-dimensional experiments when $p > 10$. We discuss the results for specific data characteristics as represented by each dataset.[7]

For the crash severity dataset with high counts and high overdispersion (Fig. 5.3), mixture models (i.e. "Log-Normal" and "Mixture Poiss") perform the best as expected since they can model overdispersion well. However, if dependency structure is the only object of interest, the Gaussian copula paired with Poisson marginals ("Copula Poisson") performs well. For the BRCA dataset with medium counts and medium overdispersion (Fig. 5.4), we note similar trends with two notable exceptions: (1) The Poisson SQR model actually performs reasonably in low dimensions suggesting that it can model moderate overdispersion. (2) The high dimensional ($d \geq 100$) Spearman's ρ difference results show that the Gaussian copula paired with Poisson marginals ("Copula Poisson") performs significantly better than the mixture model; this result suggests that copulas paired with Poisson marginals are likely better for modeling dependencies than mixture models. Finally, for the Classic3 dataset with low counts and medium overdispersion (Fig. 5.5), the Poisson SQR model seems to perform well in this low-counts setting especially in low dimensions unlike in previous data settings. While the simple independent mixture of Poisson distributions still performs well, the Poisson log-normal mixture distribution ("Log-Normal") performs quite poorly in this setting with small counts and many zeros. This poor performance of the Poisson log-normal mixture is somewhat surprising since the dispersion indices are almost all greater than one as seen in Table 5.1. The differing results between low counts and medium counts with similar overdispersion demonstrate the importance to consider both the overdispersion and the mean count values when characterizing a dataset.

[7]These basic trends are also corroborated by the two datasets in the supplementary material.

In summary, we note several overall trends. Mixture models are important for overdispersion when counts are medium or high. The Gaussian copula with Poisson marginals joint distribution can estimate dependency structure (per the Spearman metric) for a wide range of data characteristics even when the distribution does not fit the underlying data (per the MMD metric). The Poisson SQR model performs well for low count values with many zeros (i.e. sparse data) and may be able to handle moderate overdispersion.

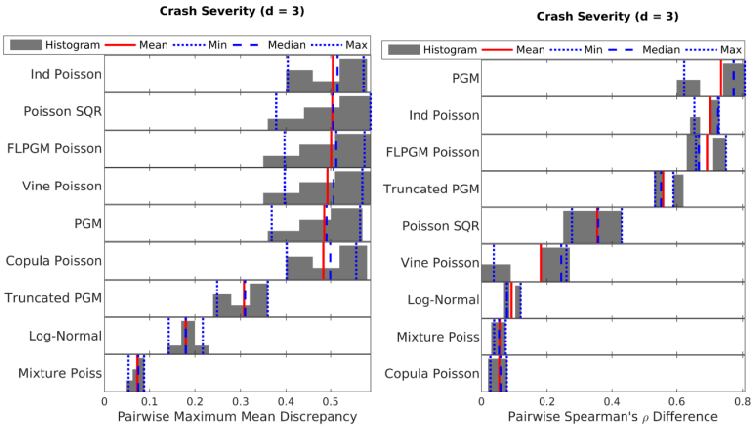


Figure 5.3: Crash severity dataset (high counts and high overdispersion): MMD (left) and Spearman ρ's difference (right). As expected, for high overdispersion, mixture models ("Log-Normal" and "Mixture Poiss") seem to perform the best.

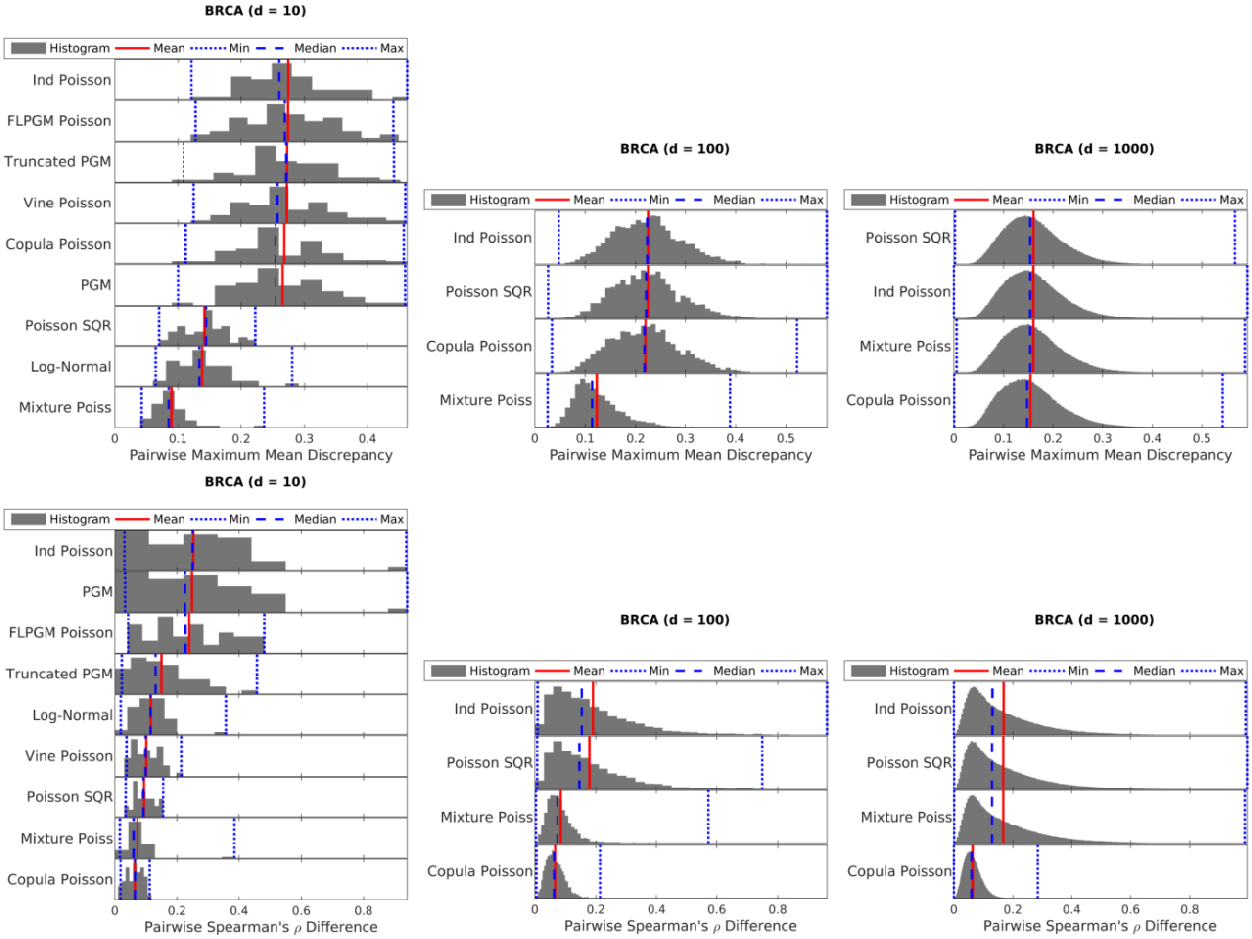


Figure 5.4: BRCA RNA-Seq dataset (medium counts and medium overdispersion): MMD (top) and Spearman ρ's difference (bottom) with different number of variables: 10 (left), 100 (middle), 1000 (right). While mixtures ("Log-Normal" and "Mixture Poiss") perform well in terms of MMD, the Gaussian copula paired with Poisson marginals ("Copula Poisson") can model dependency structure well as evidenced by the Spearman metric.

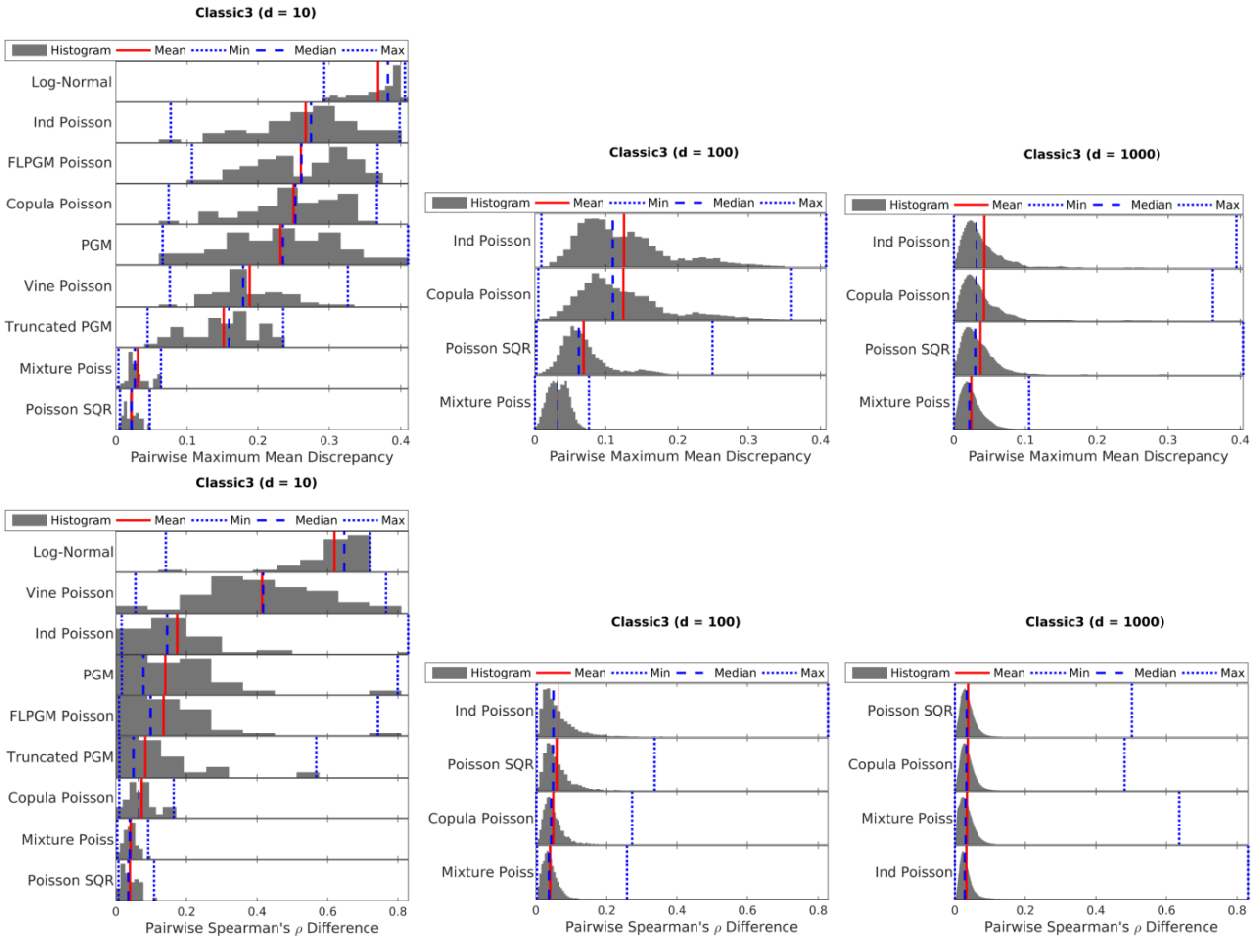


Figure 5.5: Classic3 text dataset (low counts and medium overdispersion): MMD (top) and Spearman ρ's difference (bottom) with different number of variables: 10 (left), 100 (middle), 1000 (right). The Poisson SQR model performs better on this low count dataset than in previous settings.

5.7 Discussion

While this review analyzes each model class separately, it would be quite interesting to consider combinations or synergies between the model classes. Because negative binomial distributions can be viewed as a gamma-Poisson mixture model, one simple idea is to consider pairing a copula with negative binomial marginals or developing a negative binomial SQR graphical model. As another example, we could form a finite mixture of copula-based or graphical-model-based models. This might combine the strengths of a mixture in handling multiple modes and overdispersion with the strengths of the copula-based models and graphical models which can explicitly model dependencies.

We may also consider how one type of model informs the other. For example, by the generalized Sklar's theorem [Rüschendorf, 2013], each conditional Poisson graphical model actually induces a copula—just as the Gaussian graphical model induces the Gaussian copula. Studying the copulas induced by graphical models seems to be a relatively unexplored area. On the other side, it may be useful to consider fitting a Gaussian copula paired with discrete marginals using the theoretically-grounded techniques from graphical models for sparse dependency structure estimation especially for the small sample regimes in which $p > n$; this has been studied for the case of continuous marginals in [Liu et al., 2012]. Overall, bringing together and comparing these diverse paradigms for probability models opens up the door for many combinations and synergies.

5.8 Conclusion

We have reviewed three main approaches to constructing multivariate distributions derived from the Poisson using three different assumptions: 1) the marginal distributions are derived from the Poisson, 2) the joint distribution is a mixture of independent Poisson distributions, and 3) the node-conditional distributions are derived from the Poisson. The

first class based on Poisson marginals, and in particular the general approach of pairing copulas with Poisson marginals, provides an elegant way to partially[8] decouple the marginals from the dependency structure and gives strong empirical results despite some theoretical issues related to non-uniqueness. While advanced methods to estimate the joint distribution of copulas paired with discrete marginals such as simulated likelihood [Nikoloulopoulos, 2016] or vine copula constructions provide more accurate or more flexible copula models respectively, our empirical results suggest that a simple Gaussian copula paired with Poisson marginals with the trivial distributional transform (DT) can perform quite well in practice. The second class based on mixture models can be particularly helpful for handling overdispersion that often occurs in real count data with the log-normal-Poisson mixture and a finite mixture of independent Poisson distributions being prime examples. In addition, mixture models have closed-form moments and in the case of a finite mixture, closed-form likelihood calculations—something not generally true for the other classes. The third class based on Poisson conditionals can be represented as graphical models, thus providing both compact and visually appealing representations of joint distributions. Conditional models benefit from strong theoretical guarantees about model recovery given certain modeling assumptions. However, checking conditional modeling assumptions may be impossible and may not always be satisfied for real-world count data. From our empirical experiments, we found that (1) mixture models are important for overdispersion when counts are medium or high, (2) the Gaussian copula with Poisson marginals joint distribution can estimate dependency structure for a wide range of data characteristics even when the distribution does not fit the underlying data, and (3) Poisson SQR models perform well for low count values with many zeros (i.e. sparse data) and can handle moderate overdispersion. Overall, in practice, we would recommend

[8]In the discrete case, the dependency structure cannot be perfectly decoupled from the marginal distributions unlike in the continuous case where the dependency structure and marginals can be perfectly decoupled.

comparing the three best performing methods from each class: namely the Gaussian copula model paired with Poisson marginals, the finite mixture of independent Poisson distributions, and the Poisson SQR model. This initial comparison will likely highlight some interesting properties of a given dataset and suggest which class to pursue in more detail.

This review has highlighted several key strengths and weaknesses of the main approaches to constructing multivariate Poisson distributions. Yet, there remain many open questions. For example, what are the marginal distributions of the Poisson graphical models which are defined in terms of their conditional distributions? Or conversely, what are the conditional distributions of the copula models which are defined in terms of their marginal distributions? Can novel models be created at the intersection of these model classes that could combine the strengths of different classes as suggested in the discussion section? Could certain model classes be developed in an application area that has been largely dominated by another model class? For example, graphical models are well-known in the machine learning literature while copula models are well-known in the financial modeling literature. Overall, multivariate Poisson models are poised to increase in popularity given the wide potential applications to real-world high-dimensional count-valued data in text analysis, genomics, spatial statistics, economics, and epidemiology.

Part II

Topic Models with Graphical Model Components

Summary of Part II

In the following chapters, we seek to combine the undirected Poisson graphical models described in Part I with topic models, such as the common Latent Dirichlet Allocation (LDA) model. In standard topic modeling, the topic distributions are assumed to be independent—e.g. a multinomial. However, often words within a topic are related—for example, the words "machine" and "learning" would likely be dependent in a computer science topic. Thus, we seek to replace the topic distributions with graphical models so that observations even within a topic can be dependent. In Chapter 6, we discuss two disjoint types of topic modeling generalizations that enable the components to be graphical models. Then, we design an instantiation of the first topic model generalization in Chapter 7 and an instantiation of the second topic model generalization in Chapter 8. In both cases, we present the model, an estimation algorithm, and experimental results on real-world datasets.

Chapter 6

Two Types of Topic Model Generalizations[1]

Topic modeling, as often represented by the standard Latent Dirichlet Allocation model (LDA) [Blei et al., 2003], is a way of generalizing the idea of mixture distributions. A mixture distribution assumes that each instance comes from one of k component distributions. One way of thinking about a mixture is assuming that each instance is associated with a latent indicator vector that designates which component the instance is from. The idea of topic models is to relax the assumption that the latent vector must be an indicator vector by assuming that the latent vector is positive and must sum to one. This essentially associates a latent weight vector with each data instance that signifies the instance-specific mixture of the the component distributions. Topic models can be seen as taking the convex relaxation of the indicator vectors, and thus mixture models can be viewed as a special case of topic models. LDA uses the independent multinomial as the component distributions. However, generalizing LDA to non-multinomial distributions can take two distinct forms. First, instances from LDA can be assumed to come from a multinomial whose parameters are a weighted mean (based on the instance-specific weight vector) of the component parameters; we call this generalization as *admixtures*. Second, the instances from LDA can be assumed to be a sum of latent vectors drawn from a fixed-length distribution whose lengths are determined by the instance-specific weight

[1]Portions of this chapter are from [Inouye et al., 2014b,a, 2015] with some edits for better integration into this dissertation. [Inouye et al., 2014b,a, 2015] were primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

vector. Both of these generalizations have LDA as a special case but allow for the multinomial distribution to be replaced by a graphical model. Thus, we can develop topic models in Chapters 7 and 8 that use graphical models as the component distributions.

6.1 Topic Model Generalization 1: Weighted Mean of Component Parameters (Admixtures)

In a simple *mixture model*, an observation is assumed to come from exactly one of k possible components. An illustration of this type of model is shown in Fig. 6.1 (left) in which documents are drawn from exactly one of two component distributions—the "topics" in the case of document modeling. On the other hand, for *admixtures* each document is drawn from a distribution whose parameters can be any convex combination of the component parameters, allowing each document to be explained by multiple components as illustrated in Fig. 6.1 (right).[2]

Given this intuition about admixtures, the probability of a single observation $\boldsymbol{x}$ from an *admixture* of some base distribution (e.g. multinomial, von Mises-Fisher, PMRF)—assuming that the admixture weights $\boldsymbol{w}$ and component canonical parameters $\Phi = \boldsymbol{\phi}_{1\ldots k}$ are given—is defined as:

$$\mathbb{P}_{\text{Admix.}}(\boldsymbol{x} \mid \boldsymbol{w}, \Phi) = \mathbb{P}_{\text{Base}}\left(\boldsymbol{x} \,\middle|\, \bar{\boldsymbol{\phi}} = \Psi^{-1}\left(\sum_{j=1}^{k} w_j \Psi(\boldsymbol{\phi}_j)\right)\right), \tag{6.1}$$

where Ψ allows for the mixing to occur in a suitable transformation of the parameter space. In the context of exponential families, the mixing could occur either in the canonical parameter space in which case Ψ would be the identity function, or it could occur in the mean parameter

[2]Fig. 6.1 is only meant as an illustration and not as a rigorous visualization of mixtures or admixtures. It should be noted that, in general, the KL-divergence should be minimized rather than ℓ_2 distance as suggested by the figure.

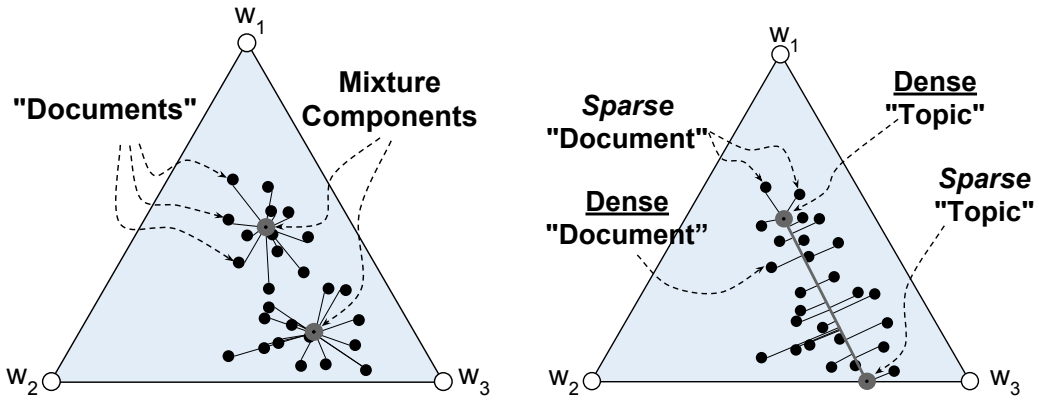


Figure 6.1: (Left) In *mixtures*, documents are drawn from exactly one component distribution. (Right) In *admixtures*, documents are drawn from a distribution whose parameters are a convex combination of component parameters.

space (such as the mean μ and covariance Σ for a multivariate Gaussian). For this paper, unless otherwise specified, we will assume that Ψ is equal to the identity function.

If priors are given for the admixture weights $\boldsymbol{w}$ and the topic parameters $\boldsymbol{\phi}_{1\ldots k}$ with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ hyperparameters respectively, the joint distribution of a single observation and the parameters is:

$$\mathbb{P}_{\text{Base}}\left(\boldsymbol{x} \,\middle|\, \bar{\boldsymbol{\phi}} = \sum_{j=1}^{k} w_j \boldsymbol{\phi}_j\right) \mathbb{P}(\boldsymbol{w}) \prod_{j=1}^{k} \mathbb{P}(\boldsymbol{\phi}_j)$$

This gives the joint distribution over a set of n independent observations as:

$$\mathbb{P}_{\text{Admix.}}(X, \mathrm{W}, \Phi) = \tag{6.2}$$
$$\prod_{i=1}^{n} \mathbb{P}_{\text{Base}}\left(\boldsymbol{x}_i \,\middle|\, \bar{\boldsymbol{\phi}} = \sum_{j=1}^{k} w_{i,j} \boldsymbol{\phi}_j\right) \mathbb{P}(\boldsymbol{w}_i) \prod_{j=1}^{k} \mathbb{P}(\boldsymbol{\phi}_j)$$

Intuitively, this admixture model formulation means that each observation can be explained by a mixture of a relatively small number of component distributions parameterized by $\boldsymbol{\phi}_j$. In the special case where w is an indicator vector, this distribution becomes a standard

mixture model where each observation is explained by only one component. In the special case where $k = 1$, the admixture simply reduces to every observation being drawn from a single base distribution. Therefore, this admixture formulation generalizes both single and mixture distributions.

This admixture model also generalizes previous topic models and provides a general framework for defining new admixture models based on any parametric distribution. In the next sections, several examples of previous admixture models are given followed by the formulation of this paper's main model—an admixture of Poisson MRFs which, to the authors' best knowledge, is the first admixture model to allow dependencies between words.

Example 1 - LDA As shown in Sec. 7.3.1, LDA assumes that each document is draw from an admixture of multinomials. The admixture weights and the parameters for each topic multinomial are drawn from a Dirichlet prior. It is important to notice that LDA mixes in the standard multinomial mean parameter space (i.e. Ψ_{LDA} is the canonical to mean parameter transformation).

Example 2 - Population Admixtures In the genetic community, the term *admixture* has been used to describe a population produced by interbreeding several previously-isolated populations into a new *admixed* population. Pritchard et al. [2000] use a model equivalent to LDA to explore this concept. Under this population model, the original ancestors of a population correspond to *topics* and individuals correspond to *documents*.

Example 3 - SAM The Spherical Admixture Model (SAM) as proposed by Reisinger et al. [2010] is an admixture model where the base distribution is a von Mises-Fisher distribution—the independent Gaussian analog defined on the unit hypersphere. The model, which is motivated by the observation that cosine distance is an important document similarity,

assumes Dirichlet and von Mises-Fisher priors on the admixture weights and component parameters respectively.

6.2 Topic Model Generalization 2: Sum of Latent Fixed-Length Variables

In standard topic models like LDA, the distribution contains a unique topic variable for every word in the corpus. Essentially, this means that every word is actually drawn from a categorical distribution. However, this does not allow us to capture dependencies between words because there is only one word being drawn at a time. Therefore, we need to reformulate LDA in a way that the words from a topic are sampled jointly from a multinomial. From this reformulation, we can then simply replace the multinomial with an LPMRF distribution to obtain an LPMRF topic model described in Chapter 8. Our reformulation of LDA groups the topic indicator variables for each word into k vectors corresponding to the k different topics. These k "topic indicator" vectors $\boldsymbol{z}^l$ are then assumed to be drawn from a multinomial with fixed length $L = \|\boldsymbol{z}^j\|$. This grouping of topic vectors yields an equivalent distribution because the topic indicators are exchangeable and independent of one another given the observed word and the document-topic distribution. This leads to the following generalization of topic models in which an observation $\boldsymbol{x}_i$ is the summation of k hidden variables $\boldsymbol{z}_i^j$:

Generic Topic Model	**Novel LPMRF Topic Model**
$\boldsymbol{w}_i \sim \text{SimplexPrior}(\alpha)$	$\boldsymbol{w}_i \sim \text{Dirichlet}(\alpha)$
$L_i \sim \text{LengthDistribution}(\bar{L})$	$L_i \sim \text{Poisson}(\lambda = \bar{L})$
$\boldsymbol{m}_i \sim \text{PartitionDistribution}(\boldsymbol{w}_i, L_i)$	$\boldsymbol{m}_i \sim \text{Multinomial}(\boldsymbol{p} = \boldsymbol{w}_i; N = L_i)$
$\boldsymbol{z}_i^j \sim \text{FixedLengthDist}(\boldsymbol{\phi}^j\ ;\ \|\boldsymbol{z}_i^j\| = m_i^j)$	$\boldsymbol{z}_i^j \sim \text{LPMRF}(\boldsymbol{\theta}^j, \Phi^j; L = m_i^j)$
$\boldsymbol{x}_i = \sum_{j=1}^k \boldsymbol{z}_i^j$	$\boldsymbol{x}_i = \sum_{j=1}^k \boldsymbol{z}_i^j.$

Note that this generalization of topic models does not require the partition distribution and the fixed-length distribution to be the same. In addition, other distributions could be substituted for the Dirichlet prior distribution on document-topic distributions like the logistic normal prior. Finally, this generalization allows for real-valued topic models for other types of data although exploration of this is outside the scope of this work.

This generalization is distinctive from the topic model generalization termed "admixtures" in [Inouye et al., 2014b]. Admixtures assume that each observation is drawn from an instance-specific base distribution whose parameters are a convex combination of previous parameters. Thus an admixture of LPMRFs could be formulated by assuming that each document, given the document-topic weights $\boldsymbol{w}_i$, is drawn from a LPMRF($\bar{\boldsymbol{\theta}}_i = \sum_j w_{ij}\boldsymbol{\theta}^j, \bar{\Phi}_i = w_{ij}\Phi^j; L = \|\boldsymbol{x}_i\|_1$). Though this may be an interesting model in its own right and useful for further exploration in future work, this is not the same as the above proposed model because the distribution of $\boldsymbol{x}_i$ is not an LPMRF but rather the distribution of a sum of different LPMRF variables—which is not an LPMRF distribution in general. One case—possibly the only case—where these two generalizations of topic models intersect is when the distribution is a multinomial (i.e. a LPMRF with $\Phi = 0$). As another distinction from APM, the LPMRF topic model *directly* generalizes LDA because the LPMRF in the above model reduces to a multinomial if $\Phi = 0$. Finally, the LPMRF topic model can actually produce topic assignments for each word similar to LDA with Gibbs sampling [Steyvers and Griffiths, 2007], whereas APM cannot assign topics to specific words but only gives a topic distribution for each document. Fully exploring the differences between this topic model generalization and the admixture generalization are quite interesting but outside the scope of this work.

Chapter 7

Admixture of Poisson MRFs[1]

7.1 Abstract

We introduce a new topic model based on an admixture of Poisson Markov Random Fields (APM), which can model dependencies between words as opposed to previous independent topic models such as PLSA [Hofmann, 1999], LDA [Blei et al., 2003] or SAM [Reisinger et al., 2010]. As one contribution, we propose a class of admixture models that generalizes previous topic models and show an equivalence between the conditional distribution of LDA and independent Poissons—suggesting that APM subsumes the modeling power of LDA. Research in both the semantic coherence of a topic models [Mimno et al., 2011, Newman et al., 2010, Stevens et al., 2012, Aletras and Stevenson, 2013] and measures of model fitness [Mimno and Blei, 2011] provide strong support that explicitly modeling word dependencies—as in APM—could be both semantically meaningful and essential for appropriately modeling real text data. Though APM shows significant promise for providing a better topic model, APM has a high computational complexity because $O(p^2)$ parameters must be estimated where p is the number of words. In light of this, we develop a parallel alternating Newton-like algorithm for training the APM model that can handle $p = 10^4$ as an important step towards scaling to large datasets. Also, motivated by simple intuitions and previous evaluations of topic models, we

[1]The majority of this chapter is from [Inouye et al., 2014b,a] with some edits for better integration into this dissertation. [Inouye et al., 2014b,a] was primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

propose a novel evaluation metric based on human *evocation* scores between word pairs (i.e. how much one word "brings to mind" another word [Boyd-Graber et al., 2006]). We provide compelling quantitative and qualitative results on the BNC corpus that demonstrate the superiority of APM over previous topic models for identifying semantically meaningful word dependencies. (MATLAB code available at: `http://bigdata.ices.utexas.edu/software/apm/`)

(Note: The Poisson MRF in this chapter refers to the SPGM model defined in Chapter 2 [Yang et al., 2013]. However, extensions of this model to use the SQR graphical model would be an interesting area of future work.)

7.2 Introduction

Topic models can be understood as a class of statistical models for document collections that model documents as admixtures over *topics*. Specifically, each topic is modeled as a distribution over words, and each document is a separate mixture of such topics (or specifically, the word distributions comprising the topics). Such an admixture can be contrasted with a vanilla mixture of topics, where each document would be drawn from a single topic.

A popular set of topic models is PLSA [Hofmann, 1999], which uses the multinomial distribution as the word distribution for any topic, and its Bayesian counterpart, LDA [Blei et al., 2003], which adds Dirichlet priors. While these topic models have proved enormously useful in modeling varied document collections and have attracted a long line of work with numerous extensions (see [Blei et al., 2010] for a review of LDA applications and trends), it has some crucial lacunae that arise from its basic use of the multinomial distribution to model word distributions for topics. There are several reasons which make the multinomial distribution an inadequate distribution for documents and topics. The primary issue is that it does not model dependencies between words: if the word "kernels" appears in a document

(specifically, a machine learning paper), the appearance of the word "graphs" might be less likely. Alternatively, if the word "classification" appears, "supervised" is more likely to appear than in general documents. A second caveat is that the multinomial distribution does not model absences of words. Lastly, the multinomial word distribution does not leverage varying document lengths. For instance, with large counts of other words, some specific word might become less likely. To address the issue of modeling word absences, Reisinger et al. [2010] proposed the use of von Mises-Fisher distribution for topic distributions. But while this addresses one issue with multinomials, it does not model word dependencies, nor does it leverage document lengths in any substantive way.

In standard topic models such as LDA [Blei et al., 2003, Griffiths and Steyvers, 2004], the primary representation for each topic is simply a list of top 10 or 15 words. To understand a topic, a person must manually consider many of the possible $\binom{10}{2}$ pairwise relationships as well as possibly larger m-wise relationships and attempt to infer abstract meaning from this list of words. Of all the $\binom{10}{2}$ pairwise relationships probably a very small number of them are direct relationships. For example, a topic with the list of words "money", "fund", "exchange" and "company" can be understood as referring to investment but this can only be inferred from a very high-level human abstraction of meaning. This problem has given rise to research on automatically labeling topics with a topic word or phrase that summarizes the topic [Lau et al., 2011, Magatti et al., 2009, Mao et al., 2012]. [Chang et al., 2009] propose to evaluate topic models by randomly replacing a topic word with a random word and evaluating whether a human can identify the intruding word. The intuition for this metric is that the top words of a good topic will be related, and therefore, a person will be able to easily identify the word that does not have any relationship to the other words. [Mimno et al., 2011, Newman et al., 2010, Aletras and Stevenson, 2013] compute statistics related to Pointwise Mutual Information for all pairs of top words in a topic and attempt to correlate this with human judgments. All of these metrics suggest that

capturing semantically meaningful relationships between pairs of words is fundamental to the interpretability and usefulness of topic models as a document summarization and exploration tool. This need for modeling dependencies can also be motivated in part by [Mimno and Blei, 2011] who investigated whether the multinomial (i.e. independent) assumption of word-topic distributions actually fits real-world text data. Somewhat unsurprisingly, [Mimno and Blei, 2011] found that the multinomial assumption was often violated and thus gives evidence that models with word dependencies—such as APM—may be a fundamentally more appropriate model for text data.

Previous research in topic modeling has implicitly uncovered this issue with model misfit by finding that models with 50, 100 or even 500 topics tend to perform better on semantic coherence experiments than smaller models with only 10 or 20 topics [Stevens et al., 2012]. Though using more topics may allow topic models to ignore the issue of word dependencies, using more topics can make the coherence of a topic model more difficult as suggested by [Stevens et al., 2012] who found that using 100 or 500 topics did not significantly improve the coherence results over 50 topics. Intuitively, a topic model with a much smaller number of topics (e.g. 5 or 10) is easier to comprehend. For instance, if training on newspaper text, the number of topics could roughly correspond to the number of sections in a newspaper such as news, weather and sports. Or, if modeling an encyclopedia, the top-level topics could be art, history, science, and society. Thus, rather than using more topics, APM opens the way for a promising topic model that can overcome this model misfit issue while only using a small number of topics.

We propose using Poisson MRFs [Yang et al., 2012] described in Chapter 2, and in particular the SPGM variant in [Yang et al., 2013],[2] for topic distributions and using the

[2]Because we are not estimating the joint likelihood, we just compute regular Poisson regressions instead of sublinear Poisson regressions—implicitly setting the SPGM cutoff points R_0 and R to high values such that they do not have an effect. Clearly, using Poisson SQR model would be an interesting area of future work that avoids this issue with using unintuitive cutoff points.

resulting admixture of Poisson MRFs (APM) for modeling document collections. These Poisson MRFs allow modeling multivariate count data and use dependencies among the count variables to represent the joint distribution compactly. Moreover, since these are graphical model distributions, the dependencies are Markov with respect to an undirected graph, which thus provides a visually appealing representation of any topic—see Fig. 7.1—in contrast to just a list of words as in PLSA/LDA.

We position the Poisson MRF in context of topic models by showing that the conditional distributions of the classical LDA model can be written as a Poisson MRF

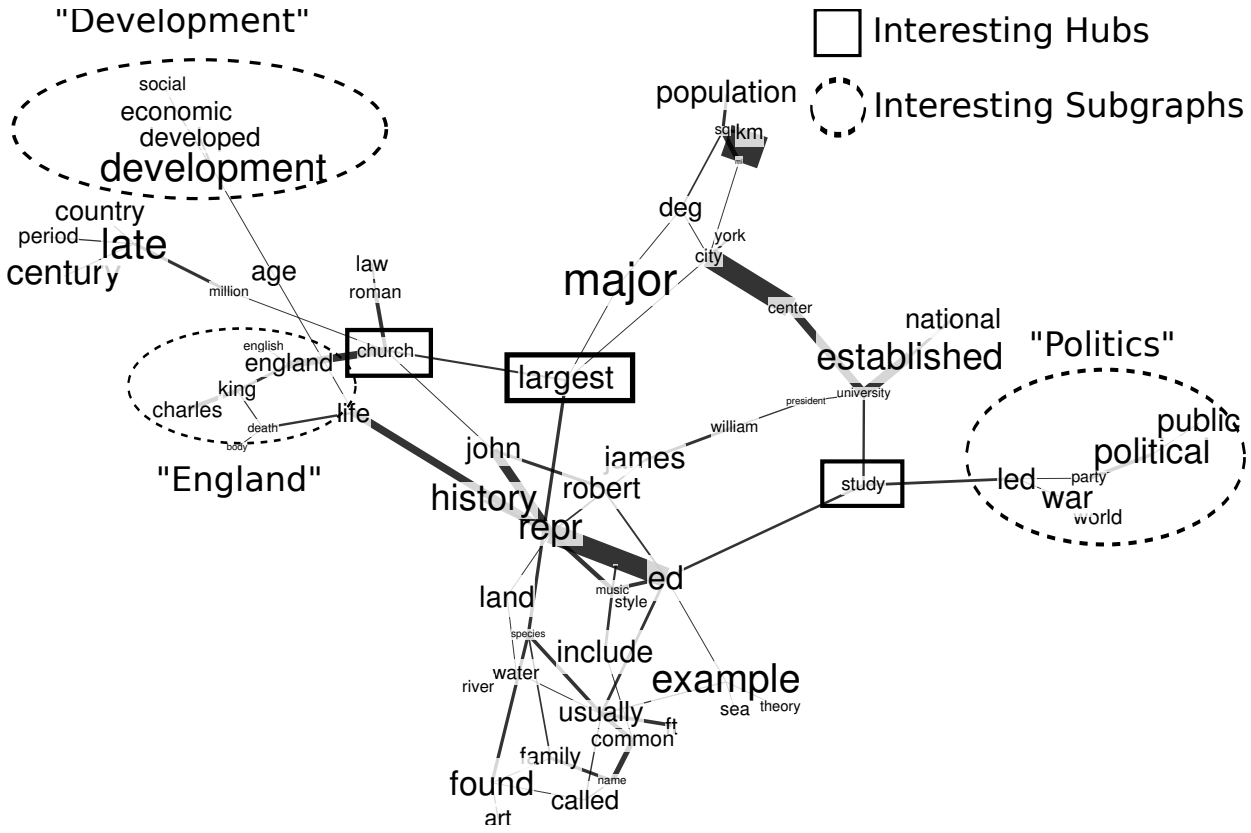


Figure 7.1: A Poisson MRF can provide interesting insights into a text corpus including multiple word senses (hubs of graph) and semantic concepts (coherent subgraphs).

where the underlying graph has no edges and hence no dependencies between words; this connection—which was only recently discovered in the context of matrix factorization [Gopalan et al., 2013]—not only puts into relief the assumptions made by LDA but also opens the door to other approximate inference schemes for LDA (which however we do not explore here). In other contributions of this paper, we define a new class of models called admixtures and show that this class generalizes previous topic models, which thus opens the door to other topic models based on non-Poisson distributions.

Even though APM shows promise for being a significantly more powerful and more realistic topic model than previous models, the original paper acknowledged the significant computational complexity. Instead of needing to fit $O(k(n+p))$ parameters, APM needs to estimate $O(k(n+p^2))$ parameters. Therefore, we develop a parallel alternating algorithm whose independent subproblems are solved using a Newton-like algorithm similar to the algorithms developed for sparse inverse covariance estimation [Hsieh et al., 2011]. As in [Hsieh et al., 2011], this new APM algorithm exploits the sparsity of the solution to significantly reduce the computational time for computing the approximate Newton direction. However, unlike [Hsieh et al., 2011], the APM model is solving for k Poisson MRFs simultaneously whereas [Hsieh et al., 2011] is only solving for a single Gaussian MRF. Another difference from [Hsieh et al., 2011] is that the whole algorithm can be easily parallelized up to $\min(n, p)$.

To help measure the semantic utility of APM, we develop a novel evaluation metric that more directly evaluates the APM model against human judgments of semantic relatedness—a notion called *evocation* introduced by [Boyd-Graber et al., 2006]. Intuitively, the idea is that humans seek to understand traditional topic models by looking at the list of top words. They will implicitly attempt to find how these words are related and extract some more abstract meaning that generalizes the set of words. Thus, this evaluation metric attempts to explicitly score how well pairs of words capture some

semantically meaningful word dependency. Previous research has evaluated topic models using word similarity measures [Stevens et al., 2012]. However, our work is different from [Stevens et al., 2012] in three significant ways: 1) our metrics use evocation rather than similarity (e.g. antonyms should have high evocation but low similarity), 2) we evaluate top individual word pairs instead of rough aggregate statistics, and 3) we evaluate a topic model that directly captures word dependencies (i.e. APM). We demonstrate that APM substantially outperforms other topic models in both quantitative and qualitative ways.

7.3 Poisson MRFs in the Context of Topic Models

7.3.1 LDA Conditionals Equivalent to Independent Poissons

In this section, we place Poisson MRFs in the context of topic models by showing the equivalence between the conditionals of LDA and an independent Poisson MRF.[3] LDA assumes the following generative process for a new document given that the topic weights for the document $\boldsymbol{w}$ and the topic distribution parameters $\boldsymbol{\phi}_{1...k}$ are known: 1) Draw $\tilde{x} \sim$ Poisson($\tilde{\lambda}$) 2) For each of the $\tilde{x}$ words: (a) Draw topic index $z \sim$ Categorical($\boldsymbol{w}$) (b) Draw word $v \sim$ Categorical($\boldsymbol{\phi}_z$). Notice that because $\tilde{x}$ is independent of the other variables in LDA, it is often simply ignored when estimating the model parameters. In our model, however, $\tilde{x}$ cannot be ignored because words can be dependent. By marginalizing out the topic variable z, step 2 can be collapsed into a draw from a multinomial with a single parameter $\tilde{\boldsymbol{\phi}}$, which is simply a weighted average over the topic distribution parameters $\boldsymbol{\phi}_z$. This yields the following modified step: 2') Draw document $\boldsymbol{x} \sim \text{Mult}(\tilde{\boldsymbol{\phi}} = \sum_{j=1}^{k} w_j \boldsymbol{\phi}_j \mid N = \tilde{x})$. Therefore, the probability of a document $\boldsymbol{x}$ given $\boldsymbol{w}$ and $\boldsymbol{\phi}_{1...k}$ is: $\mathbb{P}_{\text{Poiss}}(\tilde{x} \mid \tilde{\lambda}) \mathbb{P}_{\text{Mult}}(\boldsymbol{x} \mid \tilde{\boldsymbol{\phi}} = \sum_{j=1}^{k} w_j \boldsymbol{\phi}_j, N = \tilde{x})$.

[3] Gopalan et al. [2013] recently introduced the connection between LDA and Poisson models in the context of matrix factorization.

Amazingly, this Poisson-multinomial joint distribution is equivalent to p independent Poissons [Bishop et al., 2007]:

$$\begin{aligned}
\mathbb{P}_{\text{Ind. Poiss}}\left(\boldsymbol{x} \mid \lambda_1, \cdots, \lambda_p\right) &= \prod_{s=1}^{p} \frac{e^{-\lambda_s}}{x_s!} \lambda_s^{x_s} \\
&= \frac{\tilde{x}!}{\tilde{x}!} \frac{e^{-\tilde{\lambda}}}{\prod_{s=1}^{p} x_s!} \prod_{s=1}^{p} \left(\frac{\tilde{\lambda}\lambda_s}{\tilde{\lambda}}\right)^{x_s} \\
&= \frac{e^{-\tilde{\lambda}}}{\tilde{x}!} \tilde{\lambda}^{\tilde{x}} \frac{\tilde{x}!}{\prod_{s=1}^{p} x_s!} \prod_{s=1}^{p} \left(\frac{\lambda_s}{\tilde{\lambda}}\right)^{x_s} \\
&= \mathbb{P}_{\text{Poiss}}\left(\tilde{x} \mid \tilde{\lambda}\right) \mathbb{P}_{\text{Mult}}\left(\boldsymbol{x} \mid \theta = \left(\lambda_1, \cdots, \lambda_p\right)/\tilde{\lambda}, \mathrm{N} = \tilde{x}\right)
\end{aligned}$$

where $\tilde{\lambda} = \sum_{s=1}^{p} \lambda_s$ and $\tilde{x} = \sum_{s=1}^{p} x_s$. Therefore, a PMRF directly generalizes the conditional distribution of PLSA/LDA by relaxing the independence assumption. To more fully generalize LDA, priors must be added to a PMRF as proposed next.

7.3.2 Adding Priors to a PMRF

Similar to LDA's prior, a conjugate prior on the parameters of a PMRF can be defined as being proportional to:

$$\exp\{\boldsymbol{\beta}^T\boldsymbol{\theta} + \boldsymbol{\beta}^T\Theta\boldsymbol{\beta} - \gamma \mathrm{A}\left(\boldsymbol{\theta}, \Theta\right) - \lambda_{\boldsymbol{\theta}}\|\boldsymbol{\theta}\|_2^2 - \lambda\|\mathrm{vec}(\Theta)\|_1\},$$

where $\forall s,\ \beta_s > 0$, $\gamma \geq 0$, $\lambda_{\boldsymbol{\theta}} > 0$ and $\lambda > \max_{i,j} \beta_i\beta_j$.[4] One observation is that when $\Theta = 0$, $\exp(\theta_s)$ is essentially $\mathrm{Gam}(\text{shape} = \boldsymbol{\beta}_s; \text{scale} = 1)$. Therefore, for independent Poissons, this is similar to using Gamma priors. The posterior distribution merely modifies the hyperparameters to be $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \boldsymbol{x}$ and $\tilde{\gamma} = \gamma + 1$. Therefore, because this prior adds pseudo-counts $\boldsymbol{\beta}$ to the observations for parameter estimation, this prior for PMRFs is analogous to a Dirichlet prior for multinomials as in LDA.

[4]The conditions on the hyperparameters are needed for normalization. In practice, $\lambda_{\boldsymbol{\theta}}$ can be set arbitrarily small and is thus ignored in subsequent discussions. The λ hyperparameter is used for ℓ_1 regularization as discussed in Sec. 7.5.1.

7.4 Admixture of Poisson MRFs

With the background on Poisson MRFs in Chapter 2 and the development of admixtures in Chapter 6, an admixture of Poisson MRFs (APM) can be developed. Relaxing the independence assumption of previous admixture models such as LDA, APM assumes that the base distribution is a PMRF. This yields the following joint distribution:

$$\begin{aligned} &\mathbb{P}_{\text{APM}}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\theta}_{1\ldots k}, \Theta_{1\ldots k}) \\ &\quad = \mathbb{P}_{\text{PMRF}}\left(\boldsymbol{x} \,\middle|\, \bar{\boldsymbol{\theta}} = \sum_{j=1}^{k} w_j \boldsymbol{\theta}_j, \bar{\Theta} = \sum_{j=1}^{k} w_j \Theta_j\right) \\ &\qquad \times \mathbb{P}_{\text{Dir}}(\boldsymbol{w}) \prod_{j=1}^{k} \mathbb{P}(\boldsymbol{\theta}_j, \Theta_j) \end{aligned} \tag{7.1}$$

where $\mathbb{P}_{\text{Dir}}(\boldsymbol{w} \mid \boldsymbol{\alpha})$ is a Dirichlet prior on the admixture weights (similar to LDA) and $\mathbb{P}(\boldsymbol{\theta}_j, \Theta_j)$ is the PMRF prior defined in Sec. 7.3.2. Because of the equivalence described in Sec. 7.3.1, APM subsumes the expressive power of LDA. The primary difference between an independent APM and LDA is that LDA mixes in the standard multinomial parameter space whereas APM mixes in the canonical parameter space. An interesting open area for future research could be admixing the component PMRFs in a different parameter space such as the mean parameter space.[5] Fundamentally, however, this model is much more expressive than all previous admixture models because it allows for dependencies between words.

7.4.1 Topic Representation

In the APM model, topics are represented as PMRFs, and therefore, each topic provides a full graph over words showing word dependencies rather than just a list of words as in independent models such as LDA (see Fig. 7.3 in Sec. 7.7 for example topic graphs).

[5] For more information on the relationship between the mean and canonical parameter spaces, see [Wainwright and Jordan, 2008].

This representation opens up a whole new area for interpreting, exploring and visualizing topics using a graph. In addition, all the metrics and algorithms on graphs such as tree width or shortest path could be used to explore each topic.

7.4.2 Document Representation

Documents could be represented in at least two different ways under the APM model. First, they could be represented by their admixture weights, and therefore, APM could be used as a type of dimensionality reduction technique. Second, each document can be represented as a full graph over words just like a topic because each document is associated with an admixed PMRF. This graph representation provides a powerful new way to visualize and summarize a document that was not possible with independent models like LDA.

7.5 Parameter Estimation by Optimizing Approximate Posterior

The parameters of an admixture of Poisson MRFs can be estimated by minimizing the negative log posterior. Because the true log-likelihood of a Poisson MRF is computationally intractable for complex multivariate distributions [Wainwright and Jordan, 2008], the pseudo log-likelihood—which approximates the joint distribution as a product of node conditionals—will be used instead. With the Dirichlet prior on $\boldsymbol{w}$ and the prior described in Sec. 7.3.2 on the component parameters, the approximate posterior is:

$$\begin{aligned} \mathcal{P} &\approx \hat{\mathcal{P}}(\mathrm{W}, \boldsymbol{\theta}_{1\ldots k}, \Theta_{1\ldots k} \mid X) \\ &\propto \sum_{i=1}^{n} \left\{ \left[\sum_{s=1}^{p} \eta_{s,i} \hat{x}_{s,i} - (\gamma{+}1)\mathrm{A}\left(\eta_{s,i}\right) \right] + (\boldsymbol{\alpha}{-}1)^T \log(\boldsymbol{w}_i) \right\}, \end{aligned} \tag{7.2}$$

where $\hat{\boldsymbol{x}} = \boldsymbol{x}{+}\boldsymbol{\beta}$ and $\eta_{s,i} = \sum_{j=1}^{k} w_{i,j}(\boldsymbol{\theta}_{j,s} + \Theta_{j,s}\hat{\boldsymbol{x}}_{i,\setminus s})$ is the canonical parameter of a univariate Poisson.

7.5.1 Enforcing Sparsity of Θ_j by ℓ_1 Regularization

For interpretability, generalizability and computational tractability, the parameters of high-dimensional MRFs are often assumed to be sparse (i.e. a small number of non-zeros compared to zeros). This sparsity assumption is usually incorporated into the problem by adding an ℓ_1 regularization term to the objective function. This ℓ_1 regularized estimator has been shown to have theoretical guarantees on structural recovery for Bernoulli MRFs/Ising Models [Ravikumar et al., 2010], Gaussian MRFs [Ravikumar et al., 2011] and, more recently, Poisson MRFs [Yang et al., 2012, 2013]. For similar reasons, APM assumes that the parameter matrices (Θ_j) for each topic PMRF are sparse and, like the aforementioned methods, estimates this sparse solution by using an ℓ_1 regularization term. Intuitively, this sparsity assumption makes sense because most words are only directly related to a small subset of other relevant words.

7.5.2 Unconstrained Optimization

Along with the regularization of the Θ_j parameter matrices, APM requires that the columns of the admixture weights matrix W be probability vectors (i.e. properly defined mixture weights that lie on the k-dimensional simplex). This leads to the following unconstrained optimization problem:

$$\underset{\mathrm{W},\boldsymbol{\theta}_{1\ldots k},\Theta_{1\ldots k}}{\arg\min} \; -\hat{\mathcal{P}} + \delta_{\mathbb{W}}(\mathrm{W}) + \lambda \sum_{j=1}^{k} \|\mathrm{vec}(\Theta_j)\|_1 \tag{7.3}$$

where λ is the ℓ_1 regularization parameter, $\mathbb{W}$ is the set of all possible matrices such that the columns are probability vectors and $\delta_{\mathbb{W}}(\mathrm{W}) = \{0, \text{if } \mathrm{W} \in \mathbb{W};\ \infty, \text{otherwise}\}$.

7.6 Parallel Alternating Newton-like Algorithm for APM

The parameters for APM are estimated by maximizing the joint approximate posterior over all variables.[6] Instead of maximizing jointly over all parameters, we split the problem into alternating convex optimization problems. Let us denote the likelihood part (i.e. the smooth part) of the optimization function as $g(\mathrm{W}, \boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k})$ and the non-smooth ℓ_1 regularization term as h where the full negative posterior is defined as $f = g + h$. The smooth part of the approximate posterior can be written as:

$$g = -\frac{1}{n}\sum_{i=1}^{n}\sum_{s=1}^{p}\Big[\sum_{j=1}^{k} w_{ij}x_{is}(\theta_s^j + \boldsymbol{x}_i^T\Phi_s^j) - \exp\big(\sum_{j=1}^{k} w_{ij}(\theta_s^j + \boldsymbol{x}_i^T\Phi_s^j)\big)\Big], \tag{7.4}$$

where $\boldsymbol{x}_i$ is the word-count vector for the ith document, $\boldsymbol{w}_i$ is the admixture weight vector for the ith document, and $\boldsymbol{\theta}^j$ and Φ^j are the PMRF parameters for the jth component (see Appendix B.2 for derivation). By writing g in this form, it is straightforward to see that even though the whole optimization problem is not convex because of the interaction between the admixture weights w and the PMRF parameters, the problem is convex if either the admixture weights W or the component parameters $\boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k}$ are held fixed. To simplify the notation in the following sections, we combine the node (which is analogous to an intercept term in regression) and edge parameters by defining $\boldsymbol{z}_i = [1\ \boldsymbol{x}_i^T]^T$, $\boldsymbol{\phi}_s^j = [\theta_s^j\ (\Phi_s^j)^T]^T$ and $\boldsymbol{\Phi}^s = [\boldsymbol{\phi}_s^1\, \boldsymbol{\phi}_s^2 \cdots \boldsymbol{\phi}_s^k]$.

Thus, we can alternate between optimizing two similar optimization problems where one has a non-smooth ℓ_1 regularization and the other has the constraint that $\boldsymbol{w}_i$ must lie on

[6] This posterior approximation was based on the pseudo-likelihood while ignoring the symmetry constraint so that node-wise regression parameters are independent. This leads to an overcomplete parameterization for APM. For an overview of composite likelihood methods, see [Varin et al., 2011]. For a comparison of pseudo-likelihood versus node-wise regressions, see [Lee and Hastie, 2013].

the simplex Δ^k:

$$\underset{\boldsymbol{\Phi}^1,\boldsymbol{\Phi}^2,\cdots,\boldsymbol{\Phi}^p}{\arg\min} \quad -\frac{1}{n}\sum_{s=1}^{p}\Big[\operatorname{tr}(\Psi^s\boldsymbol{\Phi}^s) - \sum_{i=1}^{n}\exp(\boldsymbol{z}_i^T\boldsymbol{\Phi}^s\boldsymbol{w}_i)\Big] + \sum_{s=1}^{p}\lambda\|\operatorname{vec}(\boldsymbol{\Phi}^s)_{\setminus 1}\|_1 \tag{7.5}$$

$$\underset{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_n\in\Delta^k}{\arg\min} \quad -\frac{1}{n}\sum_{i=1}^{n}\Big[\boldsymbol{\psi}_i^T\boldsymbol{w}_i - \sum_{s=1}^{p}\exp(\boldsymbol{z}_i^T\boldsymbol{\Phi}^s\boldsymbol{w}_i)\Big], \tag{7.6}$$

where $\boldsymbol{\psi}_i$ and Ψ^s are constants in the optimization that can be computed from the data matrix X and the other parameters that are being held fixed (see Alg. 2 in Appendix B.4 for computation of Ψ^s). This alternating scheme is analogous to Alternating Least Squares (ALS) for Non-negative Matrix Factorization (NMF) [Lee and Seung, 2006] and EM-like algorithms such as k-means. By writing the optimization as in Eq. 7.5 and Eq. 7.6, we also expose the simple independence between the subproblems because they are simple summations. Thus, we can easily parallelize both optimization problems up to $\min(n, p)$ with little overhead and simple changes to the code—in our MATLAB implementation, we only changed a `for` loop to a `parfor` loop.

7.6.1 Newton-like Algorithms for Subproblems

For each of the subproblems, we develop Newton-like optimization algorithms. For the component PMRFs, we borrow several important ideas from [Hsieh et al., 2011] including *fixed* and *free* sets of variables for the ℓ_1 regularized optimization problem. The overall idea is to construct a quadratic approximation around the current solution and approximately optimize this simpler function to find a step direction. Usually, finding the Newton direction requires computing the Hessian for all the optimization variables but because of the ℓ_1 regularization, we only need to focus on variables that might be non-zero. This set of *free* variables, denoted $\mathcal{F}$, can be simply determined from the gradient and current iterate [Hsieh et al., 2011]. Since usually there is only a small number of *free* variables compared to *fixed* variables (i.e. λ is large enough), we can simply run coordinate descent on these free variables

and only implicitly calculate Hessian information as needed in each coordinate descent step. After finding an approximate Newton direction, we find a step size that satisfies the Armijo rule and then update the iterate (see Alg. 2 in Appendix B.4).

We also employed a similar Newton-like algorithm for estimating the admixture weights. Instead of the ℓ_1 regularization term, however, this subproblem has the constraint that the admixture weights $\boldsymbol{w}_i$ must lie on the simplex so that each document can be properly interpreted as a convex mixture of over topic parameters. For this constraint, we used a dual-coordinate descent algorithm to find the approximate Newton direction as in [Yu et al., 2011].

Finally, we put both subproblem algorithms together and alternate between the two (see Alg. 1 in Appendix B.4). For tracing through different λ parameters, λ is initially set to ∞ so that the model trains an independent APM model first. Then, the initial $\lambda = \lambda_{\max}$ is found by computing the largest gradient of the final independent iteration. Every time the alternating algorithm converges, the value of λ is decreased so that a set of models is trained for decreasing values of λ.

7.6.2 Timing Results

We conducted two main timing experiments to show that the algorithm can be efficiently parallelized and the algorithm can scale to reasonably large datasets. For the parallel timing experiment, we used the BNC corpus described in Sec. 7.8.1 ($n = 4049$, $p = 1646$) and fixed $k = 5$, $\lambda = 8$ and a total of 30 alternating iterations. For the large data experiment, we used a Wikipedia dataset formed from a recent Wikipedia dump by choosing the top 10k words neglecting stop words and then selecting the longest documents. We ran several main iterations of the algorithm with this dataset while fixing the parameters $k = 5$ and $\lambda = 0.5$. All timing experiments were conducted on the TACC Maverick system with Intel Xeon E5-2680 v2 Ivy Bridge CPUs (2.80 GHz), 20 CPUs per

node, and 12.8 GB memory per CPU (`https://www.tacc.utexas.edu/`).

The parallel timing results can be seen in Fig. 7.2 (left) which shows that the algorithm does have almost linear speedup when parallelizing across multiple workers. Though we only had access to a single computer with 20 processors, substantially more speed up could be obtained by using more processors on a distributed computing system. This simple parallelism makes this algorithm viable for much larger datasets. The timing results for the Wikipedia can be seen in Fig. 7.2 (right). These results give an approximate computational complexity of $O(np^2)$ which show that the proposed algorithm has the potential for scaling to datasets where n is $O(10^5)$ and p is $O(10^4)$. The $O(p^2)$ comes from the fact that there are p subproblems and each subproblem needs to calculate the gradient which is $O(p)$ as well as approximate the Newton direction for a subset of the variables. The first iteration takes longer because the initial parameter values are naïvely set to 0 whereas future iterations start from reasonable initial value.

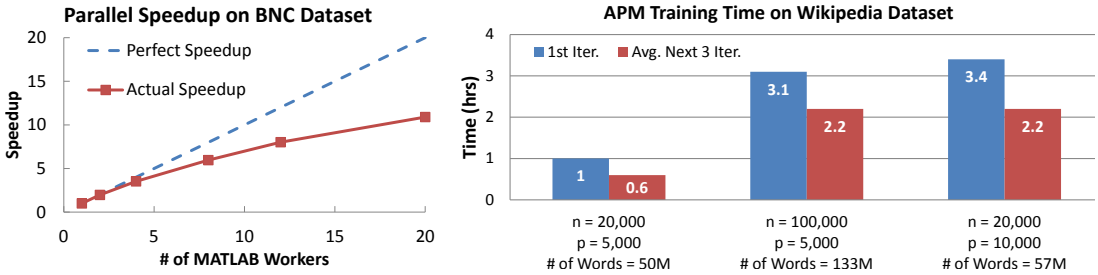


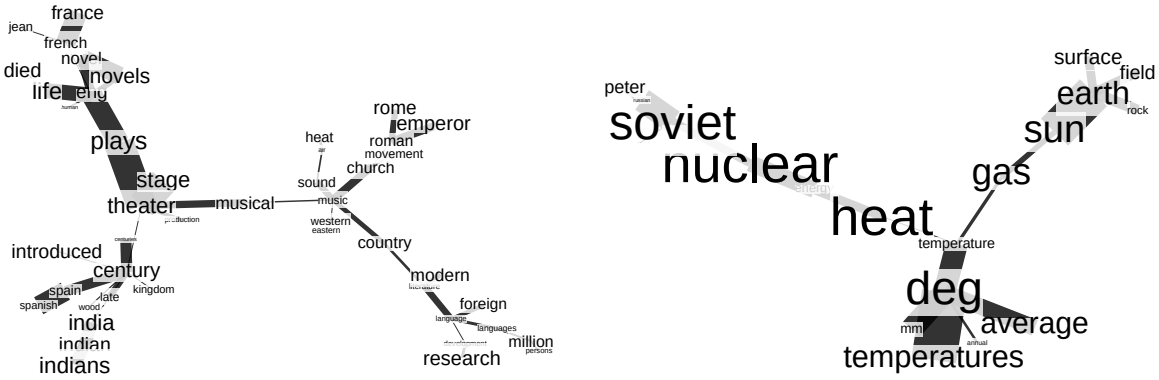
Figure 7.2: (left) The speedup on the BNC dataset shows that the algorithm scales approximately linearly with the number of workers because the subproblems are all independent. (right) The timing results on the Wikipedia dataset show that the algorithm scales to larger datasets and has a computational complexity of approximately $O(np^2)$.

7.7 Preliminary Experiments

Because previous admixture models have been independent, it is difficult to directly compare APM to previous models. Therefore, first, an experiment was conducted by running

APM (with $k = 5$ and $p = 500$) on approximately 31,000 articles of the Grolier encyclopedia.[7] Visualizations of the topics were constructed using the graph visualization program Gephi[8] in order to show some qualitative results on the model output and suggest that APM can provide a more interesting, intuitive and visually appealing representation of topics than merely a list of words as in standard topic models. Two topics of this run can be seen in Fig. 7.3 and other topic graph examples are given in the appendix. Also, a simple experiment was conducted to give some evidence, though inconclusive, that the APM model subsumes the power of the LDA model because of the model equivalence described in Sec. 7.3.1.

7.7.1 Qualitative Experiment



(a) "Music and Fine Arts" (b) "Temperature"

Figure 7.3: These APM topic visualizations illustrate that PMRFs are much more intuitive than multinomials (as in LDA/PLSA), which can only be represented as a list of words. Word size signifies relative word frequency and edge width signifies the strength of word dependency (only positive dependencies shown).

The graphs as seen in Fig. 7.3 have many interesting structural features that can

[7] www.cs.nyu.edu/~roweis/data.html

[8] www.gephi.org

be interpreted.[9] In the first example (Fig. 7.3(a)), the word "musical" is a hub word that connects the two concepts of "theatre" and "music". A similar idea happens in the second example (Fig. 7.3(b)) where "temperature" connects the concepts of "heat", "gas" and "deg".

Another interesting feature is chains of words whose endpoints are not directly related but only related through other words. For example, the chain "music" ↔ "musical" ↔ "theater" ↔ "plays" suggests that "music" and "plays" are related, albeit indirectly. The chain "sun" ↔ "gas" ↔ "temperature" ↔ "heat" ↔ "nuclear" in Fig. 7.3(b) shows the connection that the sun is related to nuclear reactions through the words "heat", "gas" and "temperature". For other chains, the endpoints are not related even though each edge seems reasonable. For example, the chain "novel" ↔ "eng" ↔ "plays" ↔ "theater" ↔ "musical" ↔ "music" ↔ "church" has logical connections for each edge but "novel" is not usually associated with "church".

Though these features give evidence for the usefulness and power of APM, they do not capture any of the negative dependencies between words—words that do *not* tend to co-occur. For example, the words "novel" and "math" would not tend to co-occur. This might be helpful in excluding documents from certain categories in document categorization. For example, if the words "history", "war" and "politics" appear in a document, the document is unlikely to be science literature. Though it may be more difficult to visualize these negative dependencies, negative dependencies can provide interesting structural information of the underlying dataset.

7.8 Evocation Metric

Boyd-Graber et al. [2006] introduced the notion of *evocation* which denotes the idea

[9]These graphs were manually filtered to simplify the whole graph. A future area of research could be to automatically filter the graph to important clusters.

of which words "evoke" or "bring to mind" other words. There can be many types of evocation including the following examples from [Boyd-Graber et al., 2006]: [rose - flower] (example), [brave - noble] (kind), [yell - talk] (manner), [eggs - bacon] (co-occurrence), [snore - sleep] (setting), [wet - desert] (antonymy), [work - lazy] (exclusivity), and [banana - kiwi] (likeness). This is distinctive from word similarity or synonymy since two words can have very different meanings but "bring to mind" the other word (e.g. antonyms). This notion of word relatedness is a much simpler but potentially more semantically meaningful and interpretable than word similarity. For instance, "work" and "lazy" do not have similar meanings but are related through the semantic meanings of the words. Another difference is that—unlike word semantic similarity— words that generally appear in very different contexts yet mean the same thing would probably not have a high evocation score. For example, "networks" and "graphs" both have a definition that means a set of nodes and edges yet usually one word is chosen in a particular context.

Recent work in evaluating topic models [Mimno et al., 2011, Newman et al., 2010, Stevens et al., 2012, Aletras and Stevenson, 2013] formulate automated metrics based on automatically scoring all pairs of top words and noticing that they correlate with human judgment of overall topic coherence. All of these metrics are based on the common assumption that a person should be able to understand a topic by understanding the abstract semantic connections between the word pairs. Thus, *evocation* is a reasonable notion for evaluating topic modeling because it directly evaluates the level of semantic connection between word pairs. In addition, this new evocation metric provides a way to explicitly evaluate the edge matrices of APM, which would be ignored in previous metrics because explicit word dependencies are not modeled in other topic models.

We now formally define our evocation metric. Given human-evaluated scores for a subset of word pairs $\mathcal{H}$ and the corresponding weights given by a topic model for this subset of word pairs $\mathcal{M}$, let us define $\pi_{\mathcal{M}}(j)$ to be an ordering of the word pairs induced by $\mathcal{M}$ such

that $\mathcal{M}_{\pi(1)} \geq \mathcal{M}_{\pi(2)} \geq \cdots \geq \mathcal{M}_{\pi(|\mathcal{H}|)}$. Then, the top-$m$ evocation metric is simply:

$$\text{Evoc}_m(\mathcal{M}, \mathcal{H}) = \sum_{j=1}^{m} \mathcal{H}_{\pi_{\mathcal{M}}(j)} \,. \tag{7.7}$$

Note that the scaling of $\mathcal{M}$ is inconsequential because $\mathcal{M}$ is only needed to define an ordering or ranking of $\mathcal{H}$. For example, $\hat{\mathcal{M}} = \alpha \exp(\mathcal{M})$ would yield the same evocation score for all scalar values $\alpha > 0$ because the ordering would be maintained. Essentially, $\mathcal{M}$ merely induces an ordering of the word pairs and the evocation score is the sum of the human scores for these top m word pairs.

For APM, the word pair weights come primarily from the PMRF edge matrices $\Phi^{1\ldots k}$—the PMRF node vectors are only used to provide an ordering if there are not enough non-zeros in the edge matrices. For the other multinomial-based topic models which do not have parameters explicitly associated with word-pairs, we can compute the most likely word pairs in a topic by multiplying their corresponding marginal probabilities. This weighting corresponds to the probability that two independent draws from the topic distribution produce the word pair and thus is the most obvious choice for multinomial-based topic models.

Since this metric only gives a way to evaluate one topic, we consider two ways of determining the overall evocation score for the whole topic model: Evoc-1 $= \sum_{j=1}^{k} \frac{1}{k} \text{Evoc}_m(\mathcal{M}^j, \mathcal{H})$ and Evoc-2 $= \text{Evoc}_m(\sum_{j=1}^{k} \frac{1}{k} \mathcal{M}^j, \mathcal{H})$. In words, these are "average evocation of topics" and "evocation of average topic" respectively. Evoc-1 measures whether all or at least most topics capture meaningful word associations since it can be affected by uninteresting topics. Evoc-2 is reasonable for measuring whether the topic model as a whole is capturing word semantics even if some of the topics are not capturing interesting word associations. This second measure has some relation to the word similarity measure of topic coherence in [Stevens et al., 2012]. However, [Stevens

et al., 2012] uses similarity rather than evocation, does not directly evaluate top individual word pairs and does not evaluate any models with word dependencies such as APM.

7.8.1 Experimental Setup

Human-Scored Evocation Dataset The original human-scored evocation dataset was produced by a set of trained undergraduates in which 1,000 words were hand selected primarily based on their frequency and usage in the British National Corpus (BNC) [Boyd-Graber et al., 2006]. From the possible pairwise evaluations, approximately 10% of the word pairs were randomly selected to be manually scored by a set of trained undergraduates. The second dataset was constructed by predicting the pairs of words that were likely to have a high evocation using a standard machine learning classifier. This new set of pairs was scored using Amazon MTurk (`mturk.com`) by using the original dataset as a control [Nikolova et al., 2009]. Though these scores are between synsets—which are a word, part-of-speech and sense triplet—, we mapped all of the synsets to word, part-of-speech pairs since that is the only information we have for the BNC corpus. This led to a total of 1646 words. In addition, though the evocation dataset has scores for directed relationships (i.e. word1 $\rightarrow$ word2 could have a different score than word2 $\rightarrow$ word1), we averaged these two scores because the directionality of the relationship is not modeled by APM or any other topic model.

BNC Corpus Because the evocation dataset was based on the BNC corpus, we used the BNC corpus for our experiments. We processed the BNC corpus by lemmatizing each word using the WordNetLemmatizer included in the nltk package (`nltk.org`) and then attaching the part-of-speech, which is already included in the BNC corpus. We only retained the counts for the 1646 words that occurred in the human-scored datasets but processed all 4049 documents in the corpus.

APM Model Parameters We trained APM on the BNC corpus with several different parameter settings including various λ and β parameter settings. We also trained two particular APM models denoted APM-LowReg and APM-HeldOut. APM-LowReg uses a very small regularization parameter so that almost all edges are non-zero. APM-HeldOut automatically selects a reasonable value for λ based on the likelihood of a held-out set of the documents. Thus, the APM-HeldOut model does not require a user-specified λ parameter but—as seen in the following sections—still performs reasonably well even compared to the APM model in which many different parameter settings are attempted. In addition, the APM-HeldOut can stop the training early when the model begins to overfit the data rather than tracing through all the λ parameters—this could lead to a significant gain in model training time. The authors suggest that APM-HeldOut is a simple baseline model for future comparison if a user does not want to specify λ.

Other Models For comparison, we trained five other models: Correlated Topic Models (CTM), Hierarchical Dirichlet Process (HDP), Latent Dirichlet Allocation (LDA), Replicated Softmax (RSM), and a naïve random baseline (RND). CTM models correlations between topics [Lafferty and D., 2006]. HDP is a non-parametric Bayesian model that selects the number of topics based input data and hyperparameters [Teh et al., 2006]. The standard topic model LDA was trained using MALLET [McCallum, 2002]. LDA was trained for at least 5,000 iterations and HDP was trained for at least 300 iterations since HDP is computationally expensive. RSM is an undirected topic model based on Restricted Boltzmann Machines (RBM) [Salakhutdinov and Hinton, 2009]. The random model is merely the expected evocation score if edges are ranked at random. We ran a full factorial experimental setting where all the combinations of a set of parameter values were trained to give a fair comparison between models (see Appendix B.3 for a summary of parameter values). All these comparison models only indirectly model dependencies between words through the latent variables since the topic distributions are multinomials whereas APM

can directly model the dependencies between words since the topic distributions are Poisson MRFs.

Selecting Best Parameters We randomly split the human scores into a 50% tuning split and 50% testing split. Note that we have a *tuning* split rather than a training split because the model training algorithms are unsupervised (i.e. they never see the human scores) so the only supervision occurs in selecting the final model parameters (i.e. during the tuning phase). Therefore, we selected the final parameters based on the tuning split and computed the final evocation scores on the test split. Thus, even when selecting the parameter settings, the modeling process never sees the test data.

7.8.2 Main Results

The Evoc-1 and Evoc-2 scores with $m = 50$ for all models can be seen in Fig. 7.4.[10] For Evoc-1, the APM models significantly outperform all other models for a small number of topics and even capture many semantically meaningful word pairs with a single topic. For higher number of topics, the APM models seem to perform only competitively with previous topic models. It seems that APM-LowReg performs better with a small number of topics whereas APM-HeldOut—which generally chooses a relatively high λ—seems more robust for large number of topics. These trends are likely caused because there is a relatively small number of documents ($n = 4049$) so the APM-LowReg begins to significantly overfit the data as the number of topics increases whereas APM-HeldOut does not seem to overfit as much. For all the APM models, the degradation in performance as the number of topics increases is most likely caused by the fact that a Poisson MRF with $O(p^2)$ parameters is a much more flexible distribution than a multinomial, and thus, fewer topics are needed to appropriately

[10]For simplicity and comparability, we grouped HDP into the topic number that was closest to its discovered number of topics because HDP can select a variable number of topics.

model the data. These results also give some evidence that APM can succinctly model the data with a much smaller number of topics than is needed for independent topic models; this succinctness could be particularly helpful for the interpretability and intuitions of topic models.

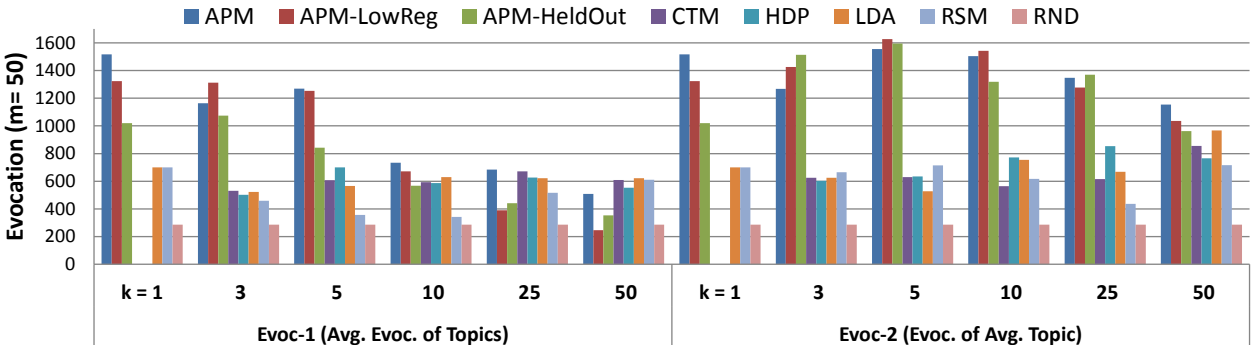


Figure 7.4: Both Evoc-1 scores (left) and Evoc-2 scores (right) demonstrate that APM usually significantly outperforms other topic models in capturing meaningful word pairs.

For the Evoc-2 score, the APM models—including the APM-HeldOut model which automatically determines a λ from the data—significantly outperform previous topic models even for a large number of topics. This supports the idea that APM only needs a small number of topics to capture many of the semantically meaningful word dependencies. Thus, when increasing the number of topics beyond 5, the performance does not decrease as in Evoc-1. It is likely that this discrepancy is caused by the fact that many of the edges are concentrated in a small number of topics even when the number of topics is 10 or 25. As expected because of previous research in topic models, most other topic models perform slightly better with a larger number of topics. Though it is possible that using 100 or 500 topics for these topic models might give an evocation score better than APM with 5 topics, this would only enforce the idea that APM can perform better or at least competitively with previous topic models while only using a comparatively small number of topics.

Qualitative Analysis of Top 20 Word Pairs for Best LDA and APM Models To validate the intuition of using evocation as an human-standard evaluation metric, we present the top 20 word pairs for the best standard topic model—in this case LDA—and the best APM model for the Evoc-2 metric as seen in Table 7.1. The best performing LDA model was trained with 50 topics, $\alpha = 1$ and $\beta = 0.0001$. The best APM model was the APM-LowReg model trained with only 5 topics and a small regularization parameter $\lambda = 0.05$. It is important to note that the best model for LDA has 50 topics while the best model for APM only has 5 topics. As before, this reinforces the theme that APM can capture more semantically meaningful word pairs with a smaller number of topics than previous topic models.

Table 7.1: Top 20 words for LDA (left) and APM (right)

Human Score	Word Pair	Human Score	Word Pair
100	run.v ↔ car.n	38	woman.n ↔ man.n
82	teach.v ↔ school.n	38	give.v ↔ church.n
69	school.n ↔ class.n	38	wife.n ↔ man.n
63	van.n ↔ car.n	38	engine.n ↔ car.n
51	hour.n ↔ day.n	35	publish.v ↔ book.n
50	teach.v ↔ student.n	32	west.n ↔ state.n
44	house.n ↔ government.n	32	year.n ↔ day.n
44	week.n ↔ day.n	25	member.n ↔ give.v
38	university.n ↔ institution.n	25	dog.n ↔ animal.n
38	state.n ↔ government.n	25	seat.n ↔ car.n

Human Score	Word Pair	Human Score	Word Pair
100	telephone.n ↔ call.n	57	question.n ↔ answer.n
97	husband.n ↔ wife.n	57	prison.n ↔ cell.n
82	residential.a ↔ home.n	51	mother.n ↔ baby.n
76	politics.n ↔ political.a	50	sun.n ↔ earth.n
75	steel.n ↔ iron.n	50	west.n ↔ east.n
75	job.n ↔ employment.n	44	weekend.n ↔ sunday.n
75	room.n ↔ bedroom.n	41	wine.n ↔ drink.v
72	aunt.n ↔ uncle.n	38	south.n ↔ north.n
72	printer.n ↔ print.v	38	morning.n ↔ afternoon.n
60	love.v ↔ love.n	38	engine.n ↔ car.n

As seen in Table 7.1, APM captures many more word pairs with a human score greater than 50, whereas LDA only captures a few. One interesting example is that LDA finds two word pairs [woman.n - wife.n] and [wife.n - man.n] that capture some semantic notion of marriage. However, APM directly captures this semantic meaning with [husband.n - wife.n]. APM also discovers several other familial relationships such as [aunt.n - uncle.n] and [mother.n - baby.n]. In addition, APM identifies multiple semantically coherent yet high-level word pairs such as [residential.a - home.n], [steel.n - iron.n], [job.n - employment.n] and [question.n - answer.n], whereas LDA finds several low-level word pairs such as [member.n - give.v], [west.n - state.n] and [year.n - day.n]. These overall trends become even more

evident when looking at the top 50 word pairs as can be found in Appendix B.5. Both the quantitative evaluation metrics (i.e. Evoc-1 and Evoc-2) as well as a qualitative exploration of the top word pairs give strong evidence that APM can succinctly capture both more interesting and higher-level semantic concepts through word dependencies than previous independent topic models.

7.9 Related Work

Many probabilistic models for documents have been constructed using the multinomials. Nigam et al. [2000] introduced a mixture of multinomials to model document collections, and later, Hofmann [1999] proposed an admixture of multinomials called Probabilistic Latent Semantic Analysis (PLSA). This model was followed by the very successful Latent Dirichlet Allocation (LDA) topic model proposed by Blei et al. [2003] that added priors to the distributions as well as provided a more coherent framework for extending the model. There have been numerous extensions of LDA that incorporate other knowledge such as author information [Steyvers et al., 2004], time [Blei et al., 2006] and topic dependency [Lafferty and D., 2006]. However, none of these models considers dependencies between words since the base distribution is multinomial.

Replicated Softmax [Salakhutdinov and Hinton, 2009] uses a restricted Boltzmann machine (RBM) with parameter biases to create a generative model for word count vectors. The hidden layer is binary-valued and allows for topic parameters to be mixed in the canonical parameter space (similar to APM). *Wordfish* [Slapin and Proksch, 2008] is a Poisson IRT (Item Response Theory) model that attempts to characterize the latent position of a political party based on political manifestos (e.g. determining left or right wing political views). Though *Wordfish* also adds fixed-effect parameters, this model is similar to an independent APM model with $k = 2$ (i.e. only one latent dimension). Both *Replicated Softmax* and *Wordfish* significantly differ from APM because they do not

consider word dependencies.

Sparse Word Graphs [Nallapati et al., 2007] attempts to create graph visualizations of the topics by combining LDA and Bernoulli MRFs (Ising model) in a two-stage approach. First, LDA is used to estimate the topic assignments for every word in the corpus. Then, these topic assignments are used to train k independent Bernoulli MRFs for each topic. To transform the LDA output into the input for the Bernoulli MRF estimation algorithm, binary word-document matrices are constructed for each topic based on the LDA topic assignments. Though this leads to a graph over words for each topic, one major difference with APM is that this two-stage method is not a unified probabilistic model but rather two separate probability models. Another significant difference is that *Sparse Word Graphs* estimates simpler Bernoulli MRFs instead of PMRFs as in APM.

In [Pleple, 2013], users can interactively add soft constraints to LDA so that the probability of the words in the constraint set will tend to be similar (e.g. either all low or all high probability). The soft dependency is added through a latent constraint variable and only provides indirect dependence of words rather than direct dependence between words as in APM. Another difference is that these constraints can only be supplied as user-specified disjoint groups of words rather than automatically-discovered arbitrary structure as in APM.

Collins and Schapire [2001] develop a generalization of PCA by using the likelihood of exponential families as the loss function instead of squared loss—which would correspond to Gaussian errors. While exponential PCA is related to admixtures, it does not place constraints on the admixture weights but rather allows them to be arbitrary real numbers. This is analogous to the difference between SVD and constrained non-negative matrix factorization (NMF).

7.10 Conclusion

We motivated the need for more expressive topic models that consider word dependencies—such as APM—by considering previous work on topic model evaluation metrics. We developed a novel topic model based on an admixture of Poisson MRFs that can model dependencies between words unlike all previous topic models that assume word independence. Independent Poisson MRFs are shown to generalize the conditional distributions of LDA, which thus suggests that APM subsumes the expressive power of LDA and adds significantly greater modeling power than LDA. In addition to APM, a generalized class of admixture models is defined which opens the way for admixtures of any parametric distribution. We overcame the significant computational barrier of estimating the parameters of APM by providing a fast alternating Newton-like algorithm which can be easily parallelized. We also proposed a new evaluation metric based on human evocation scores that seeks to measure whether a model is capturing semantically meaningful word pairs. Finally, we presented compelling quantitative and qualitative measures showing the superiority of APM in capturing semantically meaningful word pairs. In addition, this metric suggests new evaluations of topic models based on evaluating top word pairs rather than top words. One drawback with the current human-scored data is that only a small portion of the word pairs have been scored. Thus, one extension is to dynamically collect more human scores as needed for evaluation. The development of APM opens up a whole new area of research with many interesting open questions in both theory (e.g. scalability, other admixtures, hyperparameter choice) and applications (e.g. visualization, user interaction, document exploration). This work also opens the door for exciting new word-semantic applications for APM such as Word Sense Induction using topic models [Lau et al., 2012], keyword expansion or suggestion, document summarization, and document visualization because APM is capturing semantically meaningful relationships between words.

Chapter 8

Fixed-Length Poisson MRF Topic Models[1]

8.1 Abstract

We propose a novel topic model that uses LPMRF graphical model (see Chapter 3) as a base distribution for the second topic model generalization described in Chapter 6. We show the effectiveness of our LPMRF distribution over multinomial models by evaluating the test set perplexity on a dataset of abstracts and Wikipedia. Qualitatively, we show that the positive dependencies discovered by LPMRF are interesting and intuitive. Finally, we show that our algorithms are fast and have good scaling. (code available online)

8.2 Related Work

In [Nallapati et al., 2007], the LDA topic assignments for each word are used to train a separate Ising model—i.e. a Bernoulli MRF—for each topic in a heuristic two-stage procedure. Instead of modeling dependencies *a posteriori*, we formulate a generalization of topic models that allows the LPMRF distribution to directly replace the multinomial. This allows us to compute a topic model and word dependencies *jointly* under a *unified* model as opposed to the two-stage heuristic procedure in [Nallapati et al., 2007].

[1]The majority of this chapter is from [Inouye et al., 2015] with some edits for better integration into this dissertation. [Inouye et al., 2015] was primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

8.3 LPMRF Topic Model

8.3.1 Optimization Problem

Using the second topic model generalization described in Chapter 6, we can create a joint optimization problem to solve for the topic matrix $\boldsymbol{Z}_i = [\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, \ldots, \boldsymbol{z}_i^k]$ for each document and to solve for the shared LPMRF parameters $\boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k}$. The optimization is based on minimizing the negative log posterior:

$$\begin{aligned}\underset{\boldsymbol{Z}_{1\ldots n}, \boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k}}{\arg\min} \; & -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\log(\mathbb{P}_{\text{LPMRF}}(\boldsymbol{z}_i^j|\boldsymbol{\theta}^j, \Phi^j, m_i^j)) \\ & -\sum_{i=1}^{n}\log(\mathbb{P}_{\text{prior}}(m_i^{1\ldots k})) - \sum_{j=1}^{k}\log(\mathbb{P}_{\text{prior}}(\boldsymbol{\theta}^j, \Phi^j)) \\ \text{s.t.} \quad & \boldsymbol{Z}_i\mathbf{e} = \boldsymbol{x}_i, \quad \boldsymbol{Z}_i \in \mathbb{Z}_+^{k\times p},\end{aligned}$$

where $\mathbf{e}$ is the all ones vector. Notice that the observations $\boldsymbol{x}_i$ only show up in the constraints. The prior distribution on $m_i^{1\ldots k}$ can be related to the Dirichlet distribution as in LDA by taking $\mathbb{P}_{\text{prior}}(m_i^{1\ldots k}) = \mathbb{P}_{\text{Dir}}(m_i^j / \sum_\ell m_i^\ell | \boldsymbol{\alpha})$. Also, notice that the documents are all independent if the LPMRF parameters are known so this optimization can be trivially parallelized.

8.3.2 Connection to Collapsed Gibbs Sampling

This optimization is very similar to the collapsed Gibbs sampling for LDA [Steyvers and Griffiths, 2007]. Essentially, the key part to estimating the topic models is estimating the topic indicators for each word in the corpus. The model parameters can then be estimated directly from these topic indicators. In the case of LDA, the multinomial parameters are trivial to estimate by merely keeping track of counts and thus the parameters can be updated in constant time for every topic resampled. This also suggests that an interesting area of future work would be to understand the connections between collapsed Gibbs sampling and

this optimization problem. It may be possible to use this optimization problem to speed up Gibbs sampling convergence or provide a MAP phase after Gibbs sampling to get non-random estimates.

8.3.3 Estimating Topic Matrices $\boldsymbol{Z}_{1...n}$

For LPMRF topic models, the estimation of the LPMRF parameters given the topic assignments requires solving another complex optimization problem. Thus, we pursue an alternating EM-like scheme as in LPMRF mixtures. First, we estimate LPMRF parameters with the PMRF algorithm from [Inouye et al., 2014a], and then we optimize the topic matrix $\boldsymbol{Z}_i \in \mathbb{R}^{p \times k}$ for each document. Because of the constraints on $\boldsymbol{Z}_i$, we pursue a simple dual coordinate descent procedure. We select two coordinates in row r of $\boldsymbol{Z}_i$ and determine if the optimization problem can be improved by moving a words from topic ℓ to topic q. Thus, we only need to solve a series of simple univariate problems. Each univariate problem only has x_{is} number of possible solutions and thus if the max count of words in a document is bounded by a constant, the univariate subproblems can be solved efficiently. More formally, we are seeking a step size a such that $\widehat{\boldsymbol{Z}}_i = \boldsymbol{Z}_i + a\boldsymbol{e}_r\boldsymbol{e}_\ell^T - a\boldsymbol{e}_r\boldsymbol{e}_q^T$ gives a better optimization value than $\boldsymbol{Z}_i$. If we remove constant terms w.r.t. a, we arrive at the following univariate optimization problem (suppressing dependence on i because each of the n subproblems are independent):

$$\begin{aligned}\underset{-z_r^\ell \le a \le z_r^q}{\arg\min}\ & -a[\theta_r^\ell - \theta_r^q + 2\boldsymbol{z}_\ell^T\Phi_r^\ell - 2\boldsymbol{z}_q^T\Phi_r^q] + [\log((z_r^\ell + a)!) + \log((z_r^q - a)!)] \\ & + A_{m^\ell + a}(\boldsymbol{\theta}^\ell, \Phi^\ell) + A_{m^q + a}(\boldsymbol{\theta}^q, \Phi^q) - \log(\mathbb{P}_{\text{prior}}(\tilde{m}^{1...k})),\end{aligned}$$

where $\tilde{m}$ is the new distribution of length based on the step size a. The first term is the linear and quadratic term from the sufficient statistics. The second term is the change in base measure of a word is moved. The third term is the difference in log partition function if the length of the topic vectors changes. Note that the log partition function can be precomputed

so it merely costs a table lookup. The prior also only requires a simple calculation to update. Thus the main computation comes in the inner product $\boldsymbol{z}_\ell^T \Phi_r^\ell$. However, this inner product can be maintained very efficiently and updated efficiently so that it does not significantly affect the running time.

8.4 Perplexity Experiments

We evaluated our novel LPMRF model using perplexity on a held-out test set of documents from a corpus composed of research paper abstracts[2] denoted Classic3 and a collection of Wikipedia documents. The Classic3 dataset has three distinct topic areas: medical (Medline, 1033), library information sciences (CISI, 1460) and aerospace engineering (CRAN, 1400). More details about the Classic3 dataset can be found in Chapter 5 Table 5.1.

8.4.1 Experimental Setup

We train all the models using a 90% training split of the documents and compute the held-out perplexity on the remaining 10% where perplexity is equal to $\exp(-\mathcal{L}(X^{\text{test}}|\boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k})/N_{\text{test}})$, where $\mathcal{L}$ is the log likelihood and N_{test} is the total number of words in the test set. We evaluate single, mixture and topic models with both the multinomial as the base distribution and LPMRF as the base distribution at $k = \{1, 3, 10, 20\}$. The topic indicator matrices $\boldsymbol{Z}_i$ for the test set are estimated by fitting a MAP-based estimate while holding the topic parameters $\boldsymbol{\theta}^{1\ldots k}, \Phi^{1\ldots k}$ fixed.[3] For a single multinomial or LPMRF, we set the smoothing parameter β to 10^{-4}.[4] We select the LPMRF models using all combinations of 20 log spaced λ between 1 and 10^{-3}, and 5

[2] `http://ir.dcs.gla.ac.uk/resources/test_collections/`

[3] For topic models, the likelihood computation is intractable if averaging over all possible $\boldsymbol{Z}_i$. Thus, we use a MAP simplification primarily for computational reasons to compare models without computationally expensive likelihood estimation.

[4] For the LPMRF, this merely means adding 10^{-4} to y-values of the node-wise Poisson regressions.

linearly spaced weighting function constants c between 1 and 2 for the weighting function described in Sec. 3.3.1. In order to compare our algorithms with LDA, we also provide perplexity results using an LDA Gibbs sampler [Steyvers and Griffiths, 2007] for MATLAB [5] to estimate the model parameters. For LDA, we used 2000 iterations and optimized the hyperparameters α and β using the likelihood of a tuning set. We do not seek to compare with many topic models because many of them use the multinomial as a base distribution which could be replaced by a LPMRF but rather we simply focus on simple representative models.[6]

8.4.2 Perplexity Results

The perplexity results for all models can be seen in Fig. 8.1. Clearly, a single LPMRF significantly outperforms a single multinomial on the test dataset both for the Classic3 and Wikipedia datasets. The LPMRF model outperforms the simple multinomial mixtures and topic models in all cases. This suggests that the LPMRF model could be an interesting replacement for the multinomial in more complex models. For a small number of topics, LPMRF topic models also outperforms Gibbs sampling LDA but does not perform as well for larger number of topics. This is likely due to the well-developed sampling methods for learning LDA. Exploring the possibility of incorporating sampling into the fitting of the LPMRF topic model is an excellent area of future work. We believe LPMRF shows significant promise for replacing the multinomial in various probabilistic models.

[5] `http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm`

[6] We could not compare to APM [Inouye et al., 2014b,a] because it is not computationally tractable to calculate the likelihood of a test instance in APM, and thus we cannot compute perplexity.

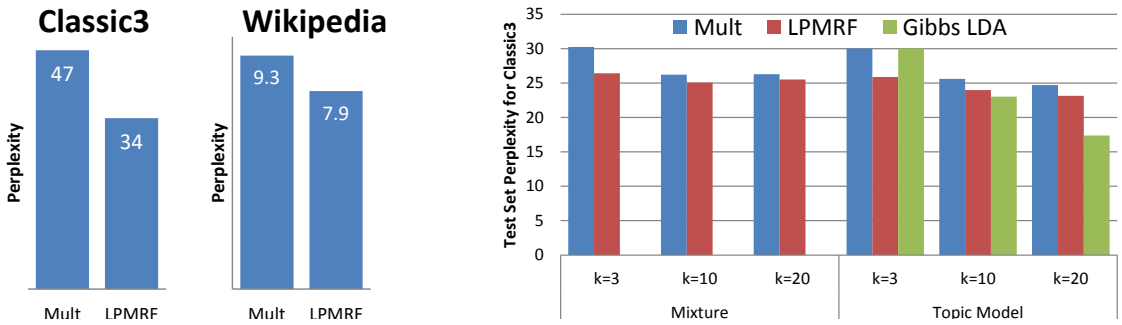


Figure 8.1: (Left) The LPMRF models quite significantly outperforms the multinomial for both datasets. (Right) The LPMRF model outperforms the simple multinomial model in all cases. For a small number of topics, LPMRF topic models also outperforms Gibbs sampling LDA but does not perform as well for larger number of topics.

8.5 Qualitative Analysis of LPMRF Parameters

In addition to perplexity analysis, we present the top words, top positive dependencies and the top negative dependencies for the LPMRF topic model in Table 8.1. Notice that in LDA, only the top words are available for analysis but an LPMRF topic model can produce intuitive dependencies. For example, the positive dependency "language+natural" is composed of two words that often co-occur in the library sciences but each word independently does not occur very often in comparison to "information" and "library". The positive dependency "stress+reaction" suggests that some of the documents in the Medline dataset likely refer inducing stress on a subject and measuring the reaction. Or in the aerospace topic, the positive dependency "non+linear" suggests that non-linear equations are important in aerospace. Notice that these concepts could not be discovered with a standard multinomial-based topic model.

8.6 Timing and Scalability

Finally, we explore the practical performance of our algorithms. In C++, we implemented the three core algorithms: fitting p Poisson regressions, fitting the n topic

Table 8.1: Top Words and Dependencies for LPMRF Topic Model

Topic 1			Topic 2			Topic 3		
Top words	*Top Pos. Edges*	*Top Neg. Edges*	*Top words*	*Top Pos. Edges*	*Top Neg. Edges*	*Top words*	*Top Pos. Edges*	*Top Neg. Edges*
information	states+united	paper-book	patients	term+long	cells-patient	flow	supported+simply	flow-shells
library	point+view	libraries-retrieval	cases	positive+negative	patients-animals	pressure	account+taken	number-numbers
research	test+tests	library-chemical	normal	cooling+hypothermi	patients-rats	boundary	agreement+good	flow-shell
system	primary+secondary	libraries-language	cells	system+central	hormone-protein	results	moment+pitching	wing-hypersonic
libraries	recall+precision	system-published	treatment	atmosphere+height	growth-parathyroid	theory	non+linear	solutions-turbulent
book	dissemination+sdi	information-citations	children	function+functions	patients-lens	method	lower+upper	mach-reynolds
systems	direct+access	information-citation	found	methods+suitable	patients-mice	layer	tunnel+wind	flow-stresses
data	language+natural	chemical-document	results	stress+reaction	patients-dogs	given	time+dependent	theoretical-drag
use	years+five	library-scientists	blood	low+rates	hormone-tumor	number	level+noise	general-buckling
scientific	term+long	library-scientific	disease	case+report	patients-child	presented	purpose+note	made-conducted

matrices for each document, and sampling 5,000 AIS samples. The timing for each of these components respectively can be seen in Fig. 8.2 for the Wikipedia dataset. We set $\lambda = 1$ in the first two experiments which yields roughly 20,000 non-zeros and varied λ for the third experiment. Each of the components is trivially parallelized using OpenMP (`http://openmp.org/`). All timing experiments were conducted on the TACC Maverick system with Intel Xeon E5-2680 v2 Ivy Bridge CPUs (2.80 GHz), 20 CPUs per node, and 12.8 GB memory per CPU (`https://www.tacc.utexas.edu/`). The scaling is generally linear in the parameters except for fitting topic matrices which is $O(k^2)$. For the AIS sampling, the scaling is linear in the number of non-zeros in Φ irrespective of p. Overall, we believe our implementations provide both good scaling and practical performance (code available online).

8.7 Conclusion

We extend the LPMRF distribution to topic models by using fixed-length distributions for the second generalization of Chapter 6 and develop parameter estimation methods using dual coordinate descent. We evaluate the perplexity of the proposed LPMRF models on datasets and show that they offer good performance when compared to

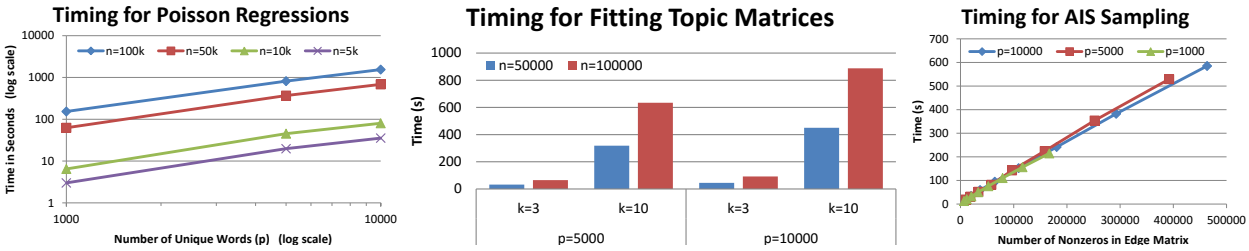


Figure 8.2: (Left) The timing for fitting p Poisson regressions shows an empirical scaling of $O(np)$. (Middle) The timing for fitting topic matrices empirically shows scaling that is $O(npk^2)$. (Right) The timing for AIS sampling shows that the sampling is approximately linearly scaled with the number of non-zeros in Φ irrespective of p.

multinomial-based models. Finally, we show that our algorithms are fast and have good scaling. Potential new areas could be explored such as the relation between the topic matrix optimization method and Gibbs sampling. It may be possible to develop sampling-based methods for the LPMRF topic model similar to Gibbs sampling for LDA.

Part III

Generalizing Graphical Models

Summary of Part III

In the next two chapters, we present two more general graphical models. First, we propose a novel graphical model for k-wise dependencies instead of merely pairwise dependencies as in Chapter 4. As an example of a triple-wise dependency in the case of text analysis, there may be three words that often co-occur together such as "deep", "neural" and "network". Or in biology, there may be three or more proteins that naturally interact because they belong in the same protein complex. The key idea is to take the k-root of the original sufficient statistics to ensure that the distribution can be normalized even with k-wise interactions.

Second, we further investigate the Gaussian-copula model paired with arbitrary marginals as described in Sec. 5.3.2. Because of its special form, the Gaussian-copula model can be seen as a semi-parametric graphical model [Liu et al., 2012]. While the graphical models described in Part I assumed that the univariate conditionals were in the exponential family, Gaussian-copula models allow for the univariate marginal distributions to be outside of the exponential family—e.g. the t distribution. Thus, the Gaussian-copula model can be seen as a generalized semi-parametric graphical model that allows arbitrary marginal distributions. While this model is well-known, we derive the closed-form solutions to the conditional distributions; more specifically, we show that the conditional of a Gaussian-copula model is another Gaussian-copula model with closed-form conditional marginals. We demonstrate the usefulness of these closed-form solutions via multiple missing value imputation experiments. In addition, these closed-form solutions provide the foundation for the graphical model visualization developed in Part IV.

Chapter 9

General Graphical Models Beyond Pairwise Dependencies[1]

9.1 Abstract

We present a novel k-way high-dimensional graphical model called the Generalized Root Model (GRM) that explicitly models dependencies between variable sets of size $k \geq 2$—where $k = 2$ is the standard pairwise graphical model. This model is based on taking the k-th root of the original sufficient statistics of any univariate exponential family with positive sufficient statistics, including the Poisson and exponential distributions. As in the recent work with square root graphical (SQR) models [Inouye et al., 2016a]—which was restricted to pairwise dependencies—we give the conditions of the parameters that are needed for normalization using the radial conditionals similar to the pairwise case [Inouye et al., 2016a]. In particular, we show that the Poisson GRM has no restrictions on the parameters and the exponential GRM only has a restriction akin to negative definiteness. We develop a simple but general learning algorithm based on ℓ_1-regularized node-wise regressions. We also present a general way of numerically approximating the log partition function and associated derivatives of the GRM univariate node conditionals—[Inouye et al., 2016a] only provided algorithm for estimating exponential SQR. To illustrate GRM, we model word counts with a Poisson GRM and show the associated k-sized variable sets. We discuss methods for reducing

[1]The majority of this chapter is from [Inouye et al., 2016b] with some edits for better integration into this dissertation. [Inouye et al., 2016b] was primarily executed and authored by David Inouye with guiding contributions and edits by the co-authors.

the parameter space in various situations.

9.2 Introduction

Most standard graphical models are restricted to pairwise dependencies between variables. For example, the Ising model for binary data and the multivariate Gaussian for real-valued data are popular pairwise graphical models. However, real-world data often exhibits triple-wise, or more generally k-wise dependencies. For example, the words *deep*, *neural* and *network* often occur together in recent research papers—note that this *triple* of words refers to something more specific than any of the two words without the third word, i.e. if a document only contains *neural* and *network* but not *deep*, then this may be a more classical paper about shallow neural networks. In the biological domain, genetic, metabolic and protein pathways play an important role in studying the development of diseases and possible interventions. These pathways are known to be complex and involve many genes or proteins rather than just simple pairwise interactions.[2]

Thus, we seek to begin bridging this gap between pairwise models and complex real-world data that contain complex k-wise interactions by defining a class of k-wise graphical models called Generalized Root Models (GRM), which can be instantiated for any $k \geq 1$ and any univariate exponential family with positive sufficient statistics including the Gaussian (using the x^2 sufficient statistic), Poisson and exponential distributions. We estimate the graphical model structure and parameters using ℓ_1-regularized node-wise regressions similar to previous work [Ravikumar et al., 2010, Yang et al., 2015, Inouye et al., 2015, 2016a]. However, unlike previous work, because the log partition function of the GRM node conditionals is not known in closed-form—even for the previous work considering the pairwise case[Inouye et al., 2016a]—we develop a novel numerical

[2] `https://www.genome.gov/27530687/`

approximation method for the GRM log partition function and related derivatives. In addition, we present a Newton-like optimization algorithm similar to [Hsieh et al., 2011] to solve the node-regressions—which significantly reduces the number of numerical log partition function approximations needed compared to gradient descent. Finally, we demonstrate the GRM model and parameter estimation algorithm on real-world text data.

9.3 Related Work

This proposed work generalizes the square root graphical model (SQR) from Chapter 4 [Inouye et al., 2016a], which only considers pairwise dependencies. Though SQR models have great promise, SQR models are limited to pairwise dependencies, and we had not provided an estimation algorithm for the Poisson SQR model in Chapter 4 because the node conditional log partition function is not known in closed form. Thus, this proposed work extends the SQR model class to include k-wise interactions where $k > 2$ and, in addition, instantiates a concrete approximation algorithm for the node conditional log partition function and associated derivatives.

In a somewhat different direction, latent variable models provide an implicit and indirect way of modeling complex dependencies. Generally, though the explicit dependencies in latent variable models are only pairwise, many variables can be related implicitly through a latent variable. For example, mixture models associate a discrete latent variable with every instance which implicitly introduces dependencies. Other more complex latent variable models such as topic models [Blei et al., 2010, Lafferty and D., 2006] can introduce even more implicit dependencies in interesting ways. While latent variable models have proven to be practically effective in helping to model complex dependencies, the development of GRM models in this paper is distinctive and somewhat orthogonal to latent variable models. As opposed to implicitly modeling dependencies through latent variables, the GRM model explicitly models dependencies between observed

variables. Thus, the discovered dependencies have an intuitive and obvious explanation in terms of the data variables. In addition, GRM models can be seen as complementary to latent variable models because GRM models can be used as base distributions for these latent variable models. For example, [Inouye et al., 2014b, 2015] explore using count-valued graphical models in mixtures and topic models. Thus, GRMs can provide new components from which to build more interesting models for real-world situations. Finally, node-conditional models such as GRM can be estimated using convex optimization problems and often have theoretical guarantees [Ravikumar et al., 2010, Yang et al., 2015] whereas latent variable models often require optimizing a non-convex function and struggle with theoretical guarantees.

9.3.1 Tensor and Outer Product Notation

We denote tensors (or multidimensional arrays) with parenthesized superscripts as $X^{(k)}$ where k is the order of the tensor. For example, $A^{(2)} \in \mathbb{R}^{p \times p}$ is a matrix, $A^{(3)} \in \mathbb{R}^{p \times p \times p}$ is a three dimensional tensor, and $A^{(k)} \in \mathbb{R}^{p \times^k}$ is a k-th order tensor. We index tensors using brackets and subscripts, e.g. $[A^{(3)}]_{1,2,3}$ is a scalar value in the multidimensional array at index $(1, 2, 3)$. We define $[A^{(\ell)}]_s \in \mathbb{R}^{p \times^{\ell-1}}$ to be a sub tensor created by fixing the last index to s and letting the others vary—in MATLAB colon indexing notation, this would be $A(:, :, \ldots, :, \mathrm{s})$. For example, if $A^{(3)} \in \mathbb{R}^{p \times^3}$, then $[A^{(3)}]_s \in \mathbb{R}^{p \times p}$ is a matrix corresponding to the s-th slice of the tensor $A^{(3)}$.

We define $\circ$ to be the outer product operation. For example, $\boldsymbol{x} \circ \boldsymbol{x} = \boldsymbol{x}\boldsymbol{x}^T \in \mathbb{R}^{p \times p}$ and $\boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x} \in \mathbb{R}^{p \times^3}$, where $[\boldsymbol{x} \circ \boldsymbol{x} \circ \boldsymbol{x}]_{s_1 s_2 s_3} = x_{s_1} x_{s_2} x_{s_3}$. For more general sizes, we denote a k-th outer product to be $\boldsymbol{x} \circ^k = \boldsymbol{x} \circ \cdots \circ \boldsymbol{x}$ such that there are k copies of $\boldsymbol{x}$ and the result is a k-th order tensor. We define $\boldsymbol{x} \circ^0 = \boldsymbol{e} = [1, 1, \cdots, 1]^T$. We also denote the inner product operation of two tensors as $\left\langle A^{(k)}, B^{(k)} \right\rangle = \sum_{s_1, \cdots, s_k} a_{s_1, \cdots, s_k} b_{s_1, \cdots, s_k}$.

9.4 Generalized Root Model

With the notation given in the previous section, we will define the GRM model. First, let the sufficient statistic and log base measure of a univariate exponential family be denoted as $T(x)$ and $B(x)$ respectively. We will also define the domain (or support) of the random variable to be $\mathcal{D}$ and it's corresponding measure to be $\mu(x)$, which is either the counting measure or Lebesgue measure depending on whether x is discrete or continuous.

Let us denote a new j-th root sufficient statistic $\widetilde{T}_j(x) = \sqrt[j]{T(x)}$ except in the case when $T(x) = f(x)^{cj}$ where c is an even positive integer. If $T(x) = f(x)^{cj}$, then we simplify $\widetilde{T}_j(x) \equiv f(x)^c$ (rather than the usual $|f(x)|^c$). For example, if $T(x) = x^2$, then $\widetilde{T}_2(x) \equiv x$ (rather than $|x|$). As in [Inouye et al., 2016a], this nuanced definition is necessary to recover the multivariate Gaussian distribution. However, for notational simplicity, we will merely write $\sqrt[j]{x}$ for $\widetilde{T}_j(x)$ throughout the paper. Note that $\widetilde{T}_j(x) = \sqrt[j]{x}$ for the Poisson and exponential GRM models. Using this simplified notation, we can define the Generalized Root Model for $k \leq p$ as:

$$\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{\Psi}_{(\cdot)}^{(\cdot)}) = \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\left\langle \Psi_{(j)}^{(\ell)}, \sqrt[j]{\boldsymbol{x}}\circ^{\ell}\right\rangle + \textstyle\sum_s B(x_s) - A(\boldsymbol{\Psi}_{(\cdot)}^{(\cdot)})\right) \tag{9.1}$$

$$A(\boldsymbol{\Psi}_{(\cdot)}^{(\cdot)}) = \log\int_{\mathcal{D}}\exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\left\langle \Psi_{(j)}^{(\ell)}, \sqrt[j]{\boldsymbol{x}}\circ^{\ell}\right\rangle + \textstyle\sum_s B(x_s)\right)\mathrm{d}\mu(\boldsymbol{x})\,, \tag{9.2}$$

where $A(\boldsymbol{\Psi}_{(\cdot)}^{(\cdot)})$ is the joint log partition function, $\boldsymbol{\Psi}_{(\cdot)}^{(\cdot)} = \left\{\Psi_{(j)}^{(\ell)} : j \in \{1,\cdots,k\}, \ell \leq j\right\}$, $\Psi_{(j)}^{(\ell)}$ are super symmetric tensors of order ℓ which are zero whenever two indices are the same. More formally, letting $\pi(\cdot)$ be an index permutation:

$$\Psi_{(j)}^{(\ell)} \in \left\{ A^{(\ell)} \;:\; \begin{array}{lll} \left[A^{(\ell)}\right]_{s_1,\cdots,s_\ell} & = [A^{(\ell)}]_{\pi(s_1,\cdots,s_\ell)} & \forall\pi(\cdot), \\ \left[A^{(\ell)}\right]_{\pi(s_u,s_v,\cdots,s_\ell)} & = 0 & \forall\{(u,v,\pi(\cdot)) : u \neq v, s_u = s_v\} \end{array} \right\}. \tag{9.3}$$

Note that the non-zeros of $\Psi_{(j)}^{(\ell)}$ define ℓ-sized variable sets (or cliques) of the underlying graphical model.

9.4.0.1 Special Cases

We now consider several special cases of this model to build some understanding of the GRMs connection to previous models. The independent model is trivially recovered if $k = 1$: $\mathbb{P}(\boldsymbol{x} \mid \Psi^{(1)}_{(1)}) = \exp\left(\langle \Psi^{(1)}_{(1)}, \boldsymbol{x}\rangle + \sum_s B(x_s) - A(\boldsymbol{\Psi}^{(\cdot)}_{(\cdot)})\right)$.

9.4.1 Square Root Graphical Model [Inouye et al., 2016a]

Another special case is the previous SQR models (i.e. $k = 2$) from [Inouye et al., 2016a] by taking (using the notation from [Inouye et al., 2016a]) $\Psi^{(1)}_{(1)} = \text{diag}(\Phi)$, $\Psi^{(1)}_{(2)} = \boldsymbol{\theta}$ and $\Psi^{(2)}_{(2)} = \tilde{\Phi}$, where $\text{diag}(\Phi)$ is a column vector of the diagonal entries and $\tilde{\Phi}$ has the same off-diagonal entries as Φ but is zero along the diagonal. Thus, the SQR model can be written as:

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x} \mid \Psi^{(1)}_{(1)}, \Psi^{(1)}_{(2)}, \Psi^{(2)}_{(2)}) \\
&\quad = \exp\left(\langle \Psi^{(1)}_{(1)}, \boldsymbol{x}\rangle + \langle \Psi^{(1)}_{(2)}, \sqrt[2]{\boldsymbol{x}}\rangle + \langle \Psi^{(2)}_{(2)}, \sqrt[2]{\boldsymbol{x}} \circ \sqrt[2]{\boldsymbol{x}}\rangle + \textstyle\sum_s B(x_s) - A(\boldsymbol{\Psi}^{(\cdot)}_{(\cdot)})\right) .
\end{aligned}$$

9.4.2 Simplified Model with Only Strongest Interaction Terms

We consider another special case such that only the strongest interaction (i.e. when $\ell = j$) terms are non-zero:

$$\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) = \exp\left(\sum_{j=1}^{k} \langle \Psi^{(j)}_{(j)}, \sqrt[j]{\boldsymbol{x}} \circ^j \rangle \; + \; \textstyle\sum_s B(x_s) - A(\boldsymbol{\Psi}^{(\cdot)}_{(\cdot)})\right) . \tag{9.4}$$

This restricted parameter space forces j-wise dependencies to only be through the j-th root term. For example, pairwise interactions are only available through the sufficient statistic $\sqrt[2]{x_s x_t}$ and ternary interactions are only available through the sufficient statistic $\sqrt[3]{x_s x_t x_r}$. Without this restriction interactions would be allowed through multiple terms, e.g. pairwise interactions would be allowed through multiple sufficient statistics $\sqrt[2]{x_s x_t}, \sqrt[3]{x_s x_t}, \cdots, \sqrt[k]{x_s x_t}$.

Thus, this simplified model is more interpretable and easier to learn while still retaining the strongest j-wise interaction terms. For our experiments, we assume this simplified model unless specified otherwise.

9.4.2.1 Node Conditionals

The node conditionals are as follows (see appendix for full derivation):

$$\mathbb{P}(x_s \,|\, \boldsymbol{x}_{-s}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) \propto \exp\left(\sum_{j=1}^{k} \eta_{js} x_s^{1/j} + B(x_s)\right), \tag{9.5}$$

where $\boldsymbol{x}_{-s}$ is all other variables except x_s, $\eta_{js} = \sum_{\ell=1}^{j} \left\langle \left[\Psi^{(\ell)}_{(j)}\right]_s, \sqrt[j]{\ell \boldsymbol{x}} \circ^{\ell-1} \right\rangle$. This is a univariate exponential family with sufficient statistics $x_s^{1/j}$, natural parameters η_{js} and base measure $B(x_s)$. Note that this reduces to the original exponential family if the interaction terms $\eta_{2s} = \cdots = \eta_{ks} = 0$. This node conditional distribution is critical for the parameter estimation that will be described in later sections.

9.4.2.2 Radial Conditionals

As in [Inouye et al., 2016a], we define the *radial* conditional distribution by fixing the unit direction $\mathbf{v} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1}$ of the sufficient statistics but allowing the scaling $z = \|\boldsymbol{x}\|_1$ to be unknown. Thus, we get the following *radial* conditional distribution (see appendix for derivation):

$$\mathbb{P}(\boldsymbol{x} = z\mathbf{v} \,|\, \mathbf{v}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) \propto \exp\left(\sum_{r \in \mathcal{R}} \eta_r(\mathbf{v}) z^r + \tilde{B}_{\mathbf{v}}(z)\right), \tag{9.6}$$

where $\mathcal{R} = \{\ell/j : j \in \{1, \cdots, k\}, \ell \leq j\}$ is the set of possible ratios, $\eta_r(\mathbf{v}) = \sum_{\{(\ell,j):\ell/j=r\}} \langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{\mathbf{v}} \circ^{\ell} \rangle$ are the exponential family parameters, z^r are the corresponding sufficient statistics and $\tilde{B}_{\mathbf{v}}(z) = \sum_s B(z v_s)$ is the base measure. Thus, the radial conditional distribution is a univariate exponential family (as in [Inouye et al.,

2016a]). The radial conditional distributions are critical for showing the normalization of GRM models.

9.4.2.3 Normalization

The previous exponential and Poisson graphical models [Besag, 1974, Yang et al., 2015] could only model negative dependencies. However, we generalize the results from the pairwise SQR model in [Inouye et al., 2016a] and show that GRM normalization for any k puts little to no restriction on the value of the parameters—thus allowing both positive and negative dependencies. For our derivations, let $\mathcal{V} = \{\mathbf{v} : \|\mathbf{v}\|_1 = 1, \mathbf{v} \in \mathbb{R}^p_+\}$ be the set of unit vectors in the positive orthant. The GRM log partition function $A(\mathbf{\Psi}^{(\cdot)}_{(\cdot)})$ can be decomposed into nested integrals over the unit direction and over the scaling z:

$$\begin{aligned} A(\boldsymbol{\theta}, \Phi) &= \log \int_{\mathcal{V}} \int_{\mathcal{Z}(\mathbf{v})} \exp\left(\sum_{j=1}^{k} \sum_{\ell=1}^{j} \left\langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{z\mathbf{v}} \circ^{\ell} \right\rangle + \textstyle\sum_s B(z v_s) \right) \mathrm{d}\mu(z)\, \mathrm{d}\mathbf{v} \\ &= \log \int_{\mathcal{V}} \int_{\mathcal{Z}(\mathbf{v})} \exp\left(\sum_{r \in \mathcal{R}} \eta_r(\mathbf{v}) z^r + \tilde{B}_{\mathbf{v}}(z) \right) \mathrm{d}\mu(z)\, \mathrm{d}\mathbf{v} \end{aligned} \tag{9.7}$$

where $\mathcal{Z}(\mathbf{v}) = \{z \in \mathbb{R}_+ : z\mathbf{v} \in \mathcal{D}\}$, and μ and $\mathcal{D}$ are the measure and domain (or support) of the random variable. Because $\mathcal{V}$ is bounded, the joint distribution will be normalizable if the radial conditional distribution is normalizable—generalizing the results from [Inouye et al., 2016a] for $k > 2$. Informally, the radial conditional distribution converges if the asymptotically largest term of $\{\eta_r(\mathbf{v}) z^r\} \cup \{B(z v_s)\}$ is monotonically decreasing at least linearly.[3] We give several examples in the following paragraphs.

[3] For more formal proofs, we refer the reader to [Inouye et al., 2016a].

9.4.3 Gaussian GRM

For the Gaussian GRM, we take the Gaussian univariate distribution with sufficient statistic $T(x) = x^2$ and $B(x) = 0$. When $k = 2$ (i.e. the standard multivariate Gaussian), the largest radial conditional term is $\eta_1 x^2$ where $\eta_1 = \langle \Psi^{(1)}(1), \mathbf{v}^2 \rangle + \langle \Psi^{(2)}(2), \mathbf{v} \circ \mathbf{v} \rangle$. Note that the radial conditional (i.e. a univariate Gaussian) is normalizable only if $\eta_1 < 0$ for all $\mathbf{v} \in \mathcal{V}$, which is equivalent to the positive definite condition on the Gaussian inverse covariance matrix. We can also consider a Gaussian-like model with $k = 3$. In this case, we have that $\eta_1 = \langle \Psi^{(1)}_{(1)}, \mathbf{v}^2 \rangle + \langle \Psi^{(2)}_{(2)}, \mathbf{v} \circ \mathbf{v} \rangle + \langle \Psi^{(2)}_{(2)}, \mathbf{v}^{\frac{2}{3}} \circ \mathbf{v}^{\frac{2}{3}} \circ \mathbf{v}^{\frac{2}{3}} \rangle$ and we need $\eta_1 < 0 \; \forall \mathbf{v} \in \mathcal{V}$. Note that the Gaussian GRM models for $k > 2$ are novel models to the authors' best knowledge.

9.4.4 Exponential GRM

Because the exponential distribution also has a constant base measure like the Gaussian, the asymptotically largest term is $\eta_1 x$ and thus we must have that $\eta_1 < 0 \; \forall \mathbf{v} \in \mathcal{V}$. However, unlike the Gaussian, in the case of the exponential distribution $\mathcal{V}$ is only positive ℓ_1-normalized vectors. This is a significantly weaker condition on the parameters than for a Gaussian and allows strong positive and negative dependencies.

9.4.5 Poisson GRM

For the Poisson distribution, the base measure is the asymptotically largest term $O(-z\log(z))$. Thus, as in [Inouye et al., 2016a], the parameters can be arbitrarily positive or negative because eventually the base measure will ensure normalizability. Note that this is true for arbitrarily large k.

9.5 Parameter Estimation

As in [Yang et al., 2015, Inouye et al., 2015, 2016a], we solve a set of independent ℓ_1-regularized node-wise regressions for each node—based on the node conditional distributions in Sec. 9.4.2.1—using a Newton-like method for convex optimization with an non-smooth ℓ_1 penalty as in [Hsieh et al., 2014, Inouye et al., 2014a, 2015]. More specifically we take the log likelihood of the node conditionals and add an ℓ_1 penalty on all interaction terms:

$$\underset{\boldsymbol{\Psi}_{(\cdot)}^{(\cdot)}}{\arg\min} -\sum_{s=1}^{p}\left(\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{k}\eta_{jsi}x_{si}^{1/j} - A(\boldsymbol{\eta}_{si})\right)\right) + \lambda\sum_{j==2}^{k}\sum_{\ell=1}^{j}\|\Psi_{(j)}^{(\ell)}\|_1\,, \tag{9.8}$$

where $\eta_{jsi} = \sum_{\ell=1}^{j}\left\langle\left[\Psi_{(j)}^{(\ell)}\right]_s, \ell\sqrt[j]{\boldsymbol{x}_i}\circ^{\ell-1}\right\rangle$ and $\|\cdot\|_1$ is an entry-wise sum of absolute values. Note that this is trivially decomposable into p subproblems and can thus be trivially parallelized to improve computation speed. We use the Newton-like method as in [Hsieh et al., 2011, Inouye et al., 2015] to greatly reduce computation. The initial innovation from [Hsieh et al., 2011] was that the Hessian only needed to be computed over a *free* set of variables each Newton iteration because of the ℓ_1 regularization which suggested sparsity of the parameters. Yet, the number of Newton iterations was very small compared to gradient descent. In the case of GRM models, whose bottleneck is the computation of the gradient of A at least under our current implementation, this Newton-like method provides even more benefit because the gradient only has to be computed a small number of times (roughly 30) in our case rather than the several thousand times that would be needed for running thousands of proximal gradient descent steps for the same level of convergence.

In the next section, we derive the gradient and Hessian for the smooth part of the optimization in terms of the gradient and Hessian of the node conditional log partition function $A(\boldsymbol{\eta})$. Then, we develop a general method for bounding the log partition function $A(\boldsymbol{\eta})$ and associated derivatives even though usually no closed-form exists.

9.5.0.1 Gradient and Hessian of GRMs

9.5.1 Notation for gradient and Hessian

Let $\text{vec}(\Psi^{(\ell)}) \in \mathbb{R}^{p^\ell}$ be the vectorized form of a tensor. For example, the vectorized form of a $p \times p$ matrix is formed by stacking the matrix columns on top of each other to form one long p^2 vector. Also, let $[x \,|\, x \in \mathcal{X}]$ be analogous to the normal set notation $\{x : x \in \mathcal{X}\}$ except that the bracket and vertical line notation creates a vector from all the elements concatenated to together. This is similar to a list comprehension in Python. For our gradient and Hessian calculations, we define the following variable transformations and give them as examples of this notation:

$$
\begin{aligned}
\mathcal{B}_s &= \left\{ \left[\text{vec}\left(\left[\Psi^{(\ell)}_{(j)} \right]_s \right) \,\middle|\, \ell \le j \right] : j \in \{1, 2, \cdots, k\} \right\} \\
&= \Bigg\{ \underbrace{\left[\text{vec}\left(\left[\Psi^{(1)}_{(1)} \right]_s \right) \right]}_{\boldsymbol{\beta}_{1s}}, \underbrace{\left[\text{vec}\left(\left[\Psi^{(1)}_{(2)} \right]_s \right), \text{vec}\left(\left[\Psi^{(2)}_{(2)} \right]_s \right) \right]}_{\boldsymbol{\beta}_{2s}}, \underbrace{\left[\text{vec}\left(\left[\Psi^{(1)}_{(3)} \right]_s \right), \cdots \right]}_{\boldsymbol{\beta}_{3s}}, \\
&\qquad \cdots, \underbrace{\left[\cdots, \text{vec}\left(\left[\Psi^{(k)}_{(k)} \right]_s \right) \right]}_{\boldsymbol{\beta}_{ks}} \Bigg\}, \\
\mathcal{Z}_{si} &= \left\{ \left[\text{vec}\left(\ell \sqrt[j]{\boldsymbol{x}_{si}} \circ^{\ell-1} \right) \,\middle|\, \ell \le j \right] : j \in \{1, 2, \cdots, k\} \right\} \\
&= \Bigg\{ \underbrace{\left[\text{vec}\left(\sqrt[1]{\boldsymbol{x}_i} \circ^0 \right) \right]}_{\boldsymbol{z}_{1s}}, \underbrace{\left[\text{vec}\left(\sqrt[2]{\boldsymbol{x}_i} \circ^0 \right), \text{vec}\left(2\sqrt[2]{\boldsymbol{x}_i} \circ^1 \right) \right]}_{\boldsymbol{z}_{2s}}, \underbrace{\left[\text{vec}\left(\sqrt[3]{\boldsymbol{x}_i} \circ^0 \right), \cdots \right]}_{\boldsymbol{z}_{3s}}, \\
&\qquad \cdots, \underbrace{\left[\cdots, \text{vec}\left(k \sqrt[k]{\boldsymbol{x}_i} \circ^{k-1} \right) \right]}_{\boldsymbol{z}_{ks}} \Bigg\}.
\end{aligned}
$$

With this notation, we have that $\eta_{jsi} = \beta_{js}^T \boldsymbol{z}_{jsi}$. Because each node regression is independent, we focus on solving one of the p subproblems for a particular s using the notation from above:

$$
\arg\min_{\mathcal{B}_s} \sum_{i=1}^{n} f_s(\mathcal{B}_s \,|\, x_{si}, \mathcal{Z}_{si}), \tag{9.9}
$$

where $f_s(\mathcal{B}_s \,|\, x_{si}, \mathcal{Z}_{si}) = -\sum_{j=1}^{k} (\beta_{js}^T \boldsymbol{z}_{jsi}) \sqrt[j]{x_{si}} + A([\beta_{js}^T \boldsymbol{z}_{jsi} \,|\, j \in \{1, \cdots, k\})$. For notational simplicity, we suppress the dependence on s and i in the derivations of the gradient and

Hessian of $f(\cdot)$ (the gradient and Hessian are merely the sum over all instances). The gradient and Hessian are as follows:

$$\nabla f(\mathcal{B} \,|\, x, \mathcal{Z}) = \left[\left(-\sqrt[j]{x} + \frac{\partial A}{\partial \eta_j} \right) \boldsymbol{z}_j \,\middle|\, j \in \{1, 2, \cdots, k\} \right], \tag{9.10}$$

$$\nabla^2 f(\mathcal{B} \,|\, x, \mathcal{Z}) = \begin{bmatrix} \left[\frac{\partial^2 A}{\partial \eta_1 \partial \eta_j} \boldsymbol{z}_j \circ \boldsymbol{z}_1 \,\middle|\, j \in \{1, 2, \cdots, k\} \right], \\ \left[\frac{\partial^2 A}{\partial \eta_2 \partial \eta_j} \boldsymbol{z}_j \circ \boldsymbol{z}_2 \,\middle|\, j \in \{1, 2, \cdots, k\} \right], \\ \vdots \\ \left[\frac{\partial^2 A}{\partial \eta_k \partial \eta_j} \boldsymbol{z}_j \circ \boldsymbol{z}_k \,\middle|\, j \in \{1, 2, \cdots, k\} \right] \end{bmatrix}. \tag{9.11}$$

Note how the gradient and Hessian are simple functions of $\boldsymbol{z}_j$ and the derivatives of $A(\boldsymbol{\eta})$. We develop bounded approximations for the $A(\boldsymbol{\eta})$ next.

9.5.1.1 Gradient and Hessian of $A(\boldsymbol{\eta})$

Because the node conditional distributions are not standard distributions, we must either derive the closed-form log partition function as done with the SQR exponential model in [Inouye et al., 2016a], or we must approximate the log partition function and its first and second derivatives. To the authors' best knowledge, even for the simplified SQR model with $k = 2$, no closed-form solution to log partition function exists for SQR node conditionals except for the discrete, Gaussian and exponential SQR models. Thus, we seek a general way to estimate the log partition function and associated derivatives.

9.5.2 Reformulated as Expectations

We first note that the gradient and Hessian of $A(\boldsymbol{\eta})$ are merely functions of particular expectations—a well-known result of exponential families:

$$A(\boldsymbol{\eta}) = \log \int_{\mathcal{D}} \exp\Big(\sum_{j=1}^{k} \eta_j x^{\frac{1}{j}} + B(x)\Big) \mathrm{d}\mu(x) \tag{9.12}$$

$$\nabla A(\boldsymbol{\eta}) = [\mathbb{E}(x^{\frac{1}{j}}) \,|\, j \in \{1, \cdots, k\}] \tag{9.13}$$

$$\nabla^2 A(\boldsymbol{\eta}) = \begin{bmatrix} \Big[\mathbb{E}(x^{\frac{1}{j}+\frac{1}{2}}) - \mathbb{E}(x^{\frac{1}{j}})\mathbb{E}(x) \,|\, j \in \{1, \cdots, k\}\Big] \\ \Big[\mathbb{E}(x^{\frac{1}{j}+\frac{1}{2}}) - \mathbb{E}(x^{\frac{1}{j}})\mathbb{E}(x^{\frac{1}{2}}) \,|\, j \in \{1, \cdots, k\}\Big] \\ \vdots \\ \Big[\mathbb{E}(x^{\frac{1}{j}+\frac{1}{k}}) - \mathbb{E}(x^{\frac{1}{j}})\mathbb{E}(x^{\frac{1}{k}}) \,|\, j \in \{1, \cdots, k\}\Big] \end{bmatrix}. \tag{9.14}$$

Thus, we need to compute expectations for at most $\binom{k}{2} + k$ functions of the form $\mathbb{E}(x^a)$. To develop our approximations under a unified framework, let us define the following function $M(a)$ and its subfunctions denoted $f(x)$ and $g(x)$:

$$M(a) = \log \int_{\mathcal{D}} x^a \exp\Big(\sum_{j=1}^{k} \eta_j x^{\frac{1}{j}} + B(x)\Big) \mathrm{d}\mu(x) \tag{9.15}$$

$$= \log \int_{\mathcal{D}} \exp\Big(\underbrace{\eta_1 x + B(x)}_{f(x)} + \underbrace{\textstyle\sum_{j=2}^{k} \eta_j x^{\frac{1}{j}} + \log(x^a)}_{g(x)}\Big) \mathrm{d}\mu(x)\,. \tag{9.16}$$

By simple inspection, we see that $M(0) = A(\eta_1, \eta_2)$ and $\mathbb{E}(x^a) = \exp\big(M(a) - M(1)\big)$. Thus, by approximating $M(a)$, we can approximate all the necessary derivatives. If $g(x) = 0$, then this is simply the log partition function of the base exponential family, which is usually known in closed form. If $g(x)$ is upper and lower bounded by a linear functions, i.e. $b_l x + c_l = g_l(x) \leq g(x) \leq g_u(x) = b_u x + c_u$, then we can form a modified functions of $f(x)$ that will be upper and lower bounds of $f(x) + g(x)$:

$$(\eta_1 + b_l)x + c_l = \hat{f}_l(x) \leq f(x) + g(x) \leq \hat{f}_u(x) = (\eta_1 + b_u)x + c_u\ . \tag{9.17}$$

Assuming $\hat{\eta}_l = \eta_1 + b_l$ and $\hat{\eta}_u = \eta_1 + b_u$ are valid parameters, we can then use the original exponential family CDF—which is usually known in closed form—to compute the needed integrals. If we know the concavity of each region, we can form linear upper and lower bounds using the theory of convexity. The secant line and the first-order Taylor series approximation form upper and lower bounds or vice versa depending on concavity. We can bound the tails of $g(x)$ with a constant function or Taylor series approximation as appropriate. See appendix for details on linear approximations for $g(x)$.

9.5.3 Approach to Bounding $M(a)$

Our approach splits the integral into $d = O(1)$ integrals which bound the integral over different subdomains of the domain. As will be seen later, we will then use the CDF function of the base exponential family to approximate the integrals over these subdomains (see appendix for full derivation):

$$M(a) \approx \log \sum_{i=1}^{d} \int_{\mathcal{D}_i} \exp(\hat{f}_i(x)) \mathrm{d}\mu(x) \tag{9.18}$$

$$= \log \sum_{i=1}^{d} \exp\left(c_i + A(\hat{\eta}_i) + \log\big(\mathrm{CDF}\big(\max(\mathcal{D}_i) \,|\, \hat{\eta}_i\big) - \mathrm{CDF}\big(\min(\mathcal{D}_i) \,|\, \hat{\eta}_i\big)\right), \tag{9.19}$$

where the domain is split into disjoint subdomains, i.e. $\{\mathcal{D}_i : \mathcal{D} = \bigcup_i^d \mathcal{D}_i, \forall i \neq j, \mathcal{D}_i \cap \mathcal{D}_j = \emptyset\}$, $(\hat{\eta}, b)$ are either $(\hat{\eta}_u, b_u)$ or $(\hat{\eta}_l, b_l)$ depending on whether the upper or lower bound is needed, $A(\hat{\eta})$ and $\mathrm{CDF}(\cdot)$ are the log partition function and CDF of the original exponential family. Note that assuming $A(\hat{\eta})$ and $\mathrm{CDF}(\cdot)$ are available in closed form—as is the case for the Poisson distribution—this approximation can be computed in $O(d) = O(1)$ time.

9.5.4 Algorithm to Find Appropriate Subdomains $\mathcal{D}_i$

We need that every subdomain has a constant concavity (i.e. either concave or convex over the subdomain) in order to use Taylor series and secant line bounds (and a constant

bound for the tails). Thus, we use the following algorithm to find subdomains that help minimize the bounds:

1. Find all real roots of $g''(x)$, denoted $(x_1'', x_2'', \cdots)$ so we know the inflection points.
2. Use inflection points and endpoints of domain (e.g. 0 and ∞ for Poisson) to define the initial subdomains.
3. Compute initial bounds for these subdomains using Eqn. 9.19.
4. Split the subdomain with the largest difference between upper and lower bounds (i.e. the subdomain with the largest error).
5. Recompute bounds for the two new subdomains formed by splitting the largest error subdomain.
6. Repeat previous two steps until d domains have been obtained.

Note that the roots of $g''(x)$ can be solved by expanding to a polynomial and then computing the eigenvalues of the companion matrix. An illustration of the method can be seen in Fig. 9.1.

9.6 Results on Text Documents

We computed the Poisson GRM model on two datasets: Classic3 and Grolier encyclopedia articles. The Classic3 dataset contains 3893 research abstracts from library and information sciences, medical science and aeronautical engineering (more information can be found in Chapter 5 Table 5.1). The Grolier encyclopedia dataset contains 5000 random articles from the Grolier encyclopedia (this is the same Grolier dataset used in Chapter 7). We set $k = 3$, $p = 500$ and $\lambda = 0.01$ for our experiments. We chose 10 interval endpoints (i.e. 9 subdomains) for our approximations. Note that this means there are at least $\binom{500}{3} \approx 2 \times 10^7$ possible parameters. We give the top 10 positive parameters for individual, edge-wise and triple-wise combinations. These results illustrate that our model and algorithm can find interesting pairwise and triple-wise words. Top 50 of both negative

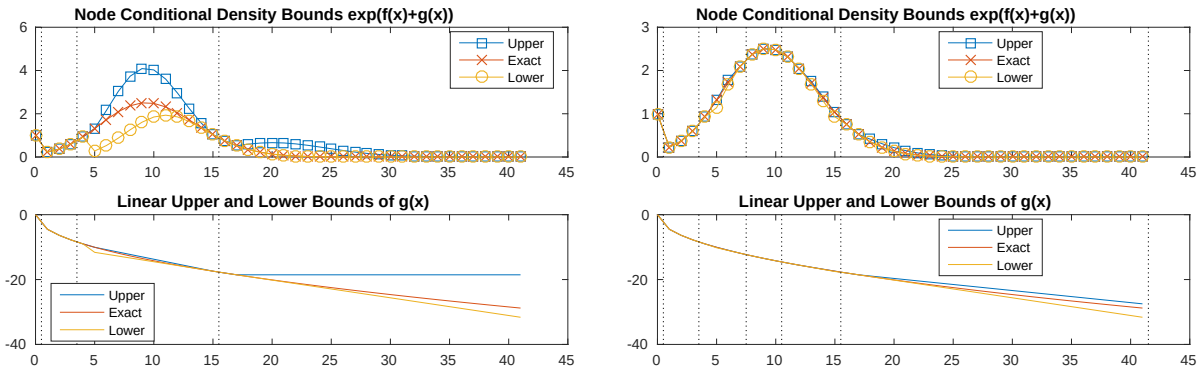


Figure 9.1: Approximation of the $M(a)$ function with $a = 0$ and $\boldsymbol{\eta} = [3.0232, -4.4966]$ for 2 subdomains (left) and for 5 subdomains (right) using the algorithm described in Sec. 9.5.1.1. The top is the actual values of the summation in Eqn. 9.16 and the bottom is the linear approximation $bx + c$ to the non-linear part $g(x)$ as in Eqn. 9.17.

Table 9.1: Table of Tuples

Classic3 Dataset

Single	Pairwise	Triple-wise
information	boundary + layer	layer + skin + friction
flow	heat + transfer	information + retrieval + storage
library	tunnel + wind	pressure + number + mach
pressure	edge + leading	layer + plate + flat
system	bone + marrow	flow + given + case
theory	angle + attack	flow + plate + flat
results	skin + friction	number + mach + investigation
data	growth + hormone	number + mach + conducted
patients	plate + flat	wing + ratio + aspect
found	shock + wave	number + based + reynolds
method	mach + numbers	pressure + ratio + jet
cells	number + mach	heat + transfer + coefficients
analysis	number + reynolds	system + retrieval + user

Grolier Encyclopedia Dataset

Single	Pairwise	Triple-wise
american	km + mi	american + city + york
century	language + languages	city + population + center
john	china + chinese	population + deg + mm
called	plants + plant	major + population + persons
city	deg + temperatures	ft + sea + level
world	music + musical	american + south + america
life	spanish + spain	deg + sq + consists
united	novel + novels	city + deg + july
system	art + painting	war + civil + union
university	poetry + poet	population + deg + elected
family	agricultural + agriculture	american + united + english
time	war + civil	war + congress + program
war	literature + literary	population + sq + persons

and positive dependencies for single, pairwise and triple-wise dependencies can be found in the appendix. The timing for these experiments using prototype code in MATLAB on TACC Maverick cluster (`https://portal.tacc.utexas.edu/user-guides/maverick`) was 2653 seconds for the Classic3 dataset and 5975 seconds for the Grolier dataset. Given the extremely large number of parameters to be optimized, this gives evidence that GRM models are computationally tractable while still wanting for some improvement.

9.7 Discussion

While it may seem at first that this model is impractical for even $k = 4$, we suggest some ideas for reducing the parameter space. First, if some parameters are known or expected a priori to be non-zero, we could only allow those parameters to be non-zero. For example, known genetic pathways could be encoded as k-wise cliques. Thousands of known pathways could be added which would only incur thousands of parameters, which is very small relative to all possible parameters. Second, the optimization could proceed in a stage-wise fashion such that the first a model is fit for $k = 1$, then this model is used to choose which parameters to allow in the next model of $k = 2$, etc. For example, we could first train a model with only pairwise parameters ($k = 2$). Then, we could find all triangles in the discovered graph and only add these parameters for training a model with $k = 3$. This heuristic would significantly reduce the number of possible parameters if the parameters are assumed to be sparse (as is usually the case with ℓ_1-regularized objectives). Third, the tensors could be constrained to be low-rank and thus only $O(p)$ values for each tensor would be needed. For example, we could assume that the pairwise tensors are low-rank matrices. For higher order tensors, a similar idea could hold, e.g. $\Psi^{(\ell)}_{(j)} = \sum_{i=1}^{M} \boldsymbol{\theta}_i \circ^{\ell}$, where M is $O(1)$.

9.8 Conclusion

We generalize the previous SQR [Inouye et al., 2016a] model to include factors of size $k > 2$. We study this general distribution by giving the node and radial conditional distributions, which provides simple conditions for normalization of the GRM class of models. We then develop an approximation technique for estimating the node-wise log partition function and associated derivatives for the Poisson case—note that [Inouye et al., 2016a] only provided an algorithm for approximating the exponential SQR model. Finally, we qualitatively demonstrated our model on two real world datasets.

Chapter 10

Closed-Form Conditional Marginals via Gaussian-Copula Models

10.1 Abstract

Motivated by the comparison of copula and graphical models in Chapter 5, we investigate a connection between Gaussian-copula models and graphical models in this chapter. More specifically, we seek to understand the form of the conditionals of Gaussian-copula distributions paired with non-Gaussian marginals. Conditional distributions provide a way to estimate missing values given known values. As another example, conditional distributions allow researchers to reason about hypothetical situations which were not seen in the training data. These applications require estimation of the conditional marginal distributions—i.e. the marginal distribution of a variable conditioned on *some but not all* other variables. The multivariate Gaussian provides closed-form solutions to the conditional marginal distributions but many real-world datasets are non-Gaussian including non-negative, count or discrete data. Therefore, we propose the use of a Gaussian copula paired with non-Gaussian marginal distributions. As a key concept, we show that the conditional distribution of a Gaussian copula model can be reduced to another Gaussian copula paired with closed-form marginals and dependency structure. This reduction allows for direct access to the mean and median of the conditional marginal distributions. We also develop two approximations for discrete conditional marginal distributions. We evaluate these closed-form conditional marginal distributions on several real-world datasets in terms of missing value imputation.

10.2 Introduction

To the author's best knowledge, the general form of these conditional marginal distributions has not been derived—in particular, showing that a conditional copula model is another copula model has not been known. Thus, the general closed-form solutions to the conditional marginals has not been known. The full conditional (i.e. conditioning on all other variables) for regression or imputation has been derived [Clemen and Reilly, 1999, Parsa and Klugman, 2011, Wang and Hua, 2014] but not a partial conditional—i.e. condition on some variables.

Some methods consider conditional copulas for sampling by sampling from one dimension at a time and building up a complete sample [Cherubini et al., 2004b, Kort, 2007]. Schmitz [2003] provide a good view of conditional distributions connected to the partial derivatives of the copula CDFs. Kaarik [2006] and Käärik and Käärik [2009] show that the conditional distribution is a ratio of copula densities and suggests that this can be used to impute a missing value given previous values—i.e. impute one value based on all the history before it. Note that this is still the full conditional, i.e. conditioning on all the history rather than only parts of the history. [Roth, 2013] gives the form of the marginal and conditional distributions for the multivariate t distribution, which supports the conjecture that the ideas in this chapter also follow for t copulas as we propose in later sections.

We derive that the conditional distributions of a Gaussian-copula model is also a Gaussian-copula model with modified parameters and marginal distributions. The beauty in this derivation is that the conditional marginal distributions are actually known in closed-form, even up to normalization constant. Thus, we can compute the likelihood or any statistic of the marginal distributions such as the median or mean.

10.3 Preliminaries

10.3.1 Notation

Let p and n denote the number of dimensions and number data instances respectively. We will generally use uppercase letters for matrices (e.g. R, W), boldface lowercase letters for vectors (i.e $\boldsymbol{x}, \boldsymbol{y}$) and lowercase letters for scalar values (i.e. x_s, y_s). Subvectors are given by subscripts either on a single value as in x_s or by a set of values $\boldsymbol{x}_B$ where B is a set indices. Submatrices are indexed in a similar manner so that R_{BA} is the submatrix formed by removing all rows not in B and all columns not in A. Also, unless noted otherwise, functions on vectors will be coordinate-wise functions. For example, $\sqrt{\boldsymbol{x}} = [\sqrt{x_1}, \sqrt{x_2}, \cdots, \sqrt{x_p}]$ or $\Phi^{-1}(\mathbf{v}) = [\Phi^{-1}(u_1), \Phi^{-1}(u_2), \cdots, \Phi^{-1}(u_p)]$. In the case where the coordinate-wise function has different functions for different variables—i.e. $F_s(x_s)\ \forall s$ —, we will denote $F(\boldsymbol{x}) = [F_1(x_1), F_2(x_2), \cdots, F_p(x_p)]$. For subvectors of these functions, we define $F_B(\boldsymbol{x}_B) = (F(\boldsymbol{x}))_B$. Finally, we will use the function $\text{diag}(\cdot)$ to either extract the diagonal of an input matrix or make a diagonal matrix from an input vector; for example, $\text{diag}(W) = [w_{11}, w_{22}, \cdots, w_{pp}]$ and $(\text{diag}(\boldsymbol{x}))_{st} = \{x_s \text{ if } s = t \text{ and } 0 \text{ otherwise}\}$.

10.3.2 Copulas

While copula-based models often refer to the multivariate CDF of the copula distribution (see Sec. 5.3.2 for a definition using multivariate CDFs), we will be working primarily with the multivariate copula density (i.e. PDF) denoted $c(\cdot)$. Given the marginal CDFs $F_s(\cdot)$ and corresponding marginal PDFs $f_s(\cdot)$, any continuous joint distribution by Sklar's Theorem can be represented as:

$$\mathbb{P}(\boldsymbol{x}) = c(F(\boldsymbol{x})) \prod_{s=1}^{p} f_s(x_s), \tag{10.1}$$

where $F(\boldsymbol{x}) = [F_1(x_1), F_2(x_2), \cdots, F_p(x_p)]$. Note that each univariate marginal $F_s(\cdot)$ can be arbitrarily different—including non-parametric univariate distributions. The Gaussian

copula with correlation matrix R can be derived from the multivariate Gaussian distribution:

$$c^{\text{Gauss}}(\mathbf{v} \mid R) = \frac{\mathbb{P}_{\text{Gauss}}(\Phi^{-1}(\mathbf{v}))}{\prod_{s=1}^{p} \mathbb{P}_{\text{Gauss}}(\Phi^{-1}(u_s))} \tag{10.2}$$
$$= |R|^{-1/2} \exp\big(-\frac{1}{2}\Phi^{-1}(\mathbf{v})^T(R^{-1} - \boldsymbol{I})\Phi^{-1}(\mathbf{v})\big),$$

where $\Phi^{-1}(\cdot)$ is the inverse CDF of the normal distribution applied coordinate-wise, R is a correlation matrix, and $\boldsymbol{I}$ is the identity matrix. Essentially, the Gaussian copula is the ratio of a multivariate normal with covariance R and a standard independent multivariate normal. We will show that the particular form of the Gaussian copula enables closed-form solutions to the conditional marginal distributions. We refer the reader to Sec. 5.3.2 for more details on copula models.

10.4 Conditional Copula Distribution

Now, let us consider conditioning on some set of variables $B \subset \{1, 2, \cdots, p\}$. Let us denote the complement of B as A and let $F_A(\boldsymbol{x}_A)$ be the vector formed by concatenating $F_t(x_t)\, \forall t \in A$. Also, let $\Phi(\cdot)$, $\Phi^{-1}(\cdot)$, and $\phi(\cdot)$ denote the CDF, inverse CDF and PDF of the standard normal distribution.

Theorem 1. *Given a joint Gaussian copula model with correlation matrix R, marginal CDFs $F_s(\cdot)$ and marginal PDFs $f_s(\cdot)$, the conditional distribution given a subset of variables $B \in \{1, 2, \cdots, p\}$ is another Gaussian copula model:*

$$\mathbb{P}(\boldsymbol{x}_A \mid \boldsymbol{x}_B, R, F, f) = c^{Gauss}(G(\boldsymbol{x}_A) \mid \widetilde{R}) \prod_{t \in A} g_t(x_t), \tag{10.3}$$

where the conditional marginal distributions—i.e. $\mathbb{P}(x_t \mid \boldsymbol{x}_B)$ are given by the following CDFs

and PDFs:

$$G_t(x_t) = \Phi\left(\frac{\Phi^{-1}(F_t(x_t)) - \mu_t}{\sigma_t}\right) \tag{10.4}$$

$$g_t(x_t) = \frac{\mathrm{d}G_t(x)}{\mathrm{d}x} = \frac{\phi\left(\frac{\Phi^{-1}(F_t(x_t))-\mu_t}{\sigma_t}\right)}{\sigma_t\,\phi(\Phi^{-1}(F_t(x_t)))}\,f_t(x_t)\,, \tag{10.5}$$

and where $\boldsymbol{\mu} = R_{AB}R_{BB}^{-1}\Phi^{-1}(F_B(\boldsymbol{x}_B))$, $\Sigma = R_{AA} - R_{AB}R_{BB}^{-1}R_{BA}$, $\sigma = \sqrt{\mathrm{diag}(\Sigma)}$, *and* $\widetilde{R} = \mathrm{diag}(\sigma)^{-1}\Sigma\mathrm{diag}(\sigma)^{-1}$.

The proof while straightforward is relatively tedious and thus is given in Sec. 10.7. We now give several useful corollaries below regarding the CDF, PDF, and inverse CDF.

Corollary 1. *The CDF and PDF of the conditional marginal distributions* $\mathbb{P}(x_t \,|\, \boldsymbol{x}_B)$ *are given in closed-form by* $G_t(x_t)$ *and* $g_t(x_t)$ *respectively.*

For the following corollary, let $F_t^{-1}(u) = \inf\{x \in \mathbb{R} : F_t(x_t) \geq u\}$ denote the generalized inverse CDF of $F_t(x_t)$.

Corollary 2. *The inverse CDF, or quantile function, of the conditional marginal distributions* $\mathbb{P}(x_t \,|\, \boldsymbol{x}_B)$ *is given in closed-form as:*

$$G_t^{-1}(u) = F_t^{-1}\big(\Phi(\sigma_t\Phi^{-1}(u) + \mu_t)\big)\,. \tag{10.6}$$

Corollary 2 means that the median is known in closed-form and direct sampling via inverse transform sampling is trivial. Another benefit of having the inverse CDF in closed form is that the mean of the distribution can be found by numerically computing the following simple univariate *definite* integral using adaptive quadrature or similar numerical integration:

$$\mathbb{E}[x_t] = \int_0^1 G_t^{-1}(u)\mathrm{d}u. \tag{10.7}$$

10.4.1 Conjecture About Multivariate t-copula Models

We give the following conjecture regarding t-copula models that is similar to Theorem 1 for Gaussian-copula models.

Conjecture 1. *Given a joint multivariate t-copula model with correlation matrix R, degree of freedom $\nu > 0$, marginal CDFs $F_s(\cdot)$ and marginal PDFs $f_s(\cdot)$, the conditional distribution given a subset of variables $B \in \{1, 2, \cdots, p\}$ is another multivariate t copula model:*

$$\mathbb{P}(\boldsymbol{x}_A \,|\, \boldsymbol{x}_B, \nu, R, F, f) = c^t(G(\boldsymbol{x}_A) \,|\, \tilde{\nu}, \widetilde{R}) \prod_{t \in A} g_t(x_t), \tag{10.8}$$

where the conditional marginal distributions—i.e. $\mathbb{P}(x_t \,|\, \boldsymbol{x}_B)$ are given by the following CDFs and PDFs:

$$G_t(x_t) = T_{\nu+|B|}\left(\frac{T_\nu^{-1}(F_t(x_t)) - \mu_t}{\sigma_t}\right) \tag{10.9}$$

$$g_t(x_t) = \frac{\mathrm{d}G_t(x)}{\mathrm{d}x} = \frac{t_{\nu+|B|}\left(\frac{T_\nu^{-1}(F_t(x_t)) - \mu_t}{\sigma_t}\right)}{\sigma_t \, t_{\nu+|B|}(T_\nu^{-1}(F_t(x_t)))} \, f_t(x_t)\,, \tag{10.10}$$

and where

$$\begin{aligned}
T &= \text{CDF of standard } t \text{ distribution} \\
T^{-1} &= \text{Inverse CDF of standard } t \text{ distribution} \\
t &= \text{PDF of standard } t \text{ distribution} \\
\tilde{\nu} &= \nu + |B| \\
\boldsymbol{\mu} &= R_{AB} R_{BB}^{-1} \Phi^{-1}(F_B(\boldsymbol{x}_B)) \\
\Sigma &= \frac{\nu + \boldsymbol{x}_B^T R_{BB}^{-1} \boldsymbol{x}_B}{\nu + |B|}(R_{AA} - R_{AB} R_{BB}^{-1} R_{BA}) \\
\sigma &= \sqrt{\mathrm{diag}(\Sigma)} \\
\widetilde{R} &= \mathrm{diag}(\sigma)^{-1} \Sigma \mathrm{diag}(\sigma)^{-1}\,.
\end{aligned}$$

While we have not proven this yet, we conjecture that this likely follows from the fact that a conditional multivariate t distribution is another multivariate t distribution with modified parameters as above and that the marginals of a multivariate t distribution are univariate t distributions [Roth, 2013]. In addition, we know that the marginal distributions of a multivariate t distribution are closed-form [Roth, 2013]. Thus, the core properties needed are still available.[1]

10.5 Extension to Mixtures

We further extend some of the results to mixtures of Gaussian-copula models. Most of the required quantities are known in closed-form or require a one-dimensional root finding algorithm in the case of the inverse CDF or quantile function.

Corollary 3. *The conditional distribution of a mixture of k Gaussian-copula models formulated as:*

$$\mathbb{P}(\boldsymbol{x} \mid R^{\cdots}, F^{\cdots}) = \sum_{j=1}^{k} \mathbb{P}(z = j \mid \boldsymbol{w}) \mathbb{P}(\boldsymbol{x} \mid z = j, R^{(j)}, F^{(j)})$$

is a reweighted mixture of Gaussian-copula models:

$$\mathbb{P}(\boldsymbol{x}_A \mid \boldsymbol{x}_B) = \sum_{j=1}^{k} \mathbb{P}(z = j \mid \widetilde{\boldsymbol{w}}) \mathbb{P}(\boldsymbol{x}_A \mid z = j, \widetilde{R}^{(j)}, G^{(j)}),$$

where $G^{(j)}(\cdot)$ and $\widetilde{R}^{(j)}$ are defined as in Thm. 1 and $\widetilde{w}_j = \frac{w_j \mathbb{P}(\boldsymbol{x}_B \mid z=j, R^{(j)}, F^{(j)})}{\sum_{m=1}^{k} w_j \mathbb{P}(\boldsymbol{x}_B \mid z=m, R^{(m)}, F^{(m)})}$.

[1] Unlike the multivariate Gaussian, however, the uncorrelated multivariate t distribution —i.e. $\Sigma = \boldsymbol{I}$—is *not* the product of marginal t distributions. However, this property is likely not needed for the derivation.

Proof.

$$
\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_A \,|\, \boldsymbol{x}_B, R^{\cdots}, F^{\cdots}) \\
&= \frac{\mathbb{P}(\boldsymbol{x}_A, \boldsymbol{x}_B, R^{\cdots}, F^{\cdots})}{\mathbb{P}(\boldsymbol{x}_B, R^{\cdots}, F^{\cdots})} \\
&= \frac{\sum_{j=1}^{k} \mathbb{P}(z = j \,|\, \boldsymbol{w}) \mathbb{P}(\boldsymbol{x}_A, \boldsymbol{x}_B \,|\, z = j, R^{(j)}, F^{(j)})}{\sum_{m=1}^{k} \mathbb{P}(z = m \,|\, \boldsymbol{w}) \mathbb{P}(\boldsymbol{x}_B \,|\, z = m, R^{(m)}, F^{(m)})} \\
&= \frac{\sum_{j=1}^{k} w_j \mathbb{P}(\boldsymbol{x}_A, \boldsymbol{x}_B \,|\, z = j, R^{(j)}, F^{(j)})}{\sum_{m=1}^{k} w_m \mathbb{P}(\boldsymbol{x}_B \,|\, z = m, R^{(m)}, F^{(m)})} \\
&= \frac{\sum_{j=1}^{k} w_j \mathbb{P}(\boldsymbol{x}_B \,|\, z = j, R^{(j)}, F^{(j)}) \mathbb{P}(\boldsymbol{x}_A \,|\, \boldsymbol{x}_B, z = j, R^{(j)}, F^{(j)})}{\sum_{m=1}^{k} w_m \mathbb{P}(\boldsymbol{x}_B \,|\, z = m, R^{(m)}, F^{(m)})} \\
&= \sum_{j=1}^{k} \widetilde{w}_j \mathbb{P}(\boldsymbol{x}_A \,|\, \boldsymbol{x}_B, z = j, R^{(j)}, F^{(j)}) \\
&= \sum_{j=1}^{k} \widetilde{w}_j \mathbb{P}(\boldsymbol{x}_A \,|\, z = j, \widetilde{R}^{(j)}, G^{(j)}),
\end{aligned}
$$

where the last step is by Thm. 1. □

The CDF of a mixture $F_{\text{mix}}(\cdot)$ is given by the CDFs of the components $F^{(j)}(\cdot)$:

$$F_{\text{mix}}(x) = \sum_{j=1}^{k} w_j F^{(j)}(x)\,.$$

Because $F_{\text{mix}}(x)$ is a convex combination of $F^{(j)}(x)$, we know that:

$$\min_j F^{(j)}(x) \leq F_{\text{mix}}(x) \leq \max_j F^{(j)}(x)\,.$$

Because of this, we can show bounds for any given $u \in [0, 1]$. Given a particular $u \in [0, 1]$, let $y = F_{\text{mix}}^{-1}(u)$, which we know exists because $F_{\text{mix}}(\cdot)$ is strictly monotonic and continuous. There exists a $j \in \{1, \cdots, k\}$ such that $F^{(j)}(y) \geq F_{\text{mix}}(y)$. Then, because $(F^{(j)})^{-1})(\cdot)$ is

strictly monotonic, we can apply the inverse transformation to both sides of the inequality to yield:

$$\begin{aligned} y &\geq (F^{(j)})^{-1}(F_{\text{mix}}(y)) \\ \Rightarrow F_{\text{mix}}^{-1}(u) &\geq (F^{(j)})^{-1}(u) \\ &\geq \min_j (F^{(j)})^{-1}(u)\,. \end{aligned}$$

This can be proved similarly for the maximum to yield the following inequalities:

$$\min_j (F^{(j)})^{-1}(u) \leq F_{\text{mix}}^{-1}(u) \leq \max_j (F^{(j)})^{-1}(u)\,. \tag{10.11}$$

Thus, the quantile function of a mixture can be upper and lower bounded by the minimum and maximum of the quantile functions of the components. Using this as a starting point, a simple univariate root finding algorithm such as Brent's method can be used to find any quantile of the mixture, including the median of the mixture.

10.6 Discrete Marginals

There are two possible ways for handling discrete marginal distributions such as the Poisson when modeling count data (e.g. word counts or protein counts). First, the discrete data can be augmented to form auxiliary continuous domain variables; this is also known as the continuous extension (CE) [Denuit and Lambert, 2005, Nikoloulopoulos, 2013b]:

$$x^* = x + (u - 0.5), \quad \text{where} \quad u \sim \text{Uniform}.$$

The one difference from [Denuit and Lambert, 2005] is that we only shift by 0.5 instead of 1 so that the median and mean are much closer to their discrete counterparts. This continuous approximation to the discrete marginals can be viewed as evenly spreading the discrete probability over the unit interval centered at the discrete values. Essentially, this

moves the count-valued marginal CDF to a piecewise-linear continuous CDF defined over the domain $[-0.5, \infty)$.

A second way is to leave the marginals discrete and handle the discrete marginals carefully. One difference is that the density function $g_t(x_t)$ from Theorem 1 can no longer be defined in terms of the derivative of $G_t(x_t)$ but we can define it as the discrete difference: $g_t(x_t) = G_t(\lfloor x_t \rfloor) - G_t(\lceil x_t - 1 \rceil)$. Also, when conditioning on a particular value of x, we modify $F(x)$ to be $\tilde{F}(x) = 0.5(F(x) + F(x-1))$ so that the conditioning value of the copula is the median value that could have generated x. From a generative point of view, any $u \in [F(x-1), F(x))$ could have generated x, and thus, we approximate the u by the median between these two values.

10.7 Proof of Theorem 1

Proof. For notational reasons, let us define $W = R^{-1}$. The block matrices of W and R are related as follows based on well-known block inversion identities:

$$W_{BB} = (R_{BB} - R_{BA}R_{AA}^{-1}R_{AB})^{-1} \tag{10.12}$$

$$= R_{BB}^{-1} + R_{BB}^{-1}R_{BA}(R_{AA} - R_{AB}R_{BB}^{-1}R_{BA})^{-1}R_{AB}R_{BB}^{-1}$$

$$W_{BA} = -R_{BB}^{-1}R_{BA}(R_{AA} - R_{AB}R_{BB}^{-1}R_{BA})^{-1} \tag{10.13}$$

$$W_{AB} = W_{BA}^{T} \tag{10.14}$$

$$= -(R_{AA} - R_{AB}R_{BB}^{-1}R_{BA})^{-1}R_{AB}R_{BB}^{-1}$$

$$W_{AA} = (R_{AA} - R_{AB}R_{BB}^{-1}R_{BA})^{-1}. \tag{10.15}$$

These can be similarly defined for R_{BB},R_{BA},R_{AB} and R_{AA} in terms of W_{BB},W_{BA},W_{AB} and W_{AA} by switching R and W in all the equations.

We will prove by induction on the size of the conditioning set $|B|$. We will first introduce a copula decomposition lemma and then proceed with the induction. The lemma

proof will be in a later section.

Lemma 1. *A Gaussian copula in $\mathbb{R}^p$ can be decomposed into a smaller Gaussian copula in $\mathbb{R}^{p-1}$ multiplied by a product of $p-1$ independent factors:*

$$\begin{aligned} & c^{Gauss}(F(\boldsymbol{x}) \,|\, R) \\ & \quad = c^{Gauss}(\Phi(\mathrm{diag}(\boldsymbol{\sigma})^{-1}\Phi^{-1}(F_A(\boldsymbol{x}_A)) - \boldsymbol{\mu})) \,|\, \widetilde{R}) \\ & \qquad \times \prod_{s=1}^{p-1} \frac{\phi(\Phi(\frac{\Phi^{-1}(F_s(x_s))-\mu_s}{\sigma_s}))}{\sigma_s \phi(\Phi^{-1}(F_s(x_s)))}, \end{aligned}$$

where $A = \{1, 2, \cdots, p-1\}$, $B = \{p\}$, and Σ, $\boldsymbol{\sigma}$, $\boldsymbol{\mu}$ and $\widetilde{R}$ are defined as in Theorem 1.

10.7.1 Base Case $|B| = 1$

We can assume that we are conditioning on the last variable x_p w.l.o.g. Thus, $B = \{p\}$ and $A = \{1, 2, \cdots, p-1\}$. First, by simple conditional probability, we know that:

$$\mathbb{P}(\boldsymbol{x}_A \,|\, x_p, R, F, f) = \frac{\mathbb{P}(\boldsymbol{x})}{\mathbb{P}(x_p)} \tag{10.16}$$

$$= \frac{c^{\mathrm{Gauss}}(F(\boldsymbol{x}) \,|\, R) \prod_{s=1}^{p} f_s(x_s)}{f_s(x_s)} \tag{10.17}$$

$$= c^{\mathrm{Gauss}}(F(\boldsymbol{x}) \,|\, R) \prod_{s=1}^{p-1} f_s(x_s) \tag{10.18}$$

Using Lemma 1, we arrive at the result after some simplification:

$$\begin{aligned} & \mathbb{P}(\boldsymbol{x}_A \,|\, x_p, R, F, f) = c^{\mathrm{Gauss}}(F(\boldsymbol{x}) \,|\, R) \prod_{s=1}^{p-1} f_s(x_s) \\ & = c^{\mathrm{Gauss}}(\Phi(\mathrm{diag}(\boldsymbol{\sigma})^{-1}\Phi^{-1}(F_A(\boldsymbol{x}_A)) - \boldsymbol{\mu})) \,|\, \widetilde{R}) \\ & \times \prod_{s=1}^{p-1} \frac{\phi(\Phi(\frac{\Phi^{-1}(F_s(x_s))-\mu_s}{\sigma_s}))}{\sigma_s \phi(\Phi^{-1}(F_s(x_s)))} f_s(x_s) \\ & = c^{\mathrm{Gauss}}(G(\boldsymbol{x}_A) \,|\, \widetilde{R}) \prod_{t \in A} g_t(x_t)\,. \end{aligned}$$

For this reduction to be a valid copula model, we must also verify that the derivative of $G_s(x)$ is $g_s(x)$:

$$\begin{aligned}
\frac{\mathrm{d}G_s(x)}{\mathrm{d}x} &= \frac{\mathrm{d}\big(\Phi(\frac{\Phi^{-1}(F_s(x))-\mu_s}{\sigma_s})\big)}{\mathrm{d}x} \\
&= \phi\left(\frac{\Phi^{-1}(F_s(x))-\mu_s}{\sigma_s}\right)\frac{1}{\sigma_s}\frac{\mathrm{d}\big(\Phi^{-1}(F_s(x))-\mu_s\big)}{\mathrm{d}x} \\
&= \phi\left(\frac{\Phi^{-1}(F_s(x))-\mu_s}{\sigma_s}\right)\frac{1}{\sigma_s}\frac{1}{\phi(\Phi^{-1}(F_s(x)))}\frac{\mathrm{d}F_s(x)}{\mathrm{d}x} \\
&= \phi\left(\frac{\Phi^{-1}(F_s(x))-\mu_s}{\sigma_s}\right)\frac{1}{\sigma_s}\frac{1}{\phi(\Phi^{-1}(F_s(x)))}f_s(x) \\
&= \frac{\phi\left(\frac{\Phi^{-1}(F_s(x))-\mu_s}{\sigma_s}\right)}{\sigma_s\phi(\Phi^{-1}(F_s(x)))}f_s(x) = g_s(x)\,.
\end{aligned}$$

10.7.2 Induction Step

For the induction step, we will generally index the parameters for $|B| = k$ with superscripts. For example, the induction hypothesis for $|B| = k$ will have parameters $G^{(k)}(\cdot)$ and $\widetilde{R}^{(k)}$. We will assume w.l.o.g that we are conditioning on the last k or $k+1$ variables—i.e. $B^{(k)} = \{p-k+1, p-k+2, \cdots, p\}$ and $B^{(k+1)} = \{p-k, p-k+1, \cdots, p\}$. Let us consider

the case of $k+1$:

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_{A^{(k+1)}} \,|\, \boldsymbol{x}_{B^{(k+1)}}, R, F, f) \\
&= \frac{\mathbb{P}(\boldsymbol{x})}{\mathbb{P}(\boldsymbol{x}_{B^{(k+1)}})} \\
&= \frac{\mathbb{P}(\boldsymbol{x})}{\mathbb{P}(\boldsymbol{x}_{B^{(k)}})\mathbb{P}(\boldsymbol{x}_{p-k} \,|\, \boldsymbol{x}_{B^{(k)}})} \\
&= \frac{\mathbb{P}(\boldsymbol{x}_{A^{(k)}} \,|\, \boldsymbol{x}_{B^{(k)}})}{\mathbb{P}(\boldsymbol{x}_{p-k} \,|\, \boldsymbol{x}_{B^{(k)}})} \\
&= \frac{c^{\text{Gauss}}(G^{(k)}(\boldsymbol{x}_A) \,|\, \widetilde{R}^{(k)}) \prod_{t=1}^{p-k} g_t^{(k)}(x_t)}{g_{p-k}^{(k)}(x_{p-k})} \\
&= c^{\text{Gauss}}(G^{(k)}(\boldsymbol{x}_{A^{(k)}}) \,|\, \widetilde{R}^{(k)}) \prod_{t=1}^{p-k-1} g_t^{(k)}(x_t)
\end{aligned}$$

Now we invoke Lemma 1 with parameters $\bar{\Sigma}$, $\bar{\boldsymbol{\sigma}}$, $\bar{\boldsymbol{\mu}}$, $\bar{R}$ and conditioning value $G_{p-k-1}^{(k)}(x_{p-k-1}) = \frac{\Phi^{-1}(F_{p-k-1}(x_{p-k-1})) - \mu_{p-k-1}^{(k)}}{\sigma_{p-k-1}^{(k)}}$:

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_{A^{(k+1)}} \,|\, \boldsymbol{x}_{B^{(k+1)}}, R, F, f) \\
&= c^{\text{Gauss}}(\Phi(\text{diag}(\bar{\boldsymbol{\sigma}})^{-1}\Phi^{-1}(G^{(k)}(\boldsymbol{x}_{A^{(k)}})) - \bar{\boldsymbol{\mu}})) \,|\, \bar{R}) \\
&\quad \times \prod_{s=1}^{p-k-1} \frac{\phi(\Phi(\frac{\Phi^{-1}(G_s^{(k)}(x_s)) - \bar{\mu}_s}{\bar{\sigma}_s}))}{\bar{\sigma}_s \phi(\Phi^{-1}(G_s^{(k)}(x_s)))} g_t^{(k)}(x_t)
\end{aligned}$$

To simplify this expression, we note that

$$\begin{aligned}
&\Phi^{-1}(G_s^{(k)}(x_s)) \\
&= \Phi^{-1}(\Phi(\frac{\Phi^{-1}(F(x_s)) - \mu_s^{(k)}}{\sigma_s^{(k)}})) \\
&= \frac{\Phi^{-1}(F(x_s)) - \mu_s^{(k)}}{\sigma_s^{(k)}} \,.
\end{aligned}$$

We can now simplify the independent product terms:

$$\begin{aligned}
&\frac{\phi(\Phi(\frac{\Phi^{-1}(G_s^{(k)}(x_s))-\bar{\mu}_s}{\bar{\sigma}_s}))}{\bar{\sigma}_s\phi(\Phi^{-1}(G_s^{(k)}(x_s)))}g_t^{(k)}(x_t) \\
&= \frac{\phi(\Phi\left(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}-\bar{\mu}_s\sigma_s^{(k)}}{\sigma_s^{(k)}\bar{\sigma}_s}\right))}{\bar{\sigma}_s\phi(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}}{\sigma_s^{(k)}})}g_t^{(k)}(x_t) \\
&= \frac{\phi(\Phi\left(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}-\bar{\mu}_s\sigma_s^{(k)}}{\sigma_s^{(k)}\bar{\sigma}_s}\right))}{\bar{\sigma}_s\phi(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}}{\sigma_s^{(k)}})}\frac{\phi(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}}{\sigma_s^{(k)}})}{\sigma_s^{(k)}\phi(\Phi^{-1}(F(x_s)))} \\
&= \frac{\phi(\Phi\left(\frac{\Phi^{-1}(F(x_s))-\mu_s^{(k)}-\bar{\mu}_s\sigma_s^{(k)}}{\sigma_s^{(k)}\bar{\sigma}_s}\right))}{\sigma_s^{(k)}\bar{\sigma}_s\phi(\Phi^{-1}(F(x_s)))}
\end{aligned}$$

Now we will show that $\Sigma^{(k+1)} = \text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})\bar{\Sigma}\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})$:

$$\begin{aligned}
\bar{\Sigma}^{-1} &= [(\widetilde{R}^{(k)})^{-1}]_{A^{(k+1)},A^{(k+1)}} \\
&= \left[\text{diag}(\boldsymbol{\sigma}^{(k)})W^{(k)}_{A^{(k)},A^{(k)}}\text{diag}(\boldsymbol{\sigma}^{(k)})\right]_{A^{(k+1)},A^{(k+1)}} \\
&= \text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})W^{(k)}_{A^{(k+1)},A^{(k+1)}}\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}}) \\
&= \text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})W^{(k+1)}\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}}) \\
&= \text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})(\Sigma^{(k+1)})^{-1}\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})
\end{aligned}$$

where the second to last step is because $W^{(k+1)}$ is merely the top upper submatrix of $W^{(k)}$. From this we know that:

$$\begin{aligned}
\boldsymbol{\sigma}^{(k+1)} &= \sqrt{\text{diag}(\Sigma^{(k+1)})} \\
&= \sqrt{\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})\bar{\Sigma}\text{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})} \\
&= [\sigma_1^{(k)}\bar{\sigma}_1, \sigma_2^{(k)}\bar{\sigma}_2, \cdots, \sigma_{p-k-1}^{(k)}\bar{\sigma}_{p-k-1}]
\end{aligned}$$

and

$$\begin{aligned}
\bar{R} &= \mathrm{diag}(\bar{\boldsymbol{\sigma}})^{-1}\bar{\Sigma}\mathrm{diag}(\bar{\boldsymbol{\sigma}})^{-1} \\
&= \mathrm{diag}(\bar{\boldsymbol{\sigma}})^{-1}(\mathrm{diag}(\boldsymbol{\sigma}^{(k)})^{-1}\Sigma^{(k+1)}\mathrm{diag}(\boldsymbol{\sigma}^{(k)})^{-1})\mathrm{diag}(\bar{\boldsymbol{\sigma}})^{-1} \\
&= \mathrm{diag}(\boldsymbol{\sigma}^{(k+1)})^{-1}\Sigma^{(k+1)}\mathrm{diag}(\boldsymbol{\sigma}^{(k+1)})^{-1} \\
&= \widetilde{R}^{(k+1)}\,.
\end{aligned}$$

Finally, we show that $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \mathrm{diag}(\boldsymbol{\sigma}^{(k)}_{A^{(k+1)}})\bar{\boldsymbol{\mu}}$. Letting $\tilde{p} = \{p-k-1\}$ $\boldsymbol{z} = \Phi^{-1}(F(\boldsymbol{x}))$ and $\bar{z} = \frac{z_{\tilde{p}} - \mu^{(k)}_{\tilde{p}}}{\sigma^{(k)}_{\tilde{p}}}$, we derive the formulas in terms of W and using the following substitution $\boldsymbol{q} = W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}\tilde{p}}$:

$$\begin{aligned}
\boldsymbol{\mu}^{(k)} &= -W^{-1}_{A^{(k)}A^{(k)}}W_{A^{(k)}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \\
\boldsymbol{\mu}^{(k+1)} &= -W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}B^{(k+1)}}\boldsymbol{z}_{B^{(k+1)}} \\
\bar{\boldsymbol{\mu}} &= -\bar{W}^{-1}_{A^{(k+1)}A^{(k+1)}}\bar{W}_{A^{(k+1)}\tilde{p}}\bar{z}
\end{aligned}$$

Let's expand $\bar{\boldsymbol{\mu}}$:

$$\begin{aligned}
&\bar{\boldsymbol{\mu}} \\
&= -(\mathrm{diag}(\boldsymbol{\sigma}^{(k)})W_{A^{(k+1)}A^{(k+1)}}\mathrm{diag}(\boldsymbol{\sigma}^{(k)}))^{-1} \\
&\quad \times (\mathrm{diag}(\boldsymbol{\sigma}^{(k)})W_{A^{(k+1)}\tilde{p}}\sigma^{(k)}_{\tilde{p}})\bar{z} \\
&= -\mathrm{diag}(\boldsymbol{\sigma}^{(k)})^{-1}W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}\tilde{p}}\sigma^{(k)}_{\tilde{p}} \\
&\quad \times \left(\frac{z_{\tilde{p}} + \mu^{(k)}_{\tilde{p}}}{\sigma^{(k)}_{\tilde{p}}}\right) \\
&= -\mathrm{diag}(\boldsymbol{\sigma}^{(k)})^{-1}\boldsymbol{q}(z_{\tilde{p}} - \mu^{(k)}_{\tilde{p}})\,.
\end{aligned}$$

Now let's decompose $\boldsymbol{\mu}^{(k)}$ in terms of submatrices and subvectors:

$$
\begin{aligned}
&\begin{bmatrix} \boldsymbol{\mu}^{(k)}_{A^{(k+1)}} \\ \mu^{(k)}_{\tilde{p}} \end{bmatrix} \\
&= -\begin{bmatrix} W_{A^{(k+1)}A^{(k+1)}} & W_{A^{(k+1)}\tilde{p}} \\ W^T_{A^{(k+1)}\tilde{p}} & W_{\tilde{p}\tilde{p}} \end{bmatrix}^{-1} \\
&\quad \times \begin{bmatrix} W_{A^{(k+1)}B^{(k)}} \\ W_{\tilde{p}B^{(k)}} \end{bmatrix} \times \boldsymbol{z}_{B^{(k)}} \\
&= -\begin{bmatrix} W^{-1}_{A^{(k+1)}A^{(k+1)}} + \boldsymbol{q}(\sigma^{(k)}_{\tilde{p}})^2\boldsymbol{q}^T & \boldsymbol{q}(\sigma^{(k)}_{\tilde{p}})^2 \\ (\sigma^{(k)}_{\tilde{p}})^2\boldsymbol{q}^T & (\sigma^{(k)}_{\tilde{p}})^2 \end{bmatrix} \\
&\quad \times \begin{bmatrix} W_{A^{(k+1)}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \\ W_{\tilde{p}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \end{bmatrix} \\
&= \begin{bmatrix} -W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}B^{(k)}}\boldsymbol{z}_{B^{(k)}} - \boldsymbol{q}(\sigma^{(k)}_{\tilde{p}})^2\boldsymbol{q}^T \\ -(\sigma^{(k)}_{\tilde{p}})^2\boldsymbol{q}^T W_{A^{(k+1)}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \end{bmatrix} \\
&+ \begin{bmatrix} -\boldsymbol{q}(\sigma^{(k)}_{\tilde{p}})^2 W_{\tilde{p}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \\ -(\sigma^{(k)}_{\tilde{p}})^2 W_{\tilde{p}B^{(k)}}\boldsymbol{z}_{B^{(k)}} \end{bmatrix} \\
&= \begin{bmatrix} -W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}B^{(k)}}\boldsymbol{z}_{B^{(k)}} - \boldsymbol{q}\mu^{(k)}_{\tilde{p}} \\ \mu^{(k)}_{\tilde{p}} \end{bmatrix}.
\end{aligned}
$$

Now let's expand $\boldsymbol{\mu}^{(k+1)}$:

$$
\begin{aligned}
&\boldsymbol{\mu}^{(k+1)} \\
&= -W^{-1}_{A^{(k+1)}A^{(k+1)}} \begin{bmatrix} W_{A^{(k+1)}\tilde{p}} & W_{A^{(k+1)}B^{(k)}} \end{bmatrix} \begin{bmatrix} z_{\tilde{p}} \\ \boldsymbol{z}_{B^{(k)}} \end{bmatrix} \\
&= -\boldsymbol{q}z_{\tilde{p}} - W^{-1}_{A^{(k+1)}A^{(k+1)}}W_{A^{(k+1)}B^{(k)}}\boldsymbol{z}_{B^{(k)}}.
\end{aligned}
$$

Finally, we look at $\boldsymbol{\mu}^{(k)}_{A^{(k+1)}} + \mathrm{diag}(\boldsymbol{\sigma}^{(k)})\bar{\boldsymbol{\mu}}$:

$$\begin{aligned}
&\boldsymbol{\mu}^{(k)} + \mathrm{diag}(\boldsymbol{\sigma}^{(k)})\bar{\boldsymbol{\mu}} \\
&= (-W^{-1}_{A^{(k+1)}A^{(k+1)}} W_{A^{(k+1)}B^{(k)}} \boldsymbol{z}_{B^{(k)}} - \boldsymbol{q}\mu^{(k)}_{\tilde{p}}) \\
&\quad + \mathrm{diag}(\boldsymbol{\sigma}^{(k)})(-\mathrm{diag}(\boldsymbol{\sigma}^{(k)})^{-1}\boldsymbol{q}(z_{\tilde{p}} - \mu^{(k)}_{\tilde{p}})) \\
&= -W^{-1}_{A^{(k+1)}A^{(k+1)}} W_{A^{(k+1)}B^{(k)}} \boldsymbol{z}_{B^{(k)}} - \boldsymbol{q} z_{\tilde{p}} \\
&= \boldsymbol{\mu}^{(k+1)}
\end{aligned}$$

Substituting these equalities back into (10.19), we see that indeed the induction hypothesis for $(k+1)$ is true. □

10.7.3 Proof of Lemma 1

Proof. We want to decompose the Gaussian copula as per the Lemma 1:

$$\begin{aligned}
&c^{\mathrm{Gauss}}(F(\boldsymbol{x}) \,|\, R) \qquad (10.19)\\
&= |R|^{-1/2} \exp\big(-\frac{1}{2}\Phi^{-1}(F(\boldsymbol{x}))^T(R^{-1} - \boldsymbol{I})\Phi^{-1}(F(\boldsymbol{x}))\big).
\end{aligned}$$

Let us substitute $\boldsymbol{z} = \Phi^{-1}(F(\boldsymbol{x}))$ and $W = R^{-1}$ and decompose the quadratic based on $\boldsymbol{z}_A$ and z_p as follows:

$$\begin{aligned}
c^{\mathrm{Gauss}}(F(\boldsymbol{x}) \,|\, R) &= |W|^{1/2} \exp\big(-\frac{1}{2}\boldsymbol{z}^T(W - \boldsymbol{I})\boldsymbol{z}\big) \qquad (10.20)\\
&= |W|^{1/2} \exp\big\{-\frac{1}{2}\big[\boldsymbol{z}_A^T(W_{AA} - \boldsymbol{I}_{AA})\boldsymbol{z}_A \qquad (10.21)\\
&\qquad + 2z_p(W_{pA} - \boldsymbol{I}_{pA})\boldsymbol{z}_A + z_p^2(w_{pp} - 1)\big]\big\} \\
&= |W|^{1/2} \exp\big\{-\frac{1}{2}\big[\boldsymbol{z}_A^T(W_{AA} - \boldsymbol{I})\boldsymbol{z}_A \qquad (10.22)\\
&\qquad + 2z_p W_{pA}\boldsymbol{z}_A + z_p^2(w_{pp} - 1)\big]\big\}
\end{aligned}$$

We now define $\boldsymbol{\mu}$, Σ, $\boldsymbol{\sigma}$ and $\widetilde{R}$ as in the theorem and simplify:

$$\boldsymbol{\mu} = R_{pA}R_{pp}^{-1}\Phi^{-1}(F_p(\boldsymbol{x}_p)) = R_{pA}\Phi^{-1}(F_p(\boldsymbol{x}_p)) \tag{10.23}$$

$$\Sigma = R_{AA} - R_{Ap}R_{pp}^{-1}R_{pA} = R_{AA} - R_{Ap}R_{Ap}^T \tag{10.24}$$

$$\sigma = \sqrt{\text{diag}(\Sigma)} \tag{10.25}$$

$$\widetilde{R} = \text{diag}(\sigma)^{-1}\Sigma\text{diag}(\sigma)^{-1}, \tag{10.26}$$

where the simplifications are by the fact that $R_{pp} = 1$ and R is symmetric. Note also that $W_{AA} = \Sigma^{-1}$. The notation is suggestive about the role of each variable in terms of the mean, covariance matrix, standard deviation and correlation matrix of the conditioned set. We now make the following substitution:

$$\boldsymbol{y} = \text{diag}(\boldsymbol{\sigma})^{-1}(\boldsymbol{z}_A - \boldsymbol{\mu}) \tag{10.27}$$

$$\boldsymbol{z}_A = \text{diag}(\boldsymbol{\sigma})\boldsymbol{y} + \boldsymbol{\mu} \tag{10.28}$$

We simplify each term of the quadratic in (10.22) based on these substitutions:

$$\begin{aligned}
&\boldsymbol{z}_A^T(W_{AA} - \boldsymbol{I})\boldsymbol{z}_A \\
&= (\text{diag}(\boldsymbol{\sigma})\boldsymbol{y} + \boldsymbol{\mu})^T(W_{AA} - \boldsymbol{I})(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y} + \boldsymbol{\mu}) \\
&= (\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^T(W_{AA} - \boldsymbol{I})(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y}) \\
&\quad + 2(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^T(W_{AA} - \boldsymbol{I})\boldsymbol{\mu} \\
&\quad + \boldsymbol{\mu}^T(W_{AA} - \boldsymbol{I})\boldsymbol{\mu} \\
&= \boldsymbol{y}^T(\text{diag}(\boldsymbol{\sigma})W_{AA}\text{diag}(\boldsymbol{\sigma}))\boldsymbol{y} - \boldsymbol{y}^T\text{diag}(\boldsymbol{\sigma})^2\boldsymbol{y} \\
&\quad + 2(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^T(W_{AA} - \boldsymbol{I})\boldsymbol{\mu} \\
&\quad + \boldsymbol{\mu}^T(W_{AA} - \boldsymbol{I})\boldsymbol{\mu} \\
&= \boldsymbol{y}^T(\widetilde{R} - \boldsymbol{I})\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\text{diag}(\boldsymbol{\sigma})^2\boldsymbol{y} \\
&\quad + 2(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^TW_{AA}\boldsymbol{\mu} - 2(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^T\boldsymbol{\mu} \\
&\quad + \boldsymbol{\mu}^TW_{AA}\boldsymbol{\mu} - \boldsymbol{\mu}^T\boldsymbol{\mu}\,,
\end{aligned} \tag{10.29}$$

where the last equality is merely by adding and subtracting $\boldsymbol{y}^T\boldsymbol{y}$. Note that the first term has the form of a Gaussian copula with correlation matrix $\widetilde{R}$. We will now expand the second term of the quadratic in (10.22) using substitution and the matrix identities for R and W described previously:

$$\begin{aligned}
&2z_pW_{pA}\boldsymbol{z}_A \\
&= 2z_p(-R_{pp}^{-1}R_{pA}(R_{AA}-R_{Ap}R_{pp}^{-1}R_{pA})^{-1})(\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu}) \\
&= -2z_pR_{pA}W_{AA}(\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu}) \\
&= -2\boldsymbol{\mu}^TW_{AA}(\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu}) \\
&= -2(\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{y})^TW_{AA}\boldsymbol{\mu}-2\boldsymbol{\mu}^TW_{AA}\boldsymbol{\mu}\,. \qquad (10.30)
\end{aligned}$$

Finally, we now expand the last term of (10.22):

$$\begin{aligned}
&z_p^2(w_{pp}-1) \\
&= z_p^2((R_{pp}^{-1}+R_{pp}^{-1}R_{pA}(R_{AA}-R_{Ap}R_{pp}^{-1}R_{pA})^{-1}R_{Ap}R_{pp}^{-1})-1) \\
&= z_p^2((1+R_{pA}W_{AA}R_{Ap})-1) \\
&= \boldsymbol{\mu}^TW_{AA}\boldsymbol{\mu}\,. \qquad (10.31)
\end{aligned}$$

Combining the term expansions from (10.29), (10.30), and (10.31) into (10.22) and canceling terms, we arrive at the following:

$$
\begin{aligned}
&c^{\text{Gauss}}(F(\boldsymbol{x}) \,|\, R) \\
&= \frac{|W|^{1/2}}{|\widetilde{R}|^{1/2}} |\widetilde{R}|^{1/2} \exp\Big\{-\frac{1}{2}\boldsymbol{y}^T(\widetilde{R}-\boldsymbol{I})\boldsymbol{y}\Big\} \\
&\times \exp\Big\{-\frac{1}{2}\big[\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T \text{diag}(\boldsymbol{\sigma})^2\boldsymbol{y} - 2(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y})^T\boldsymbol{\mu} - \boldsymbol{\mu}^T\boldsymbol{\mu}\big]\Big\} \\
&= |\text{diag}(\boldsymbol{\sigma})|^{-1} \, |\widetilde{R}|^{1/2} \exp\Big\{-\frac{1}{2}\boldsymbol{y}^T(\widetilde{R}-\boldsymbol{I})\boldsymbol{y}\Big\} \\
&\quad \times \frac{\exp\big\{-\frac{1}{2}\big[\boldsymbol{y}^T\boldsymbol{y}\big]\big\}}{\exp\big\{-\frac{1}{2}(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu})^T(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu})\big\}} \\
&= c^{\text{Gauss}}(\boldsymbol{y} \,|\, \widetilde{R}) \frac{|\text{diag}(\boldsymbol{\sigma})|^{-1} \;\exp\big\{-\frac{1}{2}\big[\boldsymbol{y}^T\boldsymbol{y}\big]\big\}}{\exp\{-\frac{1}{2}(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu})^T(\text{diag}(\boldsymbol{\sigma})\boldsymbol{y}+\boldsymbol{\mu})\}} \\
&= c^{\text{Gauss}}(\boldsymbol{y} \,|\, \widetilde{R}) \prod_{s=1}^{p} \sigma_s^{-1} \frac{\phi(y_s)}{\phi(\sigma_s y_s + \mu_s)},
\end{aligned}
\tag{10.32}
$$

where the second to last step is by the fact that:

$$
\begin{aligned}
\frac{|W|^{1/2}}{|\widetilde{R}|^{1/2}} &= \left(\frac{|W|}{|\text{diag}(\boldsymbol{\sigma})W_{AA}\text{diag}(\boldsymbol{\sigma})|}\right)^{1/2} \\
&= |\text{diag}(\boldsymbol{\sigma})|^{-1} \left(\frac{|W_{AA}||W_{pp} - W_{pA}W_{AA}^{-1}W_{Ap}|}{|W_{AA}|}\right)^{1/2} \\
&= |\text{diag}(\boldsymbol{\sigma})|^{-1} \, |W_{pp} - W_{pA}W_{AA}^{-1}W_{Ap}|^{1/2} \\
&= |\text{diag}(\boldsymbol{\sigma})|^{-1} \, |R_{pp}|^{1/2} \\
&= |\text{diag}(\boldsymbol{\sigma})|^{-1} .
\end{aligned}
$$

We now undo the substitution of $\boldsymbol{y}$ and $\boldsymbol{z}_A$ to get (10.32) in terms of the original variables and functions:

$$c^{\text{Gauss}}(F(\boldsymbol{x}) \mid R) \tag{10.33}$$

$$= c^{\text{Gauss}}(\Phi(\text{diag}(\boldsymbol{\sigma})^{-1}\Phi^{-1}(F_A(\boldsymbol{x}_A)) - \boldsymbol{\mu})) \mid \widetilde{R}) \tag{10.34}$$

$$\times \prod_{s=1}^{p-1} \frac{\phi(\Phi(\frac{\Phi^{-1}(F_s(x_s))-\mu_s}{\sigma_s}))}{\sigma_s \phi(\Phi^{-1}(F_s(x_s)))}. \tag{10.35}$$

□

10.8 Experiments

In the following experiments, we demonstrate that by using the closed-form conditional marginals of Gaussian-copula models described in Theorem 1, we can predict missing values better using non-Gaussian marginals rather than Gaussian marginals—which would correspond to a standard multivariate normal or Gaussian graphical model. The key idea is that with only a small overhead in using non-Gaussian marginals, we can form more complex missing value models that give better predictions than a joint Gaussian model. Thus, in many situations where Gaussian marginals are not appropriate, we can still use a Gaussian-copula model to make useful predictions.

10.8.1 Experimental Setup

In our experiments, we hold out one hundred observations as a test set and randomly mark 10% of the values as missing. Note that the variables missing for each test instance is different so a single regression model cannot be used. For example, in the first test instance, variables 5, 20 and 101 may be missing whereas for the second test instance variables 1, 15 and 17 may be missing. Thus, only a joint probabilistic model can be used rather than a regression model.

We train our models on the rest of the data assuming all values are observed. While it is possible to estimate a joint model even from data with missing values by marginalizing out the missing values, we do not consider this in our experiments because model estimation is not our focus; rather we focus on using the model after estimation. In addition, the closed-form conditional marginals are agnostic to the original estimation procedure. However, for clarity, we present the details of our estimation procedure. First, we use the marginal-agnostic method called the non-paranormal SKEPTIC [Liu et al., 2012], which forms an estimate of the correlation matrix $\tilde{R}$ based on pairwise Spearman's ρ statistics. From these correlation matrix estimates, we then apply the elementary estimator from [Yang et al., 2014b] to estimate the Gaussian-copula inverse covariance matrix Σ^{-1}—such that non-zeros correspond to edges in the graphical model. We use a grid of parameters for the two soft thresholding operators from [Yang et al., 2014b]; more specifically, we used five log-spaced values between 0.5 and 10^{-5} for ν and ten log-spaced values between $\lambda_{\max}$ and $10^{-5}\lambda_{\max}$ for λ, where $\lambda_{\max}$ was the smallest lambda that would yield diagonal estimates—i.e. independent variables. We skipped any values for ν that did not yield a non-singular matrix and skipped any values of λ that did not yield a positive definite matrix. Finally, we post-process this result by normalizing the rows and columns so that $R = \text{diag}(\delta)\Sigma^{-1}\text{diag}(\delta)$ is a valid correlation matrix—i.e. ones on the diagonal. Note that this estimation of the dependency structure is completely independent of the marginal estimation. For marginal estimation, we merely use MATLAB's built-in functions (e.g. `gamfit` or `expfit`) to fit the marginal distributions.

10.8.2 Datasets

We select three diverse datasets to show the wide range of applicability of Gaussian-copula models. First, we collected a text dataset of natural science research papers from the department listing of a major university website with 10709 titles and 1000 variables (i.e. unique words). For each faculty member, we manually searched for their publications

webpage or CV if available. Then, we scraped the webpages or CVs for text and split their document based on their last names to approximate selecting titles. We approximated the publication year by the nearest year in the document and selected only documents with dates in the past 10 years. Clearly, this text dataset is very noisy; but this is representative of many real-world text datasets.

Second, we model a dataset of airport delay times to demonstrate that our model and visualization works for non-negative data as well. This dataset consisted of the average delay times at 183 airports for the 365 days of the year 2014. This is similar to the airport delay dataset in Chapter 4 except that we included 183 airports instead of just the top 30 hub airports. Third, we model daily stock returns data for 222 stocks from the six sectors of energy, finance, health care, public utilities, technology and transportation of the S&P500 from 2003-2012. This dataset shows that the Gaussian-copula model can handle general data including data with negative values and thick tails. We retrieved the raw data from the free data section on `https://quantquote.com/historical-stock-data`. We then extracted the company name and sectors from `http://www.nasdaq.com/screening/company-list.aspx` while manually renaming a few sectors that were marked 'N/A' or 'Other'.

Because we have freedom in selecting the marginal distributions of the data, we selected data specific distributions. For word count data, we selected the Poisson with continuous extension and the discrete Poisson marginals because the Poisson is a standard for count data. For airport delay times, we select the gamma distribution which is defined over non-negative real values and allows various shapes including the standard exponential distribution as as special case. Note that both the univariate Poisson and gamma distributions do not have known multivariate forms yet we can use them in this context because the Gaussian-copula distribution ties them together. Finally, for stock returns data, we choose the t distribution because it is well-known that daily returns have heavy tails (i.e. rare events) unlike the Gaussian distribution which has thin tails. These choices

emphasize that the Gaussian-copula model is much more flexible than the standard multivariate Gaussian.

10.8.3 Loss Functions

We use three different loss functions for missing value prediction to fully explore the differences between the Gaussian-copula models with varied marginals and the standard Gaussian graphical model. First, we consider the classical squared loss: $\mathcal{L}_2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$, whose corresponding estimator is the mean of the distribution. Second, we evaluate the quantile loss function $\mathcal{L}_u = \sum_s u\mathbb{I}(y_s \geq \hat{y}_s)|y_s - \hat{y}_s| + (1-u)\mathbb{I}(y_s < \hat{y}_s)|y_s - \hat{y}_s|$, whose corresponding estimator is the u-th quantile of the distribution. For example, if $u = 0.5$, this corresponds to median regression. We choose to evaluate the quantile loss because the tails of the distribution often behave differently under a Gaussian model compared to a non-Gaussian model. Finally, we consider a Spearman's rank correlation loss based on the predicted mean/median $\hat{\boldsymbol{y}}$ compared to the true values $\boldsymbol{y}$: $\mathcal{L}_\rho = 1 - \rho(\boldsymbol{y}, \hat{\boldsymbol{y}})$, where $\rho(\cdot)$ is the Spearman's rank correlation function. This measure is more of a global measurement of all the estimated missing values together rather than each missing value evaluated independently.

10.8.4 Results

The results can be seen in Fig. 10.1, Fig. 10.2 and Fig 10.3 for the research paper titles, airport delays and daily stock returns respectively. We show the results for the independent models on the left of each figure and the best dependent model on the right. Note that lower is better in all figures.

For the research paper titles (Fig. 10.1), we use the high quantiles 0.99 and 0.995 because word counts are usually zero so lower quantiles should almost always be predicted as zero but the high quantiles should actually predict a one or two. The discrete Poisson model performs well according to quantile loss but does not perform as well for squared loss.

The Poisson-CE model performs well for squared loss. Thus, the loss function may suggest which model is better for a particular application.

When viewing the airport delay results (Fig. 10.2), it is clear that the gamma marginal distribution with a dependent copula always produces a model that is at least as good (often significantly better) than the Gaussian model. Likely, the skewness and heavy-tailed part of the gamma distribution is important for modeling airport delay times.

Finally, we consider the daily stock returns in Fig. 10.3 where we compare the t distribution to the Gaussian distribution. Again, it is clear that the t marginal distributions performs at least as well if not better than Gaussian marginals. This aligns with the domain-knowledge that daily stock returns have heavy tails. Granted, the t distribution is close the the Gaussian under many properties and thus does not show quite as much difference as the other datasets.

10.9 Conclusion

We derived the closed-form conditional marginal distributions for Gaussian-copula models. This enables better missing value prediction under different loss functions than the standard multivariate Gaussian with only a small computational overhead in terms of prediction. We showed that these basic results extend to mixtures and Gaussian-copula models with discrete marginals. Finally, we presented some missing value prediction results showing that using non-Gaussian marginals can indeed enable better missing value predictions than the standard multivariate Gaussian model. We hope the simple idea of closed-form conditional marginals will enable better closed-form solutions to difficult modeling and prediction problems.

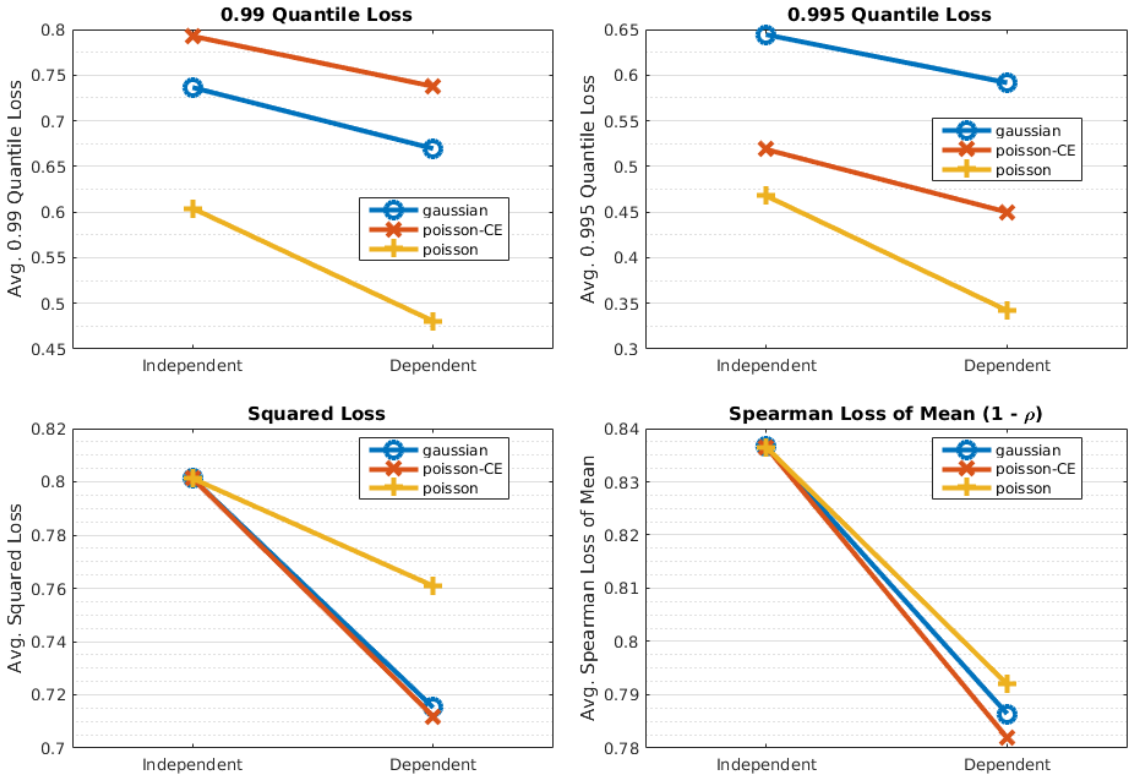


Figure 10.1: College of natural science research paper titles: The Gaussian-copula with Poisson marginal clearly outperforms the Gaussian graphical model for quantile regressions but does not perform as well for squared loss.

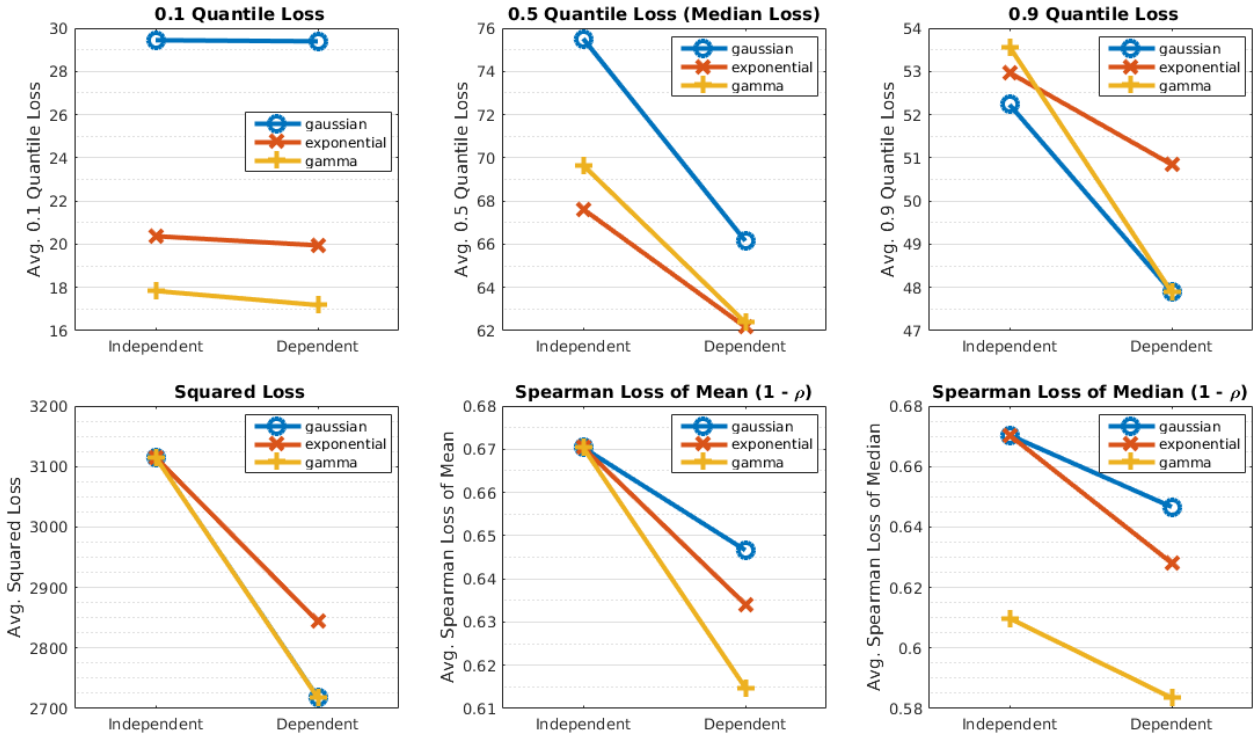


Figure 10.2: Airport delays: The Gaussian-copula with gamma marginals performs better or matches the performance of Gaussian graphical models.

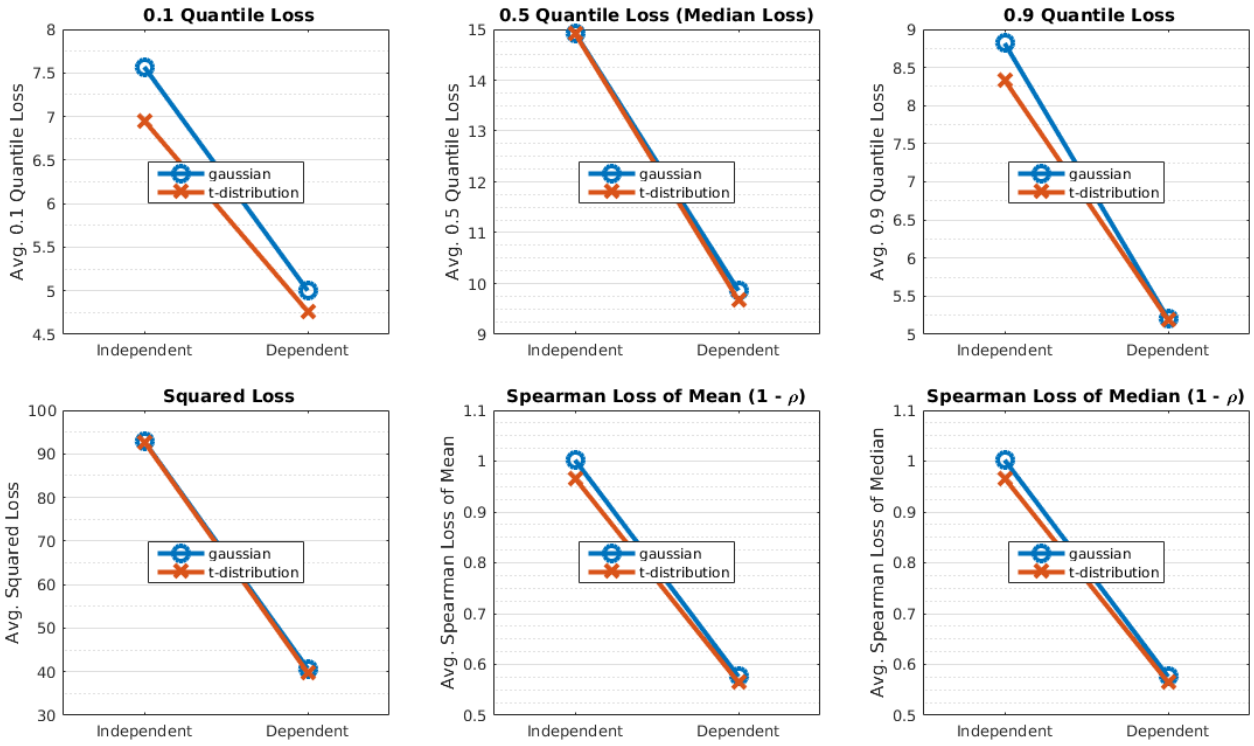


Figure 10.3: Daily stock returns: The Gaussian-copula with t distribution marginals performs better or matches the performance of Gaussian graphical models.

Part IV

Pretty, Principled and Probabilistic: Graphical Model Visualization

Summary of Part IV

The undirected graphical models described in previous chapters can provide a powerful yet flexible way to represent interesting patterns in high-dimensional data. Yet, because these models are high-dimensional, inspecting or interpreting an estimated model can be difficult even for a graphical model expert. Thus, in these next chapters, we seek to appropriately visualize graphical models in an intuitive, accessible and appealing way that does not require expert knowledge about graphical models. As a key observation, we notice that any static visualization of a graphical model would be inadequate. Therefore, we develop a novel interactive mechanism in Chapter 11 via conditional probability that enables a user to view different model perspectives. We frame the interaction idea in terms of asking 'what if?' questions so that a user does not need to understand the underlying probabilistic mechanism to use the interactive visualization.

In Chapter 12, we then propose to visualize these query-specific probabilistic graphical models by combining the intuitiveness of force-directed layouts with the beauty and readability of word clouds, which pack many words into valuable screen space while ensuring words do not overlap via pixel-level collision detection. Although both the force-directed layout and the pixel-level packing problems are challenging in their own right, we approximate both simultaneously via adaptive simulated annealing starting from careful initialization. For visualizing mixture distributions, we also design a meaningful mapping from the properties of the mixture distribution to a color in the perceptually uniform CIELUV color space. Finally, we demonstrate our approach via illustrative visualizations on real datasets.

Chapter 11

What If? Model and Interaction

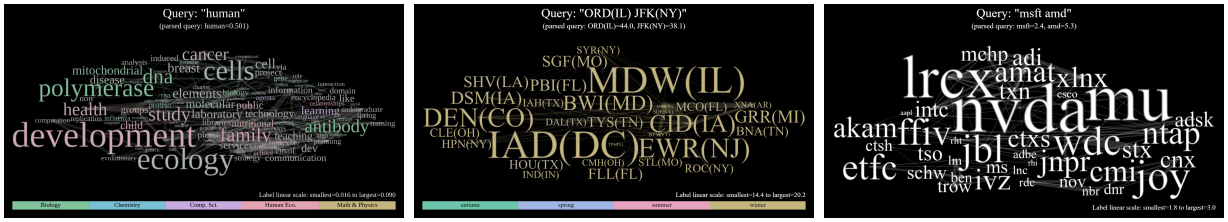


Figure 11.1: We operationalize 'what if' questions for both text and real-valued datasets by mapping simple text queries such as "human" or "msft amd" to concrete conditional probability operations on probabilistic models via quantile functions. Then, we visualize these query-specific models by combining both force-directed graph layout and pixel-level collision detection to avoid overlap. Datasets: (left) discrete word counts from natural science research paper titles, (middle) non-negative real-valued average delay times at airports and (right) real-valued daily stock returns.

11.1 Abstract

What if? questions are useful for initial exploratory analysis of diverse datasets. For example, we may ask "if we know a person is six feet tall, what is the best estimate of their weight?" Or, "if a document contains the word 'information', are the words 'visualization' and/or 'technology' more likely to appear in the document?" In this chapter, we operationalize such abstract questions into concrete operations on probability distributions which can in turn be intuitively and faithfully represented visually. Our approach defines: the probabilistic mechanism via conditional probability; the query language to map text input to a conditional probability query; and the formal underlying probabilistic model as

a Gaussian-copula graphical model—which has closed-form solutions even with hundreds of dimensions.

11.2 Introduction

During initial data analysis 'what if' questions are a natural way to interrogate data—and hopefully in turn better understand the underlying phenomena. For example, suppose an analyst has just received a set of patient survey responses from a hospital, she may then want to ask questions like: If a document contains the word "nurse", what positive or negative words are also likely to appear? And what if the word "doctor" occurs; are the most likely words similar to or different from when "nurse" occurs? This may afford insights into whether patients perceive doctors and nurses differently. As another example, consider sociologists trying to understand and characterize perceptions of gender. They may want to ask: If the word "he" appears in a text, what other words also tend to appear? And how do these differ from the words that are relatively frequently used when "she" appears in a text?

In the hard sciences, meanwhile, a biologist may have protein data from cell populations under different conditions and may want to ask questions like: If high levels of protein A are present, what other proteins tend also to be present? Or, suppose there are low levels of protein A; which other proteins tend to be present in this case? More generally, biologists often want to understand the underlying structure of biological pathways in organisms. The dependency structure between DNA, RNA and proteins is very complex, not least because there are thousands of entities. While probability models can be estimated for these complex structures, the models are often visualized using a simple static graph layout [Allen and Liu, 2013], which only provides a single high-level view. The ability to ask 'what if' questions of this model would provide a natural way for domain experts to study these complex structures. More generally, granting to analysts the ability to ask unplanned 'what if' questions would provide a powerful new tool for gleaning

understanding of overall trends and/or high level structures hidden in data. For example, in the case of text analytics, this might facilitate the detection of preprocessing errors (such as failure to remove important stopwords). And in scientific domains, interrogating data via arbitrary 'what if' queries may support data-driven hypothesis generation. The initial insights gained from this exploratory phase could critically inform additional in-depth analysis.

In the visualization community, data filtering can be viewed as helping to answer 'what if' questions. As an illustrative example, suppose we have the heights and weights of National Football League (NFL) players (see Figure 11.2) and the question is: If we know that a new football player weighs 275 pounds, can we estimate their height? One way to answer this would be to filter the data to players that weigh between 273 and 277. However, as can be seen in Figure 11.2, there is no data in the interval 273 to 277 and thus the filtering operation would be useless. While this filtering issue may seem minor when only filtering based on one dimension, the problem dramatically increases as the number of filtering dimensions grows due to the curse of dimensionality. In addition, data filtering cannot borrow information concerning patterns from other parts of the data. For example, if we expand the filter to weights between 265 and 285, we will then have a small number of data points with which to estimate height, but we will not be able to exploit the pattern that height and weight are correlated because the number of points is so small.

Machine learning models could provide one powerful computational approach to answering 'what if' questions. For example, in the NLF example above, a regression model could be fit to predict heights from weights. This model could then provide an estimated height for someone who weighs 275 pounds, even in the case that no one with this exact weight was present in the training data. Such predictions may be viewed as answering 'what if' questions that are mapped onto feature vectors. However, supervised machine learning requires these 'what if' questions to be formulated *a priori*—requiring

specification of both the variables used for prediction and the variables to be predicted. This *a priori* requirement renders the approach unsuitable for general exploratory data analysis. Furthermore, this problem becomes significantly more pronounced as the number of dimensions increases because the number of possible 'what if' questions increases dramatically.

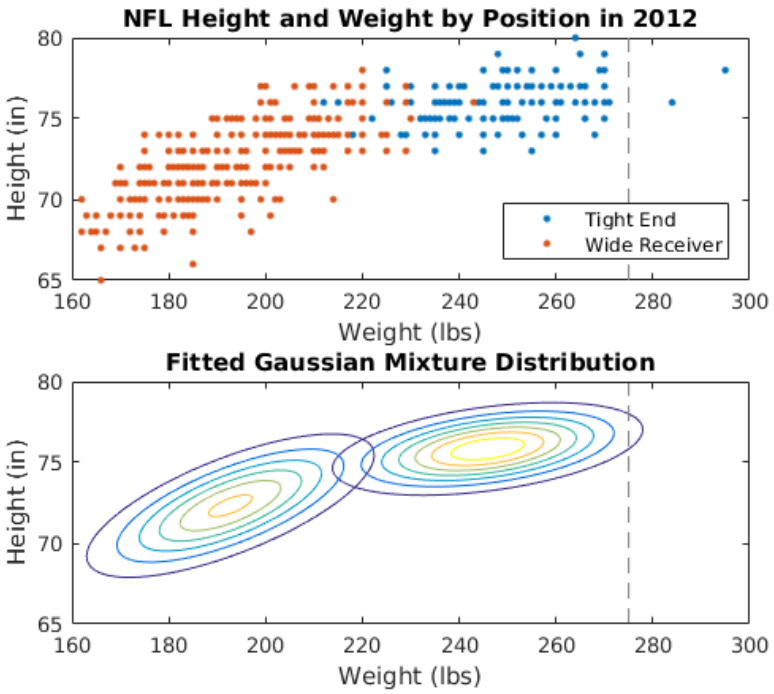


Figure 11.2: Height and weight of National Football League (NFL) players in 2012 for wide receivers and tight ends. Answering 'what if' questions via data filtering (top) can fail when there is little or no data in the region of interest such as a 275 pound NFL player (gray dashed line). However, a probabilistic model (bottom) can estimate the height of a (hypothetical) 275 pound player.

To sidestep the requirement of *a priori* specification, *unsupervised* machine learning such as clustering or low-rank matrix factorization may provide an alternative approach to answering 'what if' questions. For example, if the NFL data was clustered via k-means,

an individual's weight could be used to determine which cluster they belong to, and the (mean) height over other individuals also assigned to said cluster could be taken as a height estimate. Or, if the NFL data was projected onto a low-rank manifold (a line in the case of two dimensions), then height could be estimated by finding the intersection of the low-dimensional manifold and 275 pounds. Similarly, if the height is known, the weight could be estimated by projecting onto the manifold. However, these approaches would not provide estimates of uncertainty and would require the 'what if' question to be specified precisely (i.e., one would need to specify that an individual weighs *exactly* 275 pounds exactly rather than asking the question more broadly about a "heavy" person). By contrast, our proposed 'what if' query language maps implicit textual queries to explicit probabilistic queries via the underlying probability distribution as explained in section 11.4.

In light of these issues, we propose to use probabilistic models such as the Gaussian mixture model in Figure 11.2 to answer 'what if' questions, so that we can smooth over empty areas in the dataset and borrow power from statistical patterns in other parts of the dataset. We also propose an intuitive visualization of probabilistic models as a 'user interface' to these underlying probabilistic models. In this paper, we attempt to answer the following questions: (1) What is an appropriate underlying model for 'what if' questions? And how do we enable meaningful user interactions with this model? (1a) How do we operationalize abstract 'what if' questions into concrete operations on probability distributions? (1b) What is an intuitive way for users to specify 'what if' questions and how do we map this input to probabilistic operations? (2) How should these probabilistic models be visualized to help user understanding? (2a) What is an meaningful yet aesthetically-pleasing layout of the variable labels and how do we approximate such a layout? (2b) How do we intuitively encode probabilistic concepts as color?

The contributions of this paper are initial solutions to these questions. In particular, we propose the *Pretty, Principled and Probabilistic* framework, which instantiates a graphical

model as the underlying formal representation of the data, and then facilitates interrogation via conditional probability queries. We propose visualizing the results of these using an approach that is principled (in that it maps onto the underlying model) and pretty (borrowing aesthetics from word clouds).

In this chapter, we describe our proposed 'what if' probabilistic interaction mechanism and the query language that maps simple textual input queries onto it. Our proposed mechanism and query language could be used with any probabilistic model, but here we will argue for our choice of Gaussian-copula probabilistic graphical model class and mixtures thereof. Then, while our proposed mechanism and query language could apply to any probabilistic model, we argue for our choice of the Gaussian-copula probabilistic graphical model class and mixtures thereof that were described in Chapter 5 and Chapter 10.

11.3 Probabilistic Mechanism: Conditional Probability

We propose *conditional probability distributions* as the mechanism for answering 'what if' questions. At their core, 'what if' questions are concerned with inspecting a particular subspace of the original data space. In the example using the NFL player data discussed above, the 'what if' question seeks to inspect the subspace of possible players who are 275 pounds (or just about). As described in the introduction, data filtering is one way to explore this subspace. But data filtering is severely limited because it reduces the dataset significantly and cannot borrow patterns or trends from other areas of the dataset. However, if a probabilistic model over the data space is known (or estimated), conditional probability distributions provide a model-based filtering mechanism that generalizes beyond the training data—e.g. the heights of (hypothetical) players weighing 75, 175 and 275 pounds can be estimated regardless of whether we have seen data for players with these particular weights. Additionally, conditional distributions can exploit global patterns

evident in the data—e.g. the correlation between weight and height. Informally, conditional distributions are the distribution of filtered observations—given an infinite amount of data—around the conditioning values as the filter width approaches zero. For example, we might filter the NFL data to players that weigh between 255 and 295 pounds, a filter width of 20 pounds. If we had an infinite number of NFL players and we reduced the filter width to 0.1 pounds, we would be approaching the conditional distribution of NFL player height given a player weight of 275 pounds.

More formally, if we are modeling data with p variables denoted $\boldsymbol{x} = (x_1, x_2, \cdots, x_p)$, the conditional probability of one subset of variables $A \subset \{1, 2, \cdots, p\}$ given the other set of variables $B = A^C$ (such that $A \cup B = \{1, 2, \cdots, p\}$) is defined as: $\mathbb{P}(\boldsymbol{x}_A \,|\, \boldsymbol{x}_B) = \frac{\mathbb{P}(\boldsymbol{x}_A, \boldsymbol{x}_B)}{\mathbb{P}(\boldsymbol{x}_B)}$, where $\mathbb{P}(\boldsymbol{x}_A, \boldsymbol{x}_B)$ is the joint distribution over the two variable subsets and $\mathbb{P}(\boldsymbol{x}_B)$ is the marginal density at $\boldsymbol{x}_B$. Intuitively, this is a renormalized "slice" of the joint distribution fixed at the given values of $\boldsymbol{x}_B$. The conditional probability distribution is defined only over the domain of $\boldsymbol{x}_A$, which has fewer variables than the original distribution because $\boldsymbol{x}_B$ is fixed.

11.4 Query Language

Given the underlying probabilistic mechanism, we now consider how to elicit 'what if' queries from users, which entails three steps. First, we choose text-based queries as the input because they are simple to input, human interpretable, and portable—i.e. they can be easily represented and passed between different software programs or protocols. An alternative would be to provide a dropdown menu of possible variables followed by a value input box or input toggle. While this alternative would be more precise and preclude syntactical errors, it would likely require more user effort, especially when conditioning on multiple variables. Moreover, this type of exact input interface could be built on top of a text-based query language similar to building a database search on top of the SQL query language. Second,

we assume that each variable has a unique label so that a user does not have to specify the index of the variable. For example, the labels could be the strings "height" and "weight" for the NFL data, or if each variable corresponds to the count of a unique word in a document, then the variable label would be the word itself such as "computer" or "visualization". This assumption allows intuitive selection of conditioning variables without knowing the variable index. Third, we map the text queries to probabilistic queries using the quantile function, or inverse cumulative distribution function, of each variable. The cumulative distribution function (CDF) of a random variable is defined as: $F(x) = \mathbb{P}(X < x)$, where X is the random variable and x is a particular value of the variable. The quantile function $F^{-1}(\cdot)$ is defined as the generalized inverse CDF: $F^{-1}(u) = \inf\{x \in \mathbb{R} \,|\, F(x) = u\}$, where u is the target probability. For example, the median is equal to $F^{-1}(0.5)$, i.e. the probability of being less than the median is $u = 0.5$. As another example, the probability of being less than the 0.99 quantile is 0.99 and the probability of being greater than the 0.99 quantile is $0.01 = 1 - 0.99$. Given target quantiles for each variable in the conditioning set denoted $\{u_b \forall b \in |B|)$, we map the text query to the following conditional distribution:$\mathbb{P}(\boldsymbol{x}_A \,|\, \{x_b = F^{-1}(u_b), \forall b \in B\})$.

We choose the default target quantile to be $u = 0.95$ because we want to strongly emphasize the effect of the variable conditioning. Roughly, this means that we condition on each query variable being near the top 5% of its possible variable values. This ensures that the differences between no query and query are pronounced and noticeable; this idea is like caricature drawings that emphasize the unique characteristics of a person. As an example, if we conditioned NFL player weights using $u = 0.5$, i.e. the median value, then the median height is almost the same with or without the conditioning on the weight and thus would be nearly unnoticeable; however, if we condition on a high weight of 275 pounds, then the median height given a weight of 275 pounds is clearly different from the median height of all players. We also introduce simple "+" and "-" modifiers that can adjust the value of u: "+" $\rightarrow u = 0.99$, "++" $\rightarrow u = 0.999$, "-" $\rightarrow u = 0.05$, "–" $\rightarrow u = 0.01$, "—" $\rightarrow u = 0.001$.

In particular, these modifiers allow the user to condition on the left tail of the distribution rather than the right tail.

The true power of using the quantile function for mapping 'what if' queries is that a user does not have to provide a specific value or even know the domain of the variable values. For example, a user can specify the query "++weight" or "-weight" for the NFL data and condition on very heavy or light players respectively without having to know the weight distribution of NFL players. Or, for a biological dataset, a user could specify the query "proteinA proteinB" and know that they are conditioning on large counts of both proteinA and proteinB without having to know the distribution of raw counts of proteinA or proteinB, which may on the order of 10 for proteinA and the order of 100 for proteinB. In all cases, the interface is the same for the user regardless of the dataset and the burden of knowing specific values for each variable is allocated to the quantile function.

11.5 Underlying Probabilistic Model

While our proposed approach of mapping implicit 'what if' text queries to concrete probabilistic operations can be used for any probabilistic model, we choose the Gaussian-copula probabilistic graphical model class because (1) these models are flexible and powerful, and, (2) the dependency structure can be intuitively represented as a graph over variables, and, (3) the required operations and statistics have closed-form solutions. The Gaussian-copula model class has the properties of both probabilistic undirected graphical models and copula-based probabilistic models.

For our purposes, the two main advantages of the undirected graphical model class are: (1) the representation of the dependency structure as a graph over variables lends itself to intuitive graph visualizations, and, (2) the conditional distributions as described in section 11.3 can be computed in closed-form. The advantage of copula models, such as the Gaussian-copula model paired with non-Gaussian marginals, is that they afford *closed-form*

expressions of the marginal distributions, which in turn provide closed-form solutions to the quantile functions required for the query mapping described in section 11.4, and by the font size computation described below. We use the continuous extension (CE) way of handling discrete data described in Sec. 10.6 of Chapter 10. For more information on copula models see Chapter 5; for more details about the conditional distributions and some experiments with Gaussian-copula models see Chapter 10.

We also consider mixture distributions of Gaussian-copula models defined as: $\mathbb{P}(\boldsymbol{x}) = \sum_{j=1}^{k} w_j \mathbb{P}(\boldsymbol{x} \mid R^j, F^j)$, where w_j are the probability of each mixture component. As with the Gaussian copula models, the marginal distributions are known in closed-form and the conditional distributions can be shown to be another mixture of the same form (see Chapter 10). While the quantile function for the variables in a mixture distribution do not have a closed-form, it can be computed using standard univariate root finding algorithms such as Brent's method. Mixture models are natural in certain contexts such as text modeling where it is often assumed that the documents exhibit different core themes or topics corresponding to mixture components. More generally, mixture models greatly expand the class of distributions that can be modeled and thus are more widely applicable.

11.6 Conclusion

In this chapter, we introduced the idea of 'what if?' questions and showed how to operationalize such questions into concrete probabilistic queries. We developed a simple query language that enables non-expert users to interact with the system in an intuitive way without having to understand the underlying probabilistic notions that underlie the framework. Finally, we discussed our decision to use Gaussian-copula models and mixtures thereof as the underlying probabilistic models. This chapter builds the foundation for visualizing graphical models as will be described in the next chapter.

Chapter 12

Probabilistic Graphical Model Visualization

12.1 Introduction

Given the probabilistic mechanism, query language and underlying probabilistic model for handling 'what if' queries, we now develop a visualization for query-specific probabilistic models. We target a high-level visualization that emphasizes key statistics of the underlying query-specific model rather than a detail-oriented visualization that allows users to make very specific comparisons. As an analogy, we seek more of a caricature that quickly enables a user to identify interesting trends encoded in the model but may not be a perceptually perfect encoding. Similar to this idea, our proposed visualization emphasizes relative comparison versus absolute comparison between visual objects and considers aesthetics as well as faithfulness to the underlying model. All of this aligns with our goal of helping to answer exploratory 'what if' questions rather than deeper focused data analysis.

12.2 Visually Encoding a Probabilistic Graphical Model

The first step must determine how to encode the probabilistic model into a visual representation via model statistics or parameters. The fundamental patterns encoded in the Gaussian-copula graphical model are the variables and the dependencies between them. Because we are concerned with models that could have hundreds or even thousands of variables, we opt to select the top variables to visualize by sorting them with respect to their median values. Usually, we select between 40 and 100, depending on the dataset to give a high level overview while still retaining many of the variables. We choose 100 for

text data because the complexity is simpler to interpret whereas we choose 30 and 40 for the airport delays and stocks because the acronym meanings are less obvious unless you are an expert in the area. We choose the median as a location statistic rather than the mean because the median can be computed in closed-form, whereas the mean would require a one dimensional numerical integration (see Chapter 10). Note that these median values are the query-specific model medians rather than the data medians, and therefore the selected nodes will likely change for every 'what if' query. The font size of the variable label is scaled linearly between the minimum and maximum median values, where the maximum size is equal to ten times the minimum size; thus, the font size encodes the relative, rather than absolute, median values. As described in the previous section, we choose this relative weighting because we seek to emphasize interesting trends or patterns rather than facilitate absolute comparisons, which might be performed in subsequent, more focused analyses. In addition, our models accommodate continuous distributions that have negative values (e.g. marginal t distributions) and thus there is no general way to define a minimum value.

After selecting variables based on their median values, we select the largest positive graphical model edges (i.e. non-zeros of $\Theta = -R^{-1}$) up to three times the number of selected variables; this limit avoids the common 'hairball' effect if there are a large number of edges by merely emphasizing the most important edges. We specifically choose to display only the positive edges because negative edges are much more difficult to interpret. For example, suppose proteins A, B, C and D are connected like a chain: A-B-C-D. If the edges are all positive, then it is easy to infer that A and D are positively correlated. However, if the edges are negative, then B and D may be negatively correlated with A but C may be positively correlated with A. In addition, to visually represent negative edges, the variables would actually need to be far apart rather than close together, thereby adding very long edges in the visual representation. The size of the edge is scaled linearly between zero and the maximum edge value because the minimum value is known to be zero for all possible

distributions, unlike in the case of node sizes that could have negative minimum values. For mixture distributions, we approximate the mixture edge matrix as $\Theta_{\text{mix}} = -(\sum_{j=1}^{k} w_j R^j)^{-1}$, which can be viewed as approximating a mixture correlation matrix by a weighted average of the component correlation matrices since moments of mixture distributions are a mixture of the component moments.

For mixture models, we also carefully encode the statistics of the mixture distribution into a variable color. We choose the CIELUV color space for color selection and manipulation because they are perceptually uniform as opposed to the RGB or HSV/HSL color space which are not perceptually uniform. More specifically, we use the polar coordinates version of the CIELUV color space called LCH_{uv} which has coordinates corresponding to luminance, chroma (or colorfulness) and hue. First, we assign each mixture component a hue that is equally spaced around the circle of possible hues so that each hue is easy to distinguish. Because hue is most often associated a category, this is natural for mixture models which have components that often correspond to a category or subspace of the distribution. For example, in text modeling, the hue might correspond to various themes or topics represented by the mixture components. All components are assigned the same luminance and chroma value to avoid confusion based on colorfulness or brightness. Ideally, we would like the chroma to be as large as possible but it must be equal for all hues but this can be challenging because the CIELUV space has irregular boundaries between real colors and imaginary colors (such as a dark desaturated yellow). Therefore, we fix the luminance at 74 which gives a wide range of possible hue and chroma values—and in turn we choose a black background so that the high luminance colors will have high contrast with the background. Then, given the component hue values, we select the largest chroma value for which the component colors are visible. Formally, given a luminance of 74 and equally spaced hue values $[h_1, h_2, \cdots, h_k]$ for each of the k mixture components, we select the chroma value as follows:

$$c_1 = \cdots = c_k = c_{\max} = \arg\max_{c} \ \text{IsVisible}(L = 74, C = c, H = [h_1, \cdots, h_k]) \, .$$

Given colors for each mixture component, we now assign colors to variables based on which component would have most likely produced a high values—essentially investigating the composition of the right tail each variable. We define the tail to be values larger than the variable's 99th quantile, i.e. $x_{\text{tail}} = F^{-1}_{\text{mix}}(0.99)$. The probability mass contributed by each component to the tail is given by $\alpha_j = w_j(1 - F_j(x_{\text{tail}}))$. Note that α_j incorporates information both from the mixture weights w_j and from the specific component distribution through $F_j(\cdot)$. We assign a variable hue to the component hue whose contribution α_j is largest $h_{\text{var}} = h_{j^*}$ s.t. $j^* = \arg\max_j \alpha_j$, which means that high values are most likely to come from the j^*-th component. This provides the user with the information of the dominant component exhibited in the high values of that particular variable. However, sometimes no component is truly dominant and so we adjust the variable's chroma value based on how dominant the component is compared to the others. To encode a component's amount of dominance, we first find the second most dominant component: $\tilde{j} = \arg\max_{j \neq j^*} \alpha_j$. We then compute the information theoretical notion of binary entropy of the tail probability of being in the dominant component, i.e. α_{j^*} in comparison to the probability of being in the second most dominant component, i.e. $\alpha_{\tilde{j}}$ assuming that tail values come from one of these two components. Letting $p = \alpha_{j^*}/(\alpha_{j^*} + \alpha_{\tilde{j}})$, the binary entropy can be defined as: $E_{\text{var}} = -p\log_2(p) - (1-p)\log_2(1-p)$. This entropy expresses the uncertainty about the selected component and corresponding hue. In words, if high values of a variable have a very high probability of coming from the dominant component compared to the second most dominant component, then entropy will be close to zero whereas if high values are equally likely to come from the dominant and second most dominant, then the entropy will be close to one. Thus, we encode this entropy into the variable chroma so that variables with a strong dominant component will be colorful whereas variables that do not have any dominant component will appear gray: $c_{\text{var}} = c_{\max}(1 - E_{\text{var}})$.

Overall, this color scheme provides the user with a quick overview of the mixture

structure via the variable hues but avoids wrong assumptions if there is significant uncertainty about the hue selection—i.e. signifying the uncertainty by the lack of chroma or colorfulness. An alternative to this color scheme would have been to merely take a convex combination of the component colors based on the α_j weights. However, this could introduce many new hues and could be confusing to the user because sorting hues is confusing and difficult [Borland and Taylor, 2007]. Thus, we decided to stick with a fixed set of hues and vary the chroma values. Furthermore, reducing the chroma based on uncertainty implicitly moves the color towards all the other colors in the LUV color space thereby helping the user recognize that gray variables are equally near all hues and thus the component is uncertain.

12.3 Layout Aesthetics and Intuition

Now that we have selected the variables, relative font sizes, edges and colors for a particular 'what if' query model, we design a layout of the variable labels that combines the intuitiveness of force-directed layouts with the beauty and readability of word clouds, which pack many words into valuable screen space while ensuring words do not overlap via pixel-level collision detection. In particular, we would like related variables (based on the edges in the graphical model) to be placed near each other while keeping the layout very compact yet readable similar to word clouds such as Wordle (`wordle.net`). On the one hand, beautiful word clouds such as Wordle (`wordle.net`) focus on pixel-level collision detection to enable a compact but readable layout of labels. This space-filling and compact aesthetic criteria is related to the famous NP-complete cutting and packing problem in operations research [Bennell and Oliveira, 2008] and thus is intractable in general to compute exactly. On the other hand, placing related variable labels near each other is akin to the graph layout problem. Noack [2009] proposes a force-directed graph layout based on minimizing the potential energy of a representative physical system between nodes. An interesting extension of the force-directed layout is to add constraints on the nodes to allow

incorporation of domain knowledge [Dwyer, 2009]. Multidimensional scaling (MDS) is yet another graph layout solution in which the "ideal" distance between nodes is estimated and then nodes are placed to minimize the difference between the "ideal" distances and the 2D Euclidean distance [Kamada and Kawai, 1989]. However, both all of these graph layout problems are highly non-convex optimization problems and thus intractable to solve exactly. Given the computational difficulty of both layout problems, therefore, we propose to approximately optimize both simultaneously via adaptive simulated annealing starting from careful initialization which will be described in the following sections.

12.4 Layout Computation

12.4.1 Graph Layout Optimization Function

For graph layout, we use a general force-directed optimization function from the graph layout community defined by [Noack, 2009] called the attraction-repulsion energy model. Let $\boldsymbol{p}_s \in \mathbb{R}^2$ denote the x and y coordinates of the s-th word. Given the node weights $w_s \in \mathbb{R} \forall s$ and edge weights $w_{st} \in \mathbb{R} \forall s, t$, [Noack, 2009] defines the attraction-repulsion energy model as:

$$f(\boldsymbol{p}_{1,\ldots,p}, w_{st}) = \sum_{s,t:s \neq t} \left(-w_s w_t \frac{\|\boldsymbol{p}_s - \boldsymbol{p}_t\|_2^{r+1}}{r+1} + w_{st} \frac{\|\boldsymbol{p}_s - \boldsymbol{p}_t\|_2^{a+1}}{a+1} \right) \tag{12.1}$$

where $\frac{\|\boldsymbol{p}_s - \boldsymbol{p}_t\|_2^{-1+1}}{-1+1}$ must be read as $\log \|\boldsymbol{p}_s - \boldsymbol{p}_t\|_2$ (since y^{-1} is the derivative of $\log(y)$). Note that the gradient of this energy function gives the "forces" acting on each word and thus provides good intuition on what is happening for each node. We choose $a = 1$ and $r = -1$ because it strikes a compromise between compactness and cluster revealing force models—see [Noack, 2009] for more information. In addition, we added a gravity term $w_g \frac{\|\boldsymbol{p}_s\|_2^{g+1}}{g+1}, \forall s$ setting $w_g = 0.1$ and $g = 2$ to help keep the nodes near the center of the visualization. Finally, we set node weights all to be one and the edge weights to be proportional to the selected graphical model edge weights. While we select this particular optimization function,

the code could be easily adapted to implement other optimization functions such as the MDS-like function in [Kamada and Kawai, 1989] or even non-smooth functions that penalize the number of overlapping edges as done in [Lee et al., 2006] could be used.

12.4.2 Pixel-Level Feasibility Condition

While other word cloud visualizations ensure non-overlap by using bounding boxes as in [Barth et al., 2014b], we designed our visualization to enjoy the beauty and compactness of pixel-level precision as in word clouds like Wordle. To this end, the feasible set of label positions is constrained by the highly irregular set of positions where no two labels overlap. The interesting property of this constraint set is that testing for feasibility is relatively straightforward: merely check whether any label overlaps any other label using a pixel-level rendering of the labels. However, describing this feasibility set in mathematical notation is extremely difficult if not impossible given the numerous possible font families, characters and words. Despite the incredible complexity of this constraint set, simulated annealing can easily handle this problem because each step in the algorithm can be accepted or rejected depending on whether it satisfies the non-overlap constraint; no mathematical form of the constraint set is necessary for simulated annealing. Thus, this non-overlap constraint can be seamlessly integrated into the optimization function defined in Equation 12.1 above. One challenge for simulated annealing, however, is that it must start at an initial feasible solution—a problem we overcome as discussed in later sections.

12.4.3 Layout via Simulated Annealing

Because both the space-filling aesthetic and the graph layout problem are computationally difficult even by themselves, we decided to use simulated annealing [Kirkpatrick et al., 1983] to handle both problems in a unified way. Simulated annealing mimics the behavior of cooling atoms from a liquid to a solid. For example, very slowly

cooling quartz will create a quartz crystal whereas quickly cooling quartz will produce quartz glass. Simulated annealing is a principled way to optimize any function—even highly non-convex or combinatorial—using ideas from random sampling theory. Essentially, as in physical systems, simulated annealing takes random steps in the domain such that steps that improve the optimization value are always taken whereas steps that hurt the optimization value are only taken with a particular probability. Simulated annealing is simple to implement and only requires the evaluation of the objective function and a feasibility check at the current iterate. Simulated annealing has been used separately both for the cutting stock problem [Lai and Chan, 1997] and the graph layout problem [Lee et al., 2006]. In addition, simulated annealing could easily incorporate domain constraints similar to [Dwyer, 2009] because it would just need to check feasibility at each iteration. Therefore, we choose simulated annealing as a principled yet general algorithm to approximate multiple difficult problems.

For simulated annealing, several parameters must be chosen including proposal distribution for the next iterate (i.e. the step distribution) and cooling schedule. We choose to sequentially propose a new position for each label while fixing all other label positions. We implement an adaptive Gaussian proposal distribution where the standard deviation is scaled up or down by a factor of 1.2 so that the acceptance rate is close to 23.4%, which is the theoretical optimal for the different but related Metropolis-Hastings algorithm [Roberts et al., 1997]. This acceptance rate helps the algorithm to explore the space appropriately (i.e. many rejections) but still accept useful steps (i.e. some accepts). Note that a rejection of a step can be caused either by a higher optimization value or by a violation of the non-overlap constraint. Throughout the iterations, we track an exponential moving average at an exponential rate of 0.9 and increase or decrease sigma whenever the average acceptance rate goes above or below a 10% tolerance level around 23.4%. For the temperature schedule, we manually adjusted the schedule parameters to trade-off between

accuracy and speed, and we will give our specific values in the next section. Though we did not find the schedule parameters to be particularly sensitive, an in-depth experimental analysis may be a useful future work.

12.4.4 Visualization Algorithm

We now describe the visualization algorithm that seeks to optimize the graph layout function subject to the no overlap constraint. We walk through several important warm-start phases of the algorithm to initialize reasonably. A visual summary of the phases can be seen in the supplementary material.

(Phase 1) We randomly initialize the node positions and render the label glyphs to extract the pixel locations for implementing the pixel-level non-overlap constraint. Then, we run simulated annealing without any constraints. This places the labels approximately in the right position according to the optimization function but allows label overlap. For this phase, we set the cooling schedule to $\{1, 0.9, 0.9^2, \cdots, 0.9^{99}\}$ with five iterations per temperature.

(Phase 2) Then, we scale the font size so that a target number of constraints denoted m are violated. In our case, we choose the number of violated constraints to be equal to the number of displayed labels $m = p$. We compare $m = \{p/2, p, 2p\}$ in an experiment and find that $m = p$ provides a good tradeoff between optimization function value and compactness of visualization but we place the figure in the supplementary materials because of space constraints. This font scaling phase is critical because it automatically selects a reasonable font size without needing to know the underlying optimization function. For example, one optimization function may fit all the nodes in a 100×100 coordinate window while another optimization function may fit all the nodes in a 1×1 coordinate window, yet the relative scale of the optimization function seems uninformative.

(Phase 3) Once the scaling of the fonts has been fixed in phase 2, we need to project

the current solution onto the feasible set—i.e. the set of positions such that no constraint is violated. In this phase, we use what we call *reverse* simulated annealing. Essentially, this works similar to simulated annealing but instead of slowly cooling the temperature, we slowly heat up the temperature until the current word is placed in a feasible position. This phase has some similarity to the placement algorithm of Wordle, which starts at an initial position and slowly attempts to place a word in an increasing spiral until a feasible placement is found. However, this phase is critically different from Wordle's algorithm because the attempted positions are chosen based on the graph layout optimization function. Thus, the reverse simulated annealing seeks to place words in a way that is beneficial to the optimization function. We work greedily from the largest word to the smallest word and attempt to place each word in a position that does not overlap with any previous placed words. We compare our reverse simulated annealing algorithm to Wordle's spiral algorithm and show that our algorithm performs better both quantitatively in terms of objective function and qualitatively but again place the figure in the supplementary materials because of space constraints.

(Phase 4 and 5) In the last two phases, we run constrained simulated annealing so that the words adjust positions based on the optimization function but they are not allowed to overlap. These last two phases can be thought of as refinement phases. In phase 4, we keep the effect of gravity at 0.1 but in phase 5, we increase the effect of gravity to 1 so that all words are pulled towards the center and the final visualization is even more compact.

We implemented several details-on-demand features for the user. First, a user can hover over any of the labels and it will highlight connected labels while making unconnected labels more transparent (see Figure 12.5 or the supplementary materials for examples). This highlights the underlying structure and allows a user to quickly focus on different areas of the visualization. Second, we show the nodes name and actual median value in the lower left hand corner. This enables at least some absolute value details if the user desires some

exact values.

12.4.5 Comparison to Word Clouds

In Figure 12.1, we display examples from Wordle, a semantic word cloud based on force-directed layout[1] evaluated in [Barth et al., 2014b], and our visualization for the text dataset of research paper titles described later. The beauty and compactness of Wordle is largely due to its use of pixel-level collision detection so that for example "gene" can rest in the top part of the word "molecular". Yet, words in Wordle are place randomly and so words that are semantically related like "molecular" and "biology" or "gene" and "expression" can be in completely different parts of the visualization. The semantic word clouds on the other hand preserve word relationships as evidence by "gene" and "expression" being near each other and they cluster the words in a semantically meaningful way. Yet the semantic word clouds use rectangle bounding boxes and thus the compactness and beauty of Wordle is not maintained. Our visualization retains the strengths of both visualizations by placing related words near each other and coloring them in a semantic way, while maintaining a the very compact and space-filling beauty of Wordle. In addition, neither visualization can handle dynamic 'what if' queries because they do not have any underlying model and neither visualization can handle non-text datasets as we will show next.

12.5 Qualitative Results

12.5.1 Datasets and Implementation

We select the three diverse datasets of research paper titles, airport delay times and daily stock returns more fully described in Chapter 10 to show the wide range of applicability of our 'what if' model and visualization. For the research paper titles dataset, we used

[1]We used the online word cloud generator at `http://wordcloud.cs.arizona.edu/index.html`.

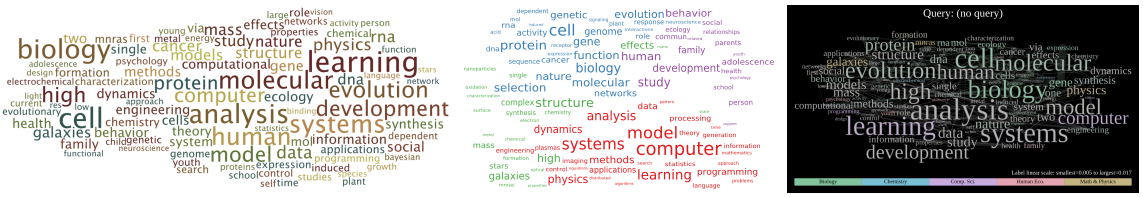


Figure 12.1: (Left to right) Wordle [Feinberg, 2010], force-directed semantic layout [Barth et al., 2014b], and our P3 visualization. These example visualizations demonstrate that while Wordle is much more compact and visually appealing and the semantic layout algorithms show informative positions and colors, P3 retains the strengths of both—while also providing dynamic 'what if' interactions and modeling non-text data which are both impossible with the other two. Furthermore, the gray colors of P3 shows that many words such as "analysis", "data" or "model" are not dominated by any particular topic even though the semantic word cloud assigns an exact color. See subsection 12.5.1 for dataset description.

preliminary versions of the P3 visualization to quickly discover that many stopwords or phrases should be removed, such as "supplementary material", "international conference on machine learning" and names of co-authors—although some author names remain in the dataset, as can be seen in some of the visualizations. At a deeper level, we actually discovered some erroneous "titles" when preparing the figures for this paper. When querying on the word "human", we were surprised that "mathematics" and "astronomy" were also highly occurring as seen in Figure 12.2 as compared to Figure 11.1 (left). Upon further investigation, we found that the dataset contained 13 titles which were merely department name listings; we removed these erroneous titles during further analysis. This example illustrates that our 'what if' model and visualization can be useful in providing insight into data errors as well as structure.

Similar to Chapter 10, we selected the gamma and t distributions for the airport delays and daily stock returns respectively. However, unlike in Chapter 10, we decided to use negative binomial instead of the Poisson for the word counts since it is a more flexible model. These choices emphasize that our 'what if' model is significantly more general than the multivariate Gaussian.

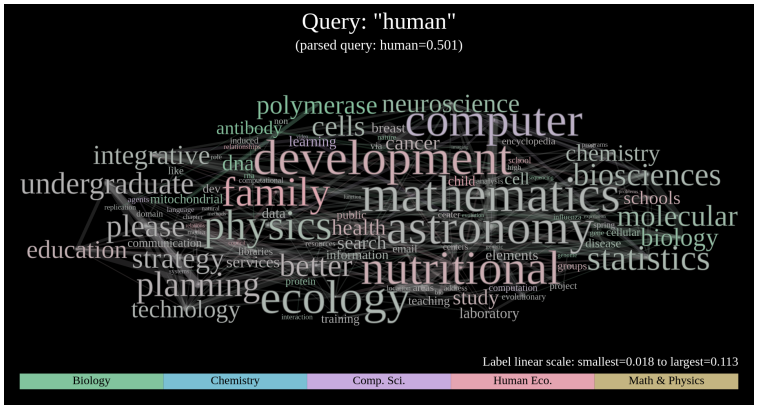


Figure 12.2: Visualizing 'what if' queries can be helpful in noticing data errors or inconsistencies that might otherwise be difficult to notice. While preparing the figures for this paper, we noticed the following unexpected high occurrence of the words "mathematics" and "astronomy" when using the query "human". Upon further investigation, we found that 13 titles that were merely department name listings.

We implemented a prototype of the 'what if' model in MATLAB, which includes model estimation, conditional query parsing and conditional model computation. Note that our framework is agnostic to the model estimation algorithm, but we provide estimation details here for transparency and completeness. We estimated the univariate marginal distributions using the built-in MATLAB fitting functions. We then used the marginal CDFs and inverse standard normal CDF to project the data into the Gaussian-copula space. Finally, we used the QUIC software [Hsieh et al., 2014] to estimate a Gaussian graphical model using ℓ_1-regularized maximum likelihood estimation.

We estimated models using a logarithmic decrease of the regularization parameter λ and choose λ based on Bayesian Information Criteria (BIC), stopping when the training time exceeded ten seconds. For the mixture models, we simply used observation metadata to split the observations into categories corresponding to each mixture component and fit a Gaussian-copula model for each category. For text, this was the high-level category of the professor's research area. For delay times, we treated the seasons of the year as categories. For the

daily stock returns data, there did not seem to be any natural way to cluster the days.Other datasets may have natural observation categories or could be clustered in an application-specific manner. However, we do not explore clustering because model estimation is not the focus of this paper. We implemented the visualization layout component as D3.js JavaScript plugin so that it can be used independently of this framework.

12.5.2 Qualitative Analysis of Visualizations

We present example 'what if' visualizations for the natural science text, airport delay times and stock returns data in Figure 12.3, Figure 12.4 and Figure 12.5 respectively. Larger figures and more examples can be found in the supplementary PDF; live examples in the browser with the corresponding JavaScript visualization code can also be found in the supplementary material. We demonstrate the generality of the queries by giving examples of single word, multiple word and negative queries. For example, the query "bayesian -nonparametric" in Figure 12.3 considers the scenario when "bayesian" has a high chance of occurring while "nonparametric" has a low chance of occurring. Surprisingly, the word word "longitudinal" shows up large and in green, corresponding to biology even though "bayesian" by itself (as seen in Figure 12.3) is focused in the math and physics category (which includes statistics). Upon further investigation, we found a series of papers by one author in biology regarding longitudinal data using Bayesian methods. This example illustrates the insights that this type of 'what if' visualization can produce. As expected, the airport delay visualizations in Figure 12.4 show that most long delays occur during winter as evidenced by the prevalence of the yellow color and many are due to Chicago airport delays at ORD and MDW. Finally, the stock returns visualizations in Figure 12.5 demonstrate that our proposed 'what if' model can even handle real-valued data with heavy tails. The visualizations show that if one stock in a sector performs well, other stocks in that sector also seem to perform well. In addition, the bottom of

Figure 12.5 shows that there appears to be strong clusters of financial stocks that depend on each other. More details and observations can be found in the figure captions. These diverse examples show the broad applicability of our 'what if' modeling framework for different types of data and applications.

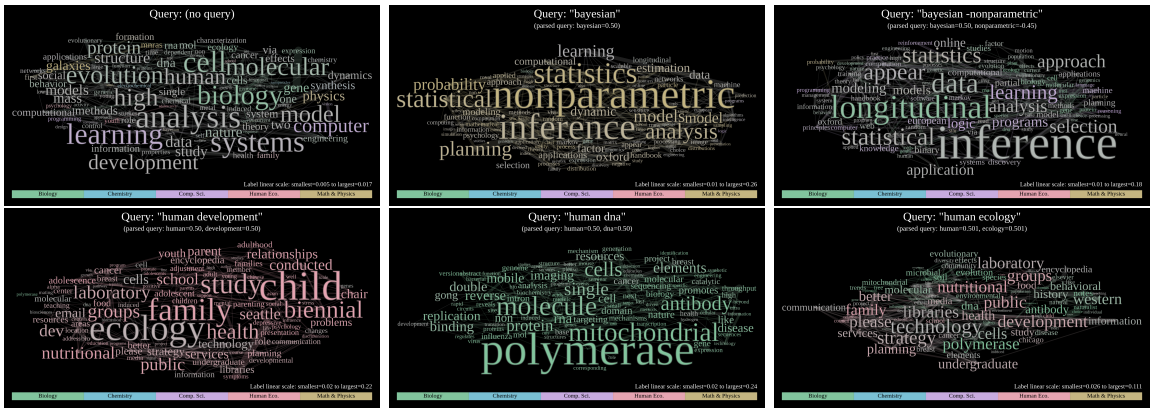


Figure 12.3: These 'what if' visualizations demonstrate how the query intuitively manipulates the underlying probabilistic model and displays related variables (in this case words) in an interpretable fashion. Some words only occur in one subject, such as "electrochemical" in chemistry; these retain the same color in all visualizations. Other words are related to multiple subjects; for example, "model" is used in many subjects and thus sometimes appears gray (top left) and sometimes yellow with query "bayesian" (top middle). Note that queries can be multiple words, affording very different viewpoints such as the bottom three visualizations which show the diversity of the word "human" usage across domains. Queries can also include 'negative' queries with the minus sign such as "bayesian -nonparametric"; these condition on a high chance of "bayesian" occurring but a low chance of "nonparametric" occurring.

12.6 Related Work

12.6.1 Graphical Models Related Work

While we choose Gaussian-copula graphical models because they offer closed-form solutions to the marginal and conditional distributions, other graphical models have been

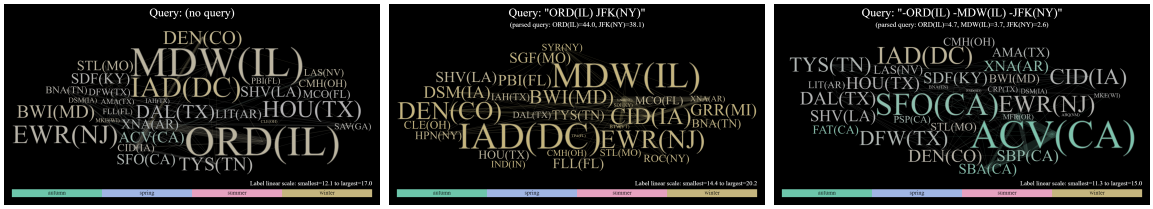


Figure 12.4: These visualizations show that long airport delay times often occur during the winter and possibly autumn months likely due to weather delays. The no query visualization (left) readily shows that the Chicago airports (MDW and ORD) have long delays in general. The query "ORD(IL) JFK(NY)" (middle) conditions on the fact that Chicago and New York have long delays; the yellow color of many variables suggests that other delays are likely in the winter. The negative query of "-ORD(IL) -MDW(IL) -JFK(NY)" means that neither the Chicago or New York airports have long delays and thus there is no cold weather delays at least in the midwest and northeast; however, distant California airports, namely ACV and SFO, may have long delays.

Figure 12.5: The query visualization of "msft amd" (left) demonstrates that if technology companies are performing well, many other technology companies also perform well (e.g. nvda xlnx). When querying "++hig", for Hartford Financial Services, other financial companies do well, and the visualization shows clusters of financial stocks centered around Lincoln National Corporation (middle) and KeyBank (right).

developed in the literature. Beyond Gaussian graphical models, there are binary [Banerjee et al., 2008] and discrete [Jalali et al., 2010] graphical models. Yang et al. [2015] derived theoretical guarantees for learning graphical models for graphical models whose conditional distributions are in the exponential family—which includes the Poisson, exponential and Gaussian univariate distributions. For discrete count data, see Chapters 3,4,7 and 8. In general, these graphical models, even the binary graphical model, do not have closed-form marginal distributions and thus would be difficult to use for 'what if' queries because a

median or mean would need to be estimated via sampling or other method.

Principled 2D visualizations of graphical models is limited. Most work in graphical models simply uses standard graph drawing algorithms. For the Gaussian graphical model, one visualization could be the biplot [Greenacre, 2010] that positions variables based on the top two singular vectors of the covariance matrix. The biplot, however, does not consider any aesthetics and thus, especially when the number of variables is larger than about 20, the plot becomes difficult to read. In addition, the biplot assumes a multivariate Gaussian distribution whereas we assume the more general class of Gaussian-copula models that could non-Gaussian marginals such as a gamma distribution. Another option is to directly visualize the correlation matrix as a heatmap of correlations, possibly color-coded. Both of these visualizations could be useful in certain contexts, however, neither one explicitly provides a way to dynamically interact with the visualization.

12.6.2 Word Clouds Related Work

The most well-known approach to generating word clouds is *Wordle*, developed by Feinberg and colleagues [Viegas et al., 2009]. Wordle generates aesthetically pleasing clouds of words following a heuristic layout procedure [Feinberg, 2010]. Since its introduction, Wordle has enjoyed great popularity, likely due to its favorable aesthetic properties and ease of use. A downside of the Wordle algorithm is that it has no clear probabilistic semantics. Words are sized in proportion to their frequencies and are then placed on the canvas somewhat arbitrarily; *the relative positions of words are meaningless*. Treating tags and words as independent renders word clouds difficult to navigate [Gwizdka and Bakelaar, 2009]. Nonetheless, despite its simplicity, researchers have found Wordles useful for preliminary analyses [McNaught and Lam, 2010].

Similar to our visualization approach, Cui et al. [Cui et al., 2010] introduced an approach that reflects semantic relationships between words in their placement. Their

approach also explicitly conveys temporal trends, thus emphasizing word trends over time. Further, they proposed informing the initial layout of words by semantic coherence. Specifically, this is formalized as the cosine similarity between (localized) word co-occurrence count vectors. The authors calculate dissimilarities between words (one minus the cosine similarities) and then project words into two-dimensions via Multi-Dimensional Scaling (MDS). This represents a heuristic approach to semantically group similar terms. Building on this work, Wu and colleagues introduced a method that improved the aesthetics of 'semantics preserving' word clouds using *seam carving* [Wu et al., 2011], a content-aware image resizing method. Barth et al. further extended this line of work by formalizing the task of generating word clouds in which word placements reflect semantic similarity in terms of the underlying geometric constraints [Barth et al., 2014a]. They refer to this problem as Contact Representation of Word Networks (CROWN). The task is to construct a layout in which two bounding boxes touch if there is an edge between the corresponding nodes. This is NP-hard in general. The authors thus propose several efficient approximations, and demonstrate empirically that these work well. In follow-up work [Barth et al., 2014b], Barth and colleagues extend their evaluation, introducing six quantitative metrics with which to evaluate generated word clouds, the primary of which they term *realized adjacencies*, which sums the similarities of all touching word nodes. Elsewhere, Wang et al. [Wang et al., 2014] recently introduced another approach to generating word clouds that attempts to preserve some notion of semantics in word placement, specifically for the application of user-generated restaurant reviews. Their approach generates and exploits dependency parses of reviews to account for the grammatical structure of text during word cloud generation (e.g., they retain only nouns, verbs, and adjectives in reviews). In other related work, Gambette and Veeronis proposed a *word tree cloud* [Gambette and Véronis, 2010] in which pairwise similarities between words are used to induce a tree. Note that all these semantic word cloud methods are restricted to text data and only provide a static layout of words while our 'what if' framework

provides dynamic visualizations for text *and* non-text data like delay times and stock returns. Thus, our approach is both more dynamic and more general than these semantic word cloud approaches.

12.6.3 Color Related Work

We choose the CIELUV color space because Sharma and Rodríguez-Pardo [2012] gives some evidence that CIELUV is better for emissive displays than the CIELAB color space, even though Berns [2001] states that the CIE commission wanted to make a distinction but did not feel there was evidence at the time. Zeileis and Hornik [2006] discusses color palettes for statistical graphics and was the inspiration for our choice of using fixed chroma and luminance for the component colors. Ihaka [2003] describe the history of color spaces and suggested that when size is a critical visual component, luminance should be fixed. Harrower and Brewer [2011] provide an online tool for the famous ColorBrewer quantitative and qualitative palettes, which enable careful color distinction even amid different conditions. While we could have used ColorBrewer palettes for our visualization, we chose to stay in the CIELUV color space so that we could correctly show gray colors that would appear appropriately between all the other colors.

12.7 Limitations

First, we note that our 'what if' models are descriptive rather than predictive. They can show historical trends or patterns but are not meant to predict future values. For example, they are not meant to predict stock prices or the delays at airports tomorrow. However, they may provide initial insights into a dataset that may help design predictive models. In addition, we only consider the particular class of Gaussian-copula undirected graphical models, which might not be a good fit for a certain application. While Gaussian-copula models are fairly general and can include discrete and continuous data types, other

models such as time series models or directed Bayesian graphical models may be a better fit. Applying our 'what if' framework and visualization to these types of models would be an interesting area of future work. On the visualization side, our current implementation is somewhat slow for more than about 100 or 200 variables. On a computer with an Intel Xeon(R) CPU 3.60GHz $\times$ 8 processor, the computation takes between 15 and 20 seconds for 100 variables and between 60 and 80 seconds for 200 variables—although the user can see the begin interacting immediately. Significant improvements in the absolute speed of our visualization algorithm could likely be made if the collision detection was more advanced using bounding boxes or similar idea.

12.8 Conclusion

Towards our goal of visualizing dynamic 'what if' queries for exploratory analysis, we first define the probabilistic mechanism to be conditional probability, which is similar to filtering in the probability space. We develop a query language that does not require the user to know the appropriate conditioning values but rather merely that they want to condition on high or low values of particular variable; this enables a simple 'what if' interface for the user. We choose the Gaussian-copula graphical model as the underlying probabilistic model because it has closed-form solutions to the required distributions and statistics. After developing this underlying model interaction for 'what if' queries, we then develop a probabilistic model visualization that combines the intuitiveness of force-directed layouts and the beauty of word clouds. Furthermore, we carefully define a color model that intuitively shows properties of the underlying mixture distribution. Finally, we demonstrate that our model and visualization is broadly applicable to both text and non-text datasets by illustrating our framework on text data, non-negative airport delay data, and heavy-tailed stock return data. While this framework for 'what if' visualization is not appropriate for in-depth analysis, it can be a useful exploratory visualization tool. We hope the fundamental

idea of tightly integrating probabilistic models and concepts with visualization could open up the door for innovative visualizations based on a probabilistic foundation.

Chapter 13

Concluding Thoughts

13.1 Overview

Part I explored the challenge of graphical models to allow *positive* dependencies even for the Poisson or exponential case. The original Poisson or exponential graphical models [Besag, 1974, Yang et al., 2015] only allowed negative dependencies—which is required to ensure the distribution is normalizable. [Yang et al., 2013] introduced Poisson variants by truncating the distribution—via hard truncation in the case of TPGM or soft truncation in the case of SPGM—or altering the Poisson to essentially be a non-negative and discrete Gaussian (QPGM). These modifications unfortunately either require unintuitive truncation hyperparameters (TPGM/SPGM) or have Gaussian-esque thin tails (QPGM). [Allen and Liu, 2012, 2013] ignore the joint consistency and normalization of the model and suggest that the parameters can still be useful even though no joint distribution exists. Given these shortcomings of previous models, we propose the LPMRF and SQR graphical models that both allow positive dependencies. Overall, we suggest using the SQR graphical model for the general situation because the solution is more elegant but the LPMRF graphical model actually provides a replacement for the multinomial distribution, which is critical for the development of LPMRF topic models in the following part and may be more generally useful as a replacement for the multinomial. We conclude this section with an extensive comparison of Poisson graphical models with multiple types of multivariate Poisson generalizations including copula-based models.

In Part II, we explore the combination of count-valued graphical models within

topic models. In standard topic models such as LDA, words are drawn individually from a categorical distribution assuming the topic assignments are known. However, multiple words need to be drawn together in order to incorporate joint graphical model distributions, which directly model dependencies between words. Thus, we reformulate topic models in two different ways so that multiple words can be drawn jointly from a graphical model. We then demonstrate two instantiations of these generalizations in the novel APM model and LPMRF topic model.

Building on previous models in Part III, we develop a generalization of the SQR graphical model for the case of k-wise rather than merely pairwise interactions. We show that this class of models is normalizable for reasonable constraints on the parameters and even give a tractable algorithm for estimating the parameters. In this part, we also consider the well-known Gaussian-copula model because it can be viewed as a semi-parametric graphical model [Liu et al., 2012]. We derive the closed-form solutions to the conditional Gaussian-copula model; though this derivation is not particularly surprising, we could not find a full derivation in the literature. In significant contrast to the other graphical models we describe, these Gaussian-copula graphical models actually have closed-form solutions including the normalization constant and thus inference, mean estimation and sampling are significantly easier and faster. Using this derivation, we then show the efficacy of these closed-form solutions in missing value imputation experiments.

Finally, in Part IV, we strive to make graphical models more accessible to non-experts. In particular, we define a simple query-based interactive interface for the user that does not require explicit knowledge of variable values nor any knowledge of the underlying graphical model. To display the results to the user, we carefully design a visualization that displays key information about the underlying model including the top variables, the graphical model edges, and information about the underlying mixture components. The query-based interface connected to our powerful visualization enables a data scientist to quickly explore the key

trends and patterns in the dataset by entering 'what if?' questions as queries. This could be useful in forming initial hypotheses about the data and determining which questions to ask next—a fundamental step in the initial exploratory analysis of new data. Overall, we hope that these novel graphical models and visualization tools enable data scientists and machine learning experts to estimate, apply, and understand their models more effectively and efficiently.

13.2 Future Work

From the experimental comparison results in Chapter 5, the SQR graphical model performs better than any of the other Poisson graphical models. However, the APM and LPMRF topic models described in Part II used the SPGM graphical model and the LPMRF graphical model respectively. Thus, a straightforward area for future work is to use the SQR graphical model as the basis for these topic models. For the admixture model (generalization 1), the SQR graphical model could directly replace the SPGM model. For the LPMRF topic model, a fixed-length SQR model could likely be derived and used instead of the LPMRF model. Thus, the elegant form of the SQR graphical models could be leveraged in the more complex topic model extensions.

Another interesting area of future work considers the interaction or intersection of copula-based models and graphical models. We investigated the Gaussian-copula model which lies at the intersection of these model classes but further connections could be made. For example, what is the form of the copula associated with the SQR graphical models? This copula is guaranteed to exist by Sklar's representation theorem. Knowing the copula form of SQR models reduces to the problem of determining the graphical model marginal distributions, which is similar to estimating the normalization constant and thus is usually computationally difficult. As another question, supposing we could approximate the copula via sampling or other approximation, how close is this implicit SQR copula to the Gaussian

copula? Could sampling from a closed-form Gaussian-copula model give good initial samples for SQR Gibbs sampling? In general, a careful consideration of the similarities and differences between these two paradigms could reveal powerful synergies.

Finally, while this work has focused on novel models and visualizations, real-world applications of this work is an important area of future work. Because we focus on the unsupervised setting in this work, any application will likely be an exploratory analysis of data rather than an explicit supervised task. In addition, because these models are founded on theoretical guarantees even in the high-dimensional setting where $n < p$, strong applications will likely be in the high-dimensional setting where n is relatively small (e.g. 100 or 1000 observations) rather than when n is very large (e.g. millions of observations). Biological applications, especially with next generation sequencing and advanced mass spectrometry, fit these two criteria. They are exploratory in nature because one major biological goal is to understand the dependency structure between genes or proteins as opposed to merely predicting an outcome, and they are in the high-dimensional setting because the number of dimensions is often large (e.g. more than 1000 genes) compared to the small number of samples (e.g. less than 100 patients).

Another possible application is qualitative coding of free-form survey text data or free-form medical notes. Informally, qualitative coding is the process of determining the topics or themes in a text dataset—a task usually manually performed by a human expert. Both the proposed graphical models and the proposed visualization framework may enable a human expert to gain a gist of the data much more quickly than reading the text individually especially when the number of documents is large (e.g. 1000 survey responses). With more work, an direct qualitative coding tool could be built on top of the graphical models and visualization that would enable a user to quickly partition the documents into useful topics—essentially, leveraging the power of the human and the computer interactively. More generally, we hope that the foundational work in more flexible graphical models, topic models,

and interactive visualization will expand the understanding of graphical models and open the door for powerful applications and tools for both experts and non-experts alike.

Appendices

Appendix A

Proofs of SQR Normalization

A.1 Proof of Exponential SQR Normalization

The basic intuition is clear by looking at the asymptotic growth of each term. However, we specifically outline the possibilities:

1. $\eta_2 > 0$: $A(\eta_1, \eta_2) \to \infty$.
2. $\eta_2 < 0$: $A(\eta_1, \eta_2) < \infty$.
3. $\eta_2 = 0$: if $\eta_1 < 0$, then $A(\eta_1, \eta_2) < \infty$, otherwise $A(\eta_1, \eta_2) \to \infty$.

In summary, we need that $\eta_2 < 0$ *or* ($\eta_2 = 0$ *and* $\eta_1 < 0$).

Case 1: $\eta_2 > 0$ Let $\hat{\eta}_2 = \eta_2/2$. First, we seek an exponential lower bound on the partition function. In particular, we want to find an $\bar{z}$ such that for all $z > \bar{z}$, $\exp(\hat{\eta}_2 z) \leq \exp(\eta_1\sqrt{z} + \eta_2 z)$. Taking the log of both sides and solving, we find that the critical points of the above inequality are at 0 and $(-2\frac{\eta_1}{\eta_2})^2$. We take the non-trivial solution of $\bar{z} = (-2\frac{\eta_1}{\eta_2})^2$. Now we need to check if the region to the right of $\bar{z}$ is possible by plugging into the original equation.

Let us try a point $\tilde{z} = a\bar{z}$ where $a > 1$:

$$\exp(\hat{\eta}_2 \tilde{z}) \overset{?}{\leq} \exp(\eta_1 \sqrt{\tilde{z}} + \eta_2 \tilde{z}) \tag{A.1}$$

$$\Rightarrow \hat{\eta}_2 \tilde{z} \overset{?}{\leq} \eta_1 \sqrt{\tilde{z}} + \eta_2 \tilde{z} \tag{A.2}$$

$$\Rightarrow (\eta_2/2 - \eta_2)\tilde{z} \overset{?}{\leq} \eta_1 \sqrt{\tilde{z}} \tag{A.3}$$

$$\Rightarrow -\frac{\eta_2}{2}\left(-2a\frac{\eta_1}{\eta_2}\right)^2 \overset{?}{\leq} \eta_1 \sqrt{\left(-2a\frac{\eta_1}{\eta_2}\right)^2} \tag{A.4}$$

$$\Rightarrow a\frac{-2\eta_1^2}{\eta_2} \overset{?}{\leq} \sqrt{a}\frac{-2\eta_1^2}{\eta_2} \tag{A.5}$$

$$\Rightarrow a \geq \sqrt{a}\,, \tag{A.6}$$

where the last line is because we assumed $a > 1$ and $\eta_2 > 0$. Thus, we can lower bound the log partition function as follows:

$$\begin{aligned} A(\eta_1, \eta_2) &= \int_0^{\bar{z}} \exp(\eta_1 \sqrt{z} + \eta_2 z)\mathrm{d}z \\ &\qquad + \int_{\bar{z}}^{\infty} \exp(\eta_1 \sqrt{z} + \eta_2 z)\mathrm{d}z \\ &\geq \int_0^{\bar{z}} \exp(\eta_1 \sqrt{z} + \eta_2 z)\mathrm{d}z + \underbrace{\int_{\bar{z}}^{\infty} \exp(\hat{\eta}_2 z)\mathrm{d}z}_{\to\infty,\, \text{since } \hat{\eta}_2 > 0} \\ &= \infty\,. \end{aligned}$$

Therefore, if $\eta_2 > 0$, the log partition function diverges and hence the joint distribution is not consistent.

Case 2: $\eta_2 < 0$ Now we will find an exponential upper bound and show that this upper bound converges—and hence the log partition function converges. In a similar manner to case 1, let $\hat{\eta}_2 = \eta_2/2$. We want to find an $\bar{z}$ such that for all $z > \bar{z}$, $\exp(\hat{\eta}_2 z) \geq \exp(\eta_1 \sqrt{z} + \eta_2 z)$—the only difference from case 1 is the direction of the inequality. Thus, using the same

reasoning as case 1, we have that $\bar{z} = (-2\frac{\eta_1}{\eta_2})^2$. Similarly, we need to check if the region to the right of $\bar{z}$ is possible by plugging into the original equation. In a analogous derivation, we arrive at the same equation as Eqn. A.5 except with the inequality is flipped:

$$\Rightarrow a\frac{-2\eta_1^2}{\eta_2} \overset{?}{\geq} \sqrt{a}\frac{-2\eta_1^2}{\eta_2} \tag{A.7}$$

$$\Rightarrow a \geq \sqrt{a}\,, \tag{A.8}$$

where the last step is because we assumed $\eta_2 < 0$ and $a > 1$—note that we do not flip the inequality because $\frac{-2\eta_1^2}{\eta_2}$ is overall a positive number. Thus, this is an upper bound on the interval $[\bar{z}, \infty]$:

$$\begin{aligned}
A(\eta_1, \eta_2) &= \int_0^{\bar{z}} \exp(\eta_1\sqrt{z} + \eta_2 z)\mathrm{d}z \\
&\qquad + \int_{\bar{z}}^{\infty} \exp(\eta_1\sqrt{z} + \eta_2 z)\mathrm{d}z \\
&\leq \int_0^{\bar{z}} \exp(\eta_1\sqrt{z} + \eta_2 z)\mathrm{d}z + \underbrace{\int_{\bar{z}}^{\infty} \exp(\hat{\eta}_2 z)\mathrm{d}z}_{\text{Upper bound}} \\
&\leq \underbrace{\int_0^{\bar{z}} \exp(\eta_1\sqrt{z} + \eta_2 z)\mathrm{d}z}_{<\infty} + \underbrace{\int_0^{\infty} \exp(\hat{\eta}_2 z)\mathrm{d}z}_{\text{Exp. log partition}} \\
&< \infty,
\end{aligned}$$

where the last step is based on the fact that a bounded integral of a finite smooth function is bounded away from ∞ and the second term is merely the log partition function of a standard exponential distribution.

Case 3: $\eta_2 = 0$ This gives the log partition function simply as:

$$A(\eta_1, \eta_2) = \int_0^{\infty} \exp(\eta_1\sqrt{z})\mathrm{d}z,$$

which has the closed form solution:

$$A(\eta_1, \eta_2) = 2\eta_1^{-2}(\eta_1\sqrt{z} - 1)\exp(\eta_1\sqrt{z})\Big|_0^\infty \tag{A.9}$$

$$= \lim_{z\to\infty} 2\eta_1^{-2}(\eta_1\sqrt{z} - 1)\exp(\eta_1\sqrt{z}) - \left(-2\eta_1^{-2}\right) \tag{A.10}$$

$$= 2\eta_1^{-2} + \lim_{z\to\infty} 2\eta_1^{-2}(\eta_1\sqrt{z} - 1)\exp(\eta_1\sqrt{z})\,. \tag{A.11}$$

The convergence critically depends on the limit in Eqn. A.11. This limit diverges to ∞ if $\eta_1 \geq 0$ but converges to 0 if $\eta_1 < 0$. Thus, if $\eta_2 = 0$, then $\eta_1 < 0$ for the log partition function to be finite.

A.2 Proof of Poisson SQR Normalization (Eqn. 4.14)

First, we take an upper bound by absorbing the $\sqrt{z}$ term:

$$A(\eta_1, \eta_2) \leq \sum_{z\in\mathbb{Z}_+} \exp\Big(\underbrace{\eta_1\sqrt{z} + \eta_2 z}_{O(z)} \tag{A.12}$$

$$\underbrace{- \sum_s \log(\Gamma(zv_s + 1))}_{O(z\log z)}\Big) \tag{A.13}$$

$$\leq \sum_{z\in\mathbb{Z}_+} \exp\Big(\eta z - \sum_s \log(\Gamma(zv_s + 1))\Big), \tag{A.14}$$

where $\eta = \eta_2 + |\eta_1|$. We continue the bound as follows:

$$A(\eta_1, \eta_2) \leq \sum_{z\in\mathbb{Z}_+} \exp\Big(\eta z - \max_s \log(\Gamma(zv_s + 1))\Big) \tag{A.15}$$

$$\leq \sum_{z\in\mathbb{Z}_+} \exp\left(\eta z - \log(\Gamma(z/p + 1))\right), \tag{A.16}$$

where Eqn. A.16 comes from the fact that $\arg\max_{v_s} \log(\Gamma(zv_s + 1)) \geq 1/p$ (simple proof by contradiction).

Now let us use the ratio test for convergent series where $a_z = \exp\Big(\eta(\mathbf{v})z - \log(\Gamma(\frac{z}{p}+1))\Big)$:

$$\lim_{z\to\infty}\frac{|a_{z+1}|}{|a_z|} = \exp(\eta(z+1) - \log(\Gamma((z+1)/p+1)) \\ - [\eta z - \log(\Gamma(z/p+1))]) \tag{A.17}$$

$$= \lim_{z\to\infty}\exp\left(\eta + \log\left(\frac{\Gamma(z/p+1))}{\Gamma((z+1)/p+1))}\right)\right) \tag{A.18}$$

$$= \exp(\eta)\lim_{z\to\infty}\frac{\Gamma(z/p+1))}{\Gamma((z/p+1+1/p))}\frac{(z/p+1)^{1/p}}{(z/p+1)^{1/p}} \tag{A.19}$$

$$= \exp(\eta)\lim_{z\to\infty}\frac{1}{(z/p+1)^{1/p}} \\ \times \lim_{z\to\infty}\frac{\Gamma(z/p+1))(z/p+1)^{1/p}}{\Gamma((z/p+1+1/p))} \tag{A.20}$$

$$= \exp(\eta)\lim_{z\to\infty}\frac{1}{(z/p+1)^{1/p}}(1) \tag{A.21}$$

$$= \exp(\eta)(0)(1) = 0 < 1, \tag{A.22}$$

where Eqn. A.20 is by the product of limits rule and Eqn. A.21 is by the well-known asymptotic properties of gamma functions. Therefore, by the ratio test, the radial conditional log partition function is bounded for any $\eta_1 < \infty$ and $\eta_2 < \infty$.

Appendix B

APM Derivations, Algorithm and Visualizations

B.1 Notational Conventions

Matrices are denoted by capital letters (e.g. X, Φ). Column vectors are denoted by lowercase bold face Roman and Greek letters (e.g. $\boldsymbol{x}$, $\boldsymbol{\theta}$). Usually, lower case letters are the columns of their upper case matrix counterparts (e.g. $\boldsymbol{x}_i$ is the ith column vector of X) except for $\boldsymbol{\theta}$ which is distinct from a column of Φ. Subscripts indicate either the column of a matrix (e.g. Φ_s) or a scalar value indexed on a vector (e.g. x_{is}, θ_s). Superscripts indicate an element of a set, which can either be a set of vectors or a set of matrices (e.g. $\boldsymbol{\theta}^j \in \boldsymbol{\theta}^{1\ldots k}$, $\Phi^j \in \Phi^{1\ldots k}$, or $\boldsymbol{\Phi}^s \in \boldsymbol{\Phi}^{1\ldots p}$).

The subscript $\backslash i$ as in $\text{vec}(\boldsymbol{\Phi}^s)_{\backslash i}$ refers to the sub vector when the ith coordinate is made to be zero. This is important when calculating the ℓ_1 regularization because the only the edge parameters are regularized and therefore the node parameters must be ignored.

B.2 Reformulation of negative pseudo log likelihood

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\mathrm{APM}}(\boldsymbol{x}_i \,|\, \boldsymbol{w}_i, \boldsymbol{\theta}^{1\dots k}, \Phi^{1\dots k}) \tag{B.1}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\mathrm{PMRF}}\big(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}^i = \sum_{j=1}^{k} w_j \boldsymbol{\theta}^j, \Phi^i = \sum_{j=1}^{k} w_j \Phi^j\big) \tag{B.2}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \Big[\big(\sum_{j=1}^{k} w_{ij}\boldsymbol{\theta}^j\big)^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \big(\sum_{j=1}^{k} w_{ij}\Phi^j\big)\boldsymbol{x}_i - \sum_{s=1}^{p} \exp\big(\sum_{j=1}^{k} w_{ij}(\theta_s^j + \boldsymbol{x}_i^T \Phi_s^j)\big)\Big] \tag{B.3}$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{s=1}^{p} \Big[\sum_{j=1}^{k} w_{ij} x_{is}(\theta_s^j + \boldsymbol{x}_i^T \Phi_s^j) - \exp\big(\sum_{j=1}^{k} w_{ij}(\theta_s^j + \boldsymbol{x}_i^T \Phi_s^j)\big)\Big] \tag{B.4}$$

B.3 Parameter Settings

A summary of the parameter settings for the models trained can be seen in Table B.1. Experiments were run over all possible combinations of the parameters given and final parameter values determined by evocation score on 50% tuning set. Note that the output edge matrices of APM (i.e. $\Phi^{1\ldots k}$) are not symmetric because the algorithm ignores the symmetry constraint thus yielding an overcomplete representation in which two estimates of the word dependency parameters are computed. These two estimates can be combined in at least 2 ways:

1. OR: Assume the combined estimate is a non-zero if either estimate is non-zero (i.e. take the OR of the estimated non-zero edges). Then, merely average both estimates.
2. AND: Assume the combined estimate is non-zero *only if both* estimates are non-zero (i.e. take the AND of the estimated non-zero edges). Then, merely average the non-zero entries.

Note that if the estimator is actually recovering the true neighborhood (i.e. a variable's non-zero dependencies with other variables), then these definitions are equivalent. However, in practice, we have found that the models are quite different yet both give reasonable results. In general, we observed that AND is easier to interpret and is less likely to overfit the training data than OR. AND also has the intuitive interpretation that two words are directly dependent on one another if and only if they are useful in predicting each other (i.e. they are non-zero coefficients in the node-wise Poisson-like regression problems). This is why we chose to use AND for APM-LowReg and APM-HeldOut. We suggest that in general AND is probably a better post-processing step than OR. However, more fully studying the effects of this post-processing step could be an area of future research.

Table B.1: Table of Parameter Settings for Models

Model	Parameter settings
APM	$k \in \{1, 3, 5, 10, 25\}$ Trace iteration $\in \{1, 2, \cdots, 15\}$ (i.e. different λ values) $\beta \in \{0, 0.01, 1\}$ Post processing of edge set $\in \{AND, OR\}$
APM-LowReg	$k \in \{1, 3, 5, 10, 25\}$ λ chosen to be very small (usually approximately $\frac{\lambda_{\max}}{2^{15}}$) $\beta = 0$ Post processing of edge set $\in \{AND\}$
APM-HeldOut	$k \in \{1, 3, 5, 10, 25\}$ Percentage of held-out documents $\in \{10\%, 20\%\}$ λ chosen by held-out training documents $\beta = \{0, 0.1\}$ Post processing of edge set $\in \{AND\}$
CTM	$k \in \{1, 3, 5, 10, 25\}$ Default parameters except for two different convergence criteria
HDP	Topic Dirichlet hyperparameter $\eta \in \{1, 0.01, 0.0001\}$ Hyperparameter resampling $\in \{yes, no\}$ Scaling for prior if hyperparameter resampling or first concentration parameter $\gamma \in \{100, 10, 1, 0.1\}$
LDA	$k \in \{1, 3, 5, 10, 25, 50\}$ Topic Dirichlet hyperparameter $\beta \in \{1, 0.01, 0.0001\}$ Document Dirichlet hyperparameter $\alpha = \{1, 0.1, 0.01\}$ Optimize hyperparameters $\in \{yes, no\}$
RSM	$k \in \{1, 3, 5, 10, 25, 50\}$ Learning rate $\in \{10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$ Maximum iterations $\in \{10^3, 10^4\}$

B.4 Algorithms

B.4.1 Main Alternating Algorithm for APM

Algorithm 1: Estimate APM parameters using an alternating scheme

Input : Data matrix $X \in \mathbb{Z}_+^{p \times n}$, number of topics k, prior hyperparameter $\beta \geq 0$

Output: Parameters $\boldsymbol{\theta}_\lambda^{1\ldots k}$, $\Phi_\lambda^{1\ldots k}$ and W_λ for different values of λ

```
W ← rand(k, n)
for λ ← ∞, λ_max, λ_max/2, λ_max/4, λ_max/8, ··· do
    while not converged do
        [θ^{1...k}; Φ^{1...k}] ← EstimateComponentPMRFs(W, X, λ, β)
        W ← EstimateAdmixtureWeights(θ^{1...k}, Φ^{1...k}, X)
    end
end
```

B.4.2 Component PMRFs Algorithm

Algorithm 2: Estimates the k node and edge parameters for word index s when W is fixed

Input : Data matrix X, admixture weights matrix W, word index s, sparsity parameter λ

Output: Parameter $\mathbf{\Phi}^s$

$\boldsymbol{Z} \leftarrow \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \end{bmatrix}$; $\quad \Psi^s \leftarrow \mathrm{W}\mathrm{diag}([x_{1s}\, x_{2s}\, \cdots x_{ns}])\boldsymbol{Z}^T$; $\quad \mathbf{\Phi}^s \leftarrow \mathbf{0}$

while *not converged* **do**

  $\forall i,\ \boldsymbol{\gamma}_i \leftarrow \exp(\boldsymbol{z}_i^T \mathbf{\Phi}^s \boldsymbol{w}_i)$; $\quad \boldsymbol{D} \leftarrow \mathbf{0}$; $\quad \boldsymbol{r} \leftarrow \mathbf{0}$; $\quad \epsilon = 0.5$; $\quad \sigma = 10^{-10}$

  $\nabla g(\mathbf{\Phi}^s) \leftarrow -(1/n)(\Psi^s - \boldsymbol{Z}\mathrm{diag}(\boldsymbol{\gamma})\mathrm{W}^T)$

  $\mathcal{F} \leftarrow \{(t,j) : t \neq s \text{ and } (|\nabla_{jt} g(\boldsymbol{\phi})| \geq \lambda \text{ or } \phi_{jt} \neq 0 \text{ or } t = 1)\}$

  while *not converged* **do**

    for $(t,j) \in \mathcal{F}$ **do**

      $a = \sum_{i=1}^n \gamma_i (w_{ij} z_{it})^2$; $\quad b = \nabla_{jt} g(\mathbf{\Phi}^s) + \sum_{i=1}^n \gamma_i w_{ij} z_{it} r_i$; $\quad c = \phi_{jt} + d_{jt}$

      $\mu \leftarrow -c + \mathcal{S}_{\lambda/a}(c - b/a)$; $\quad d_{jt} \leftarrow d_{jt} + \mu$; $\quad \forall i,\ r_i \leftarrow r_i + \mu z_{it} w_{ij}$

    end

  end

  for $\alpha \leftarrow 1, \epsilon^2, \epsilon^3, \cdots$ **do**

    $\hat{\mathbf{\Phi}}^s \leftarrow \mathbf{\Phi}^s + \alpha \boldsymbol{D}$

    $f(\hat{\mathbf{\Phi}}^s) \leftarrow -(1/n)\big(\mathrm{tr}(\Psi^s \hat{\mathbf{\Phi}}^s) - \sum_{i=1}^n \exp(\boldsymbol{z}_i^T \hat{\mathbf{\Phi}}^s \boldsymbol{w}_i)\big) + \lambda \|\mathrm{vec}(\hat{\mathbf{\Phi}}^s)_{\setminus 1}\|_1$

    if $f(\hat{\mathbf{\Phi}}^s) \leq f(\mathbf{\Phi}^s) + \alpha\sigma[\mathrm{tr}(\nabla g(\mathbf{\Phi}^s)^T \boldsymbol{D}) + \|(vec(\mathbf{\Phi}^s) + vec(\boldsymbol{D}))_{\setminus 1}\|_1 - \|vec(\mathbf{\Phi}^s)_{\setminus 1}\|_1]$ **then**

      $\mathbf{\Phi}^s \leftarrow \hat{\mathbf{\Phi}}^s$; $\quad$ break

    end

  end

end

B.5 Top 50 Word Pairs for Best LDA and APM Models

Table B.2: Top 50 Word Pairs for LDA (Left) and APM (Right)

Human Score	Model Rank	Word Pair
100	17	run.v ↔ car.n
82	6	teach.v ↔ school.n
69	4	school.n ↔ class.n
63	49	van.n ↔ car.n
51	24	hour.n ↔ day.n
50	14	teach.v ↔ student.n
44	27	house.n ↔ government.n
44	22	week.n ↔ day.n
38	26	university.n ↔ institution.n
38	10	state.n ↔ government.n
38	1	woman.n ↔ man.n
38	43	give.v ↔ church.n
38	16	wife.n ↔ man.n
38	7	engine.n ↔ car.n
35	8	publish.v ↔ book.n
32	46	west.n ↔ state.n
32	12	year.n ↔ day.n
25	45	member.n ↔ give.v
25	25	dog.n ↔ animal.n
25	13	seat.n ↔ car.n
19	32	west.n ↔ area.n
19	21	fish.n ↔ animal.n
19	20	white.a ↔ black.a
16	50	journal.n ↔ book.n
16	19	paper.n ↔ book.n
16	34	tree.n ↔ plant.n
13	35	year.n ↔ week.n
13	29	ride.v ↔ horse.n
13	3	train.n ↔ car.n
7	39	institution.n ↔ date.n
7	33	people.n ↔ family.n
7	30	teacher.n ↔ teach.v
7	9	religious.a ↔ church.n
6	41	university.n ↔ date.n
6	44	plant.n ↔ bird.n
0	48	room.n ↔ house.n
0	47	show.v ↔ first.a
0	42	record.n ↔ play.v
0	40	high.a ↔ area.n
0	38	urban.a ↔ area.n
0	37	high.a ↔ first.a
0	36	member.n ↔ authority.n
0	31	subject.n ↔ old.a
0	28	title.n ↔ subject.n
0	23	text.n ↔ language.n
0	18	give.v ↔ get.v
0	15	tell.v ↔ get.v
0	11	car.n ↔ bus.n
0	5	drive.v ↔ car.n
0	2	woman.n ↔ wife.n

Human Score	Model Rank	Word Pair
100	22	telephone.n ↔ call.n
97	10	husband.n ↔ wife.n
82	48	residential.a ↔ home.n
76	23	politics.n ↔ political.a
75	6	steel.n ↔ iron.n
75	36	job.n ↔ employment.n
75	37	room.n ↔ bedroom.n
72	11	aunt.n ↔ uncle.n
72	27	printer.n ↔ print.v
60	2	love.v ↔ love.n
57	7	question.n ↔ answer.n
57	42	prison.n ↔ cell.n
51	49	mother.n ↔ baby.n
50	28	sun.n ↔ earth.n
50	4	west.n ↔ east.n
44	31	weekend.n ↔ sunday.n
41	18	wine.n ↔ drink.v
38	5	south.n ↔ north.n
38	38	morning.n ↔ afternoon.n
38	43	engine.n ↔ car.n
35	32	publish.v ↔ book.n
35	15	green.n ↔ green.a
35	30	salt.n ↔ rice.n
35	34	copy.v ↔ copy.n
33	19	troop.n ↔ force.n
28	12	tea.n ↔ coffee.n
28	50	win.v ↔ prize.n
25	13	operational.a ↔ aircraft.n
19	17	smoke.n ↔ fire.n
19	35	white.a ↔ black.a
13	1	smoke.v ↔ cigarette.n
13	3	eat.v ↔ food.n
13	8	boil.v ↔ potato.n
13	21	ride.v ↔ horse.n
10	44	fall.v ↔ fall.n
7	9	religious.a ↔ church.n
7	20	lock.n ↔ key.n
7	25	teacher.n ↔ teach.v
7	26	check.v ↔ check.n
7	46	society.n ↔ class.n
0	14	competition.n ↔ compete.v
0	16	fox.n ↔ animal.n
0	24	smell.v ↔ smell.n
0	29	rehabilitation.n ↔ contact.n
0	33	guilty.a ↔ court.n
0	39	cat.n ↔ animal.n
0	40	similarity.n ↔ sequence.n
0	41	drive.v ↔ car.n
0	45	session.n ↔ experience.n
0	47	index.n ↔ close.v

Appendix C

Fixed-Length Topic Model Derivations

C.1 LPMRF Gibbs Sampling Derivation

As described in the main paper, we develop an LPMRF Gibbs sampler by considering the most common form of multinomial sampling, namely by taking the sum of a sequence of L Categorical variables. Thus, if $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_L \sim \text{Categorical}(\boldsymbol{\theta})$ and $\boldsymbol{x} = \sum_{\ell=1}^{L} \boldsymbol{w}_\ell \sim \text{Multinomial}(\boldsymbol{\theta}|L)$, then the probability of any particular sequence is merely the multinomial probability scaled by the inverse of the multinomial coefficient:

$$\mathbb{P}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_L \,|\, \boldsymbol{\theta}) = \binom{L}{x_1, x_2, \ldots, x_p}^{-1} \mathbb{P}_{\text{Mult}}(\boldsymbol{x} = \sum_{\ell=1}^{L} \boldsymbol{w}_\ell \,|\, \boldsymbol{\theta}, L). \tag{C.1}$$

In a similar way, we can implicitly derive the probability for a particular sequence of words whose sum is distributed as an LPMRF:

$$\mathbb{P}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_L \,|\, \boldsymbol{\theta}, \Phi) = \binom{L}{x_1, x_2, \ldots, x_p}^{-1} \mathbb{P}_{\text{LPMRF}}(\boldsymbol{x} = \sum_{\ell=1}^{L} \boldsymbol{w}_\ell \,|\, \boldsymbol{\theta}, \Phi, L) \tag{C.2}$$

$$= \exp(\boldsymbol{\theta}^T \boldsymbol{x} + \boldsymbol{x}^T \Phi \boldsymbol{x} - A_L(\boldsymbol{\theta}, \Phi) - \log(L!)). \tag{C.3}$$

To develop a Gibbs sampler, we simply need to compute the conditional probability of one of these words given all the other words. Letting $\boldsymbol{x}_{-\ell} \equiv \sum_{m \neq \ell} \boldsymbol{w}_m$, then $\boldsymbol{x} = \boldsymbol{x}_{-\ell} + \boldsymbol{w}_\ell$. Thus, using the fact that the conditional distribution is proportional to the joint distribution, we

can derive the form of the conditional distribution:

$$\mathbb{P}(\boldsymbol{w}_\ell = \mathbf{e}_s \,|\, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\ell-1}, \boldsymbol{w}_{\ell+1}, \ldots, \boldsymbol{w}_L, \boldsymbol{\theta}, \Phi) \tag{C.4}$$

$$\propto \exp(\boldsymbol{\theta}^T(\boldsymbol{x}_{-\ell} + \boldsymbol{w}_\ell) + (\boldsymbol{x}_{-\ell} + \boldsymbol{w}_\ell)^T \Phi(\boldsymbol{x}_{-\ell} + \boldsymbol{w}_\ell)) \tag{C.5}$$

$$\propto \exp(\boldsymbol{\theta}_s + 2\Phi_s \boldsymbol{x}_{-\ell}). \tag{C.6}$$

Thus, each word can be sampled given the state of all the other words thus producing an LPMRF Gibbs sampler.

C.2 Derivation of LPMRF Log Partition Upper Bound

$$A_L(\boldsymbol{\theta}, \Phi) \leq \log\Big[\Big(\sup_{\boldsymbol{x} \in \mathcal{X}_L} \exp(\boldsymbol{x}^T \Phi \boldsymbol{x}) \Big) \sum_{\boldsymbol{x} \in \mathcal{X}_L} \exp(\boldsymbol{\theta}^T \boldsymbol{x} - \textstyle\sum_s \log(x_s!)) \Big] \tag{C.7}$$

(Hölder's Inequality)

$$= \log\Big[\Big(\sup_{\boldsymbol{x} \in \mathcal{X}_L} \exp(\boldsymbol{x}^T \Phi \boldsymbol{x}) \Big) \exp(L \log(\textstyle\sum_s \exp(\theta_s)) - \log(L!)) \Big] \tag{C.8}$$

(Derived from multinomial)

$$\leq \log\Big[\exp(L^2 \lambda_{\Phi,1}) \exp(L \log(\textstyle\sum_s \exp \theta_s) - \log(L!)) \Big] \tag{C.9}$$

(Convex Relaxation of $\mathcal{X}_L$)

$$= L^2 \lambda_{\Phi,1} + L \log(\textstyle\sum_s \exp \theta_s) - \log(L!), \tag{C.10}$$

where $\lambda_{\Phi,1}$ is the maximum eigenvalue of Φ. See next section for derivation of Eqn. C.8.

C.2.1 Derivation of Multinomial Partition Function

Lemma 2. $\sum_{\boldsymbol{x} \in \mathcal{X}_L} \exp(\boldsymbol{\theta}^T \boldsymbol{x} - \sum_{s=1}^p \log(x_s!)) = \exp(L \log(\sum_s \exp \theta_s) - \log(L!))$

The derivation is based primarily on the fact that the above expression can be seen as

the normalizing factor of a reparameterized multinomial (or as a scaled version of a standard multinomial parameterization).

Proof.

$$
\begin{aligned}
&\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp(\boldsymbol{\theta}^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!)) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp(\boldsymbol{\theta}^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!) + c\boldsymbol{e}^T\boldsymbol{x} - c\boldsymbol{e}^T\boldsymbol{x} + \log(L!) - \log(L!)) \\
&\qquad \text{(where } \boldsymbol{e} = [1,1,\ldots,1]^T \text{ and } c \text{ is a constant)} \\
&= \frac{1}{L!}\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp((\boldsymbol{\theta}-c)^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!) + c\boldsymbol{e}^T\boldsymbol{x} + \log(L!)) \\
&= \frac{1}{L!}\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp((\boldsymbol{\theta}-c)^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!) + Lc + \log(L!)) \\
&= \frac{1}{L!}\exp(cL)\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp((\boldsymbol{\theta}-c)^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!) + \log(L!)) \\
&= \exp(Lc - \log(L!))\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp((\boldsymbol{\theta}-c)^T\boldsymbol{x} - \sum_{s=1}^{p}\log(x_s!) + \log(L!))
\end{aligned}
$$

Now continuing the simplification letting $c = \log\big(\sum_s^p \exp(\theta_s)\big)$, we get the following:

$$
\begin{aligned}
&\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp(\boldsymbol{\theta}^T\boldsymbol{x} - \sum_{s=1}^p \log(x_s!)) \\
&= \exp(L\log\big(\sum_s^p \exp(\theta_s)\big) - \log(L!)) \\
&\quad \times \sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp((\boldsymbol{\theta} - \log\big(\sum_s^p \exp(\theta_s)\big))^T\boldsymbol{x} - \sum_{s=1}^p \log(x_s!) + \log(L!)) \\
&= \exp(L\log\big(\sum_s^p \exp(\theta_s)\big) - \log(L!)) \underbrace{\sum_{\boldsymbol{x}\in\mathcal{X}_L} \exp(\log(\boldsymbol{\rho})^T\boldsymbol{x} - \sum_{s=1}^p \log(x_s!) + \log(L!))}_{\text{Partition function of multinomial} = 1} \qquad \text{(C.11)} \\
&= \exp(L\log\big(\sum_s^p \exp(\theta_s)\big) - \log(L!)) \underbrace{\sum_{\boldsymbol{x}\in\mathcal{X}_L} \frac{L!}{\prod_{s=1}^p x_s!} \prod_{s=1}^p \boldsymbol{\rho}_s^{x_s}}_{\text{Partition function of multinomial} = 1} \\
&= \exp(L\log\big(\sum_s^p \exp(\theta_s)\big) - \log(L!))
\end{aligned}
$$

where C.11 is derived by showing that $\boldsymbol{\rho} = \exp(\boldsymbol{\theta} - \log\big(\sum_s^p \exp(\theta_s)\big))$ is a valid standard multinomial parameter vector because the vector is positive and sums to 1:

$$
\sum_{t=1}^p \exp\Big(\theta_t - \log\big(\sum_s^p \exp(\theta_s)\big)\Big) = \frac{1}{\sum_{s=1}^p \exp(\theta_s)} \sum_{t=1}^p \exp(\theta_t) = \frac{\sum_{t=1}^p \exp(\theta_t)}{\sum_{s=1}^p \exp(\theta_s)} = 1.
$$

□

Appendix D

Generalized Root Model Derivations and Full Results

D.1 Node Conditional Derivation

$$\mathbb{P}(x_s \,|\, \boldsymbol{x}_{-s}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) = \mathbb{P}(\boldsymbol{x} = \boldsymbol{x}_{s0} + x_s \boldsymbol{e}_s \,|\, \boldsymbol{x}_{s0}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) \tag{D.1}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j} \langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{\boldsymbol{x}_{s0} + x_s \boldsymbol{e}_s} \circ^{\ell} \rangle + B(x_s)\right) \tag{D.2}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j} \langle \Psi^{(\ell)}_{(j)}, (\sqrt[j]{\boldsymbol{x}_{s0}} + \sqrt[j]{x_s \boldsymbol{e}_s}) \circ^{\ell} \rangle + B(x_s)\right) \tag{D.3}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j} \left\langle \Psi^{(\ell)}_{(j)}, \sum_{m=0}^{\ell} \binom{\ell}{m} \left(\sqrt[j]{\boldsymbol{x}_{s0}} \circ^{\ell-m}\right) \circ \left(\sqrt[j]{x_s \boldsymbol{e}_s} \circ^{m}\right) \right\rangle + B(x_s)\right) \tag{D.4}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\sum_{m=0}^{\ell} \binom{\ell}{m} \left\langle \Psi^{(\ell)}_{(j)}, \left(\sqrt[j]{\boldsymbol{x}_{s0}} \circ^{\ell-m}\right) \circ \left(\sqrt[j]{x_s \boldsymbol{e}_s} \circ^{m}\right) \right\rangle + B(x_s)\right) \tag{D.5}$$

Now we can further simplify:

$$\mathbb{P}(x_s \mid \boldsymbol{x}_{-s}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) \propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\sum_{m=0}^{\ell}\binom{\ell}{m}\left\langle\left[\Psi^{(\ell)}_{(j)}\right]_{\mathbf{I}(s,m)}, \sqrt[j]{\boldsymbol{x}_{s0}}\circ^{\ell-m}\right\rangle x_s^{m/j} + B(x_s)\right) \quad \text{(D.6)}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\sum_{m=1}^{\ell}\binom{\ell}{m}\left\langle\left[\Psi^{(\ell)}_{(j)}\right]_{\mathbf{I}(s,m)}, \sqrt[j]{\boldsymbol{x}_{s0}}\circ^{\ell-m}\right\rangle x_s^{m/j} + B(x_s)\right)$$

($m = 0$ is constant)

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\binom{\ell}{1}\left\langle\left[\Psi^{(\ell)}_{(j)}\right]_{\mathbf{I}(s,1)}, \sqrt[j]{\boldsymbol{x}_{s0}}\circ^{\ell-1}\right\rangle x_s^{1/j} + B(x_s)\right)$$

($m \geq 2$ are all zero since subtensors are zero by construction)

$$\propto \exp\left(\sum_{j=1}^{k}\left(\sum_{\ell=1}^{j}\ell\left\langle\left[\Psi^{(\ell)}_{(j)}\right]_{\mathbf{I}(s,1)}, \sqrt[j]{\boldsymbol{x}_{s0}}\circ^{\ell-1}\right\rangle\right)x_s^{1/j} + B(x_s)\right) \quad \text{(D.7)}$$

$$\propto \exp\left(\sum_{j=1}^{k}\eta_{js}x_s^{1/j} + B(x_s)\right), \quad \text{(D.8)}$$

where $\eta_{js} = \left(\sum_{\ell=1}^{j}\ell\left\langle\left[\Psi^{(\ell)}_{(j)}\right]_s, \sqrt[j]{\boldsymbol{x}_{s0}}\circ^{\ell-1}\right\rangle\right)$. See notation section for definition of $\left[\Psi^{(\ell)}_{(j)}\right]_s$. This is a univariate exponential family with sufficient statistics $x_s^{1/j}$, natural parameters η_{js}, and base measure $B(x_s)$. This recovers the SQR node conditional from [Inouye et al., 2016a] with $k = 2$.

D.2 Radial Conditional Derivation

As in [Inouye et al., 2016a], we define the *radial* conditional distribution by fixing the unit direction $\mathbf{v} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1}$ of the sufficient statistics but allowing the scaling $z = \|\boldsymbol{x}\|_1$ to be

unknown. Thus, we get the following *radial* conditional distribution:

$$\mathbb{P}(\boldsymbol{x} = z\mathbf{v} \,|\, \mathbf{v}, \boldsymbol{\Psi}^{(\cdot)}_{(\cdot)}) \propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{z\mathbf{v}}\circ^{\ell}\rangle + \textstyle\sum_s B(zv_s)\right) \tag{D.9}$$

$$\propto \exp\left(\sum_{j=1}^{k}\sum_{\ell=1}^{j}\langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{\mathbf{v}}\circ^{\ell}\rangle z^{\frac{\ell}{j}} + \textstyle\sum_s B(zv_s)\right) \tag{D.10}$$

$$\propto \exp\left(\sum_{r\in\mathcal{R}} \eta_r(\mathbf{v}) z^r + \tilde{B}_{\mathbf{v}}(z)\right), \tag{D.11}$$

where $\mathcal{R} = \{\ell/j : j \in \{1,\cdots,k\}, \ell \in \{1,\cdots,j\}\}$ is the set of possible ratios, $\eta_r(\mathbf{v}) = \sum_{\{(\ell,j):\ell/j=r\}}\langle \Psi^{(\ell)}_{(j)}, \sqrt[j]{\mathbf{v}}\circ^{\ell}\rangle$ are the exponential family parameters, z^r are the corresponding sufficient statistics, and $\tilde{B}_{\mathbf{v}}(z) = \sum_s B(zv_s)$ is the base measure. Thus, the radial conditional distribution is a univariate exponential family.

D.3 Derivation of $M(a)$ Approximation

$$M(a) \approx \log\sum_{i=1}^{d}\int_{\mathcal{D}_i}\exp(\hat{f}_i(x))\mathrm{d}\mu(x) \tag{D.12}$$

$$= \log\sum_{i=1}^{d}\exp(c_i)\int_{\mathcal{D}_i}\exp(\hat{\eta}_i x + B(x))\mathrm{d}\mu(x) \tag{D.13}$$

$$= \log\sum_{i=1}^{d}\exp(c_i)\exp\big(A(\hat{\eta}_i)\big)\Big(\mathrm{CDF}\big(\max(\mathcal{D}_i)\,|\,\hat{\eta}_i\big) - \mathrm{CDF}\big(\min(\mathcal{D}_i)\,|\,\hat{\eta}_i\big)\Big) \tag{D.14}$$

$$= \log\sum_{i=1}^{d}\exp\big(c_i + A(\hat{\eta}_i)\big)\Big(\mathrm{CDF}\big(\max(\mathcal{D}_i)\,|\,\hat{\eta}_i\big) - \mathrm{CDF}\big(\min(\mathcal{D}_i)\,|\,\hat{\eta}_i\big)\Big) \tag{D.15}$$

$$= \log\sum_{i=1}^{d}\exp\big(c_i + A(\hat{\eta}_i) + \log\big(\mathrm{CDF}\big(\max(\mathcal{D}_i)\,|\,\hat{\eta}_i\big) - \mathrm{CDF}\big(\min(\mathcal{D}_i)\,|\,\hat{\eta}_i\big)\big), \tag{D.16}$$

D.4 Linear Bounds of $g(x)$

Taylor series linear bound Upper bound if concavity = -1 and lower bound if concavity = 1:

$$q^* = \left\{ \begin{array}{ll} q_1 & , \text{if } q_2 = \infty \\ q_2 & , \text{if } q_1 = -\infty \\ \arg\max_{q_1,q_2} g(q) & , \text{otherwise} \end{array} \right\}$$

$$\begin{aligned} \hat{g}(x) &= g(q^*) + g'(q^*)(x - q^*) \\ &= \underbrace{g'(q^*)}_{b} x + \underbrace{\big(g(q^*) - q^* g'(q^*)\big)}_{c} \end{aligned}$$

Secant linear bound Upper bound if concavity = 1 and lower bound if concavity = -1:

$$\begin{aligned} b &= \frac{g(q_2) - g(q_1)}{q_2 - q_1} \\ g(q_1) &= bq_1 + c \\ \Rightarrow c &= g(q_1) - bq_1 \end{aligned}$$

Tail bounds We know there are only a finite number of inflection points so let us take the x value for the last inflection point, denoted x^*. By simple asymptotic analysis, we know that the largest non-zero term will dominate eventually. Let's assume w.l.o.g. that $\eta_{j^*} x^{\frac{1}{j^*}}$ dominates[1] and $\eta_{j^*} > 0$. Then, we know that after the last inflection point, the concavity will be negative. In addition, we know that the $g(x) \to \infty$ as $x \to \infty$. The function must be monotonically increasing after the last inflection point. Proof by contradiction: Suppose the monotonicity is negative after the last inflection point. Then, because the $g(x)$ is a continuous function and $g(x) \to \infty$ as $x \to \infty$, the function must eventually have a positive monotonicity. Yet this would switch from negative monotonicity to positive monotonicity after the last inflection point. However, this would be an inflection point that is greater than

[1] If for all j, $\eta_j = 0$, then we can take $j^* = \infty$, $\eta_{j^*} = a$.

the assumed last inflection point which leads to a contradiction. The case where $\eta_{j*} < 0$ can be proved similarly. Thus, we can use a constant function for an upper bound if concavity = 1. and we can use a constant function as a lower bound if concavity = -1. A Taylor series approximation forms an upper or lower bound depending on concavity.

D.5 Complete Results for Grolier and Classic3 Datasets

The full results of the top 50 words, pairs and triples for the Classic3 dataset can be see in Tables D.1 and D.2. The results for the Grolier dataset can be seen in Tables D.3 and D.4.

Table D.1: Top Words and Top Word Pairs for Classic3 Dataset

Top words		Top Positive Pairs		Top Negative Pairs	
-0.63	information	4.98	boundary + layer	-0.84	flow - library
-0.69	flow	4.26	heat + transfer	-0.64	information - pressure
-0.81	library	3.95	tunnel + wind	-0.59	flow - cells
-1.17	pressure	3.32	edge + leading	-0.59	pressure - library
-1.41	system	3.15	bone + marrow	-0.57	library - patients
-1.41	theory	3.04	angle + attack	-0.56	flow - system
-1.42	results	2.87	skin + friction	-0.56	information - cells
-1.42	data	2.56	growth + hormone	-0.55	information - heat
-1.46	patients	2.32	plate + flat	-0.52	information - patients
-1.57	found	2.27	shock + wave	-0.52	flow - patients
-1.58	method	2.25	mach + numbers	-0.49	theory - patients
-1.63	cells	2.13	number + mach	-0.49	information - normal
-1.66	analysis	2.10	number + reynolds	-0.47	information - found
-1.72	given	2.06	agreement + good	-0.46	information - effect
-1.72	use	2.03	attack + angles	-0.46	library - theory
-1.74	number	2.01	document + documents	-0.45	library - normal
-1.75	used	1.94	cells + cell	-0.45	library - cells
-1.76	study	1.77	journals + journal	-0.42	library - effects
-1.79	made	1.54	library + libraries	-0.41	flow - retrieval
-1.79	effect	1.53	lift + drag	-0.41	flow - growth
-1.81	time	1.51	wing + wings	-0.39	library - found
-1.81	body	1.43	shells + cylindrical	-0.39	library - cases
-1.84	research	1.42	buckling + shells	-0.37	information - wing
-1.86	cases	1.41	temperature + thermal	-0.37	information - case
-1.90	normal	1.41	free + stream	-0.36	pressure - cells
-1.92	effects	1.40	ratio + aspect	-0.35	flow - information
-1.94	present	1.39	equations + differential	-0.34	flow - subject
-1.97	discussed	1.37	boundary + layers	-0.34	results - library
-1.98	shock	1.37	point + stagnation	-0.33	information - effects
-1.99	presented	1.25	shock + waves	-0.32	information - temperature
-2.01	wing	1.25	heat + temperature	-0.32	information - surface
-2.01	surface	1.23	reynolds + transition	-0.31	flow - children
-2.03	large	1.20	wings + aspect	-0.31	library - obtained
-2.03	case	1.18	temperature + temperatures	-0.30	flow - book
-2.04	obtained	1.16	thin + shells	-0.29	flow - research
-2.06	new	1.13	science + scientific	-0.29	information - mach
-2.07	paper	1.13	cells + marrow	-0.29	theory - cells
-2.08	libraries	1.12	numbers + reynolds	-0.29	library - effect
-2.08	high	1.10	cylinder + circular	-0.28	information - equations
-2.09	problems	1.09	renal + kidney	-0.28	flow - literature
-2.12	methods	1.09	pressure + pressures	-0.28	flow - index
-2.12	well	1.04	high + speed	-0.28	flow - buckling
-2.13	development	1.03	layer + laminar	-0.27	analysis - patients
-2.14	general	1.03	information + retrieval	-0.27	information - cases
-2.14	growth	1.02	patients + therapy	-0.27	information - shock
-2.17	problem	1.02	patients + cancer	-0.26	information - boundary
-2.21	jet	1.01	jet + nozzle	-0.26	information - method
-2.21	terms	0.98	group + groups	-0.26	information - high
-2.23	systems	0.97	experimental + theoretical	-0.25	library - body
-2.24	form	0.95	buckling + stress	-0.25	information - ratio

Table D.2: Top Triples for Classic3 Dataset

Top Positive Triples						
0.51	layer	+	skin	+	friction
0.32	information	+	retrieval	+	storage
0.31	pressure	+	number	+	mach
0.31	layer	+	plate	+	flat
0.27	flow	+	given	+	case
0.24	flow	+	plate	+	flat
0.18	number	+	mach	+	investigation
0.14	number	+	mach	+	conducted
0.13	wing	+	ratio	+	aspect
0.13	number	+	based	+	reynolds
0.11	pressure	+	ratio	+	jet
0.11	heat	+	transfer	+	coefficients
0.10	system	+	retrieval	+	user
0.10	boundary	+	layer	+	experiments
0.09	mach	+	free	+	stream
0.09	pressure	+	layer	+	gradient
0.08	heat	+	temperature	+	coefficient
0.07	pressure	+	supersonic	+	base
0.07	boundary	+	shock	+	interaction
0.07	boundary	+	layer	+	distance
0.07	layer	+	shock	+	interaction
0.07	theory	+	experimental	+	experiment
0.06	flow	+	fluid	+	steady
0.06	flow	+	boundary	+	present
0.06	flow	+	body	+	revolution
0.05	flow	+	case	+	form
0.05	flow	+	body	+	shape
0.05	information	+	data	+	base
0.05	boundary	+	layer	+	found
0.05	cells	+	bone	+	marrow
0.05	flow	+	theory	+	approximation
0.05	data	+	retrieval	+	base
0.04	results	+	number	+	higher
0.04	layer	+	temperature	+	compressible
0.04	number	+	mach	+	static
0.04	boundary	+	injection	+	mass
0.04	number	+	mach	+	approximately
0.04	flow	+	hypersonic	+	region
0.04	theory	+	wing	+	wings
0.04	growth	+	human	+	hormone
0.04	number	+	mach	+	lower
0.04	heat	+	transfer	+	blunt
0.03	number	+	mach	+	increasing
0.03	number	+	boundary	+	increasing
0.03	boundary	+	layer	+	measurements
0.03	number	+	boundary	+	reynolds
0.03	flow	+	body	+	conditions
0.03	information	+	field	+	science
0.03	flow	+	number	+	based
0.03	flow	+	data	+	experimental

Top Negative Triples					
-0.35	boundary	-	layer	-	conditions
-0.20	number	-	mach	-	numbers
-0.09	boundary	-	layer	-	wing
-0.06	flow	-	number	-	numbers
-0.05	boundary	-	layer	-	time
-0.05	layer	-	shock	-	laminar
-0.04	number	-	mach	-	solution
-0.04	boundary	-	layer	-	jet
-0.03	heat	-	transfer	-	jet
-0.03	boundary	-	solutions	-	turbulent
-0.02	flow	-	mach	-	reynolds
-0.02	pressure	-	number	-	numbers
-0.01	boundary	-	layer	-	flutter
0.00	flow	-	mach	-	velocity
0.00	number	-	mach	-	problems

Table D.3: Top Words and Top Word Pairs for Grolier Dataset

Top words		Top Positive Pairs			Top Negative Pairs		
-1.62	american	8.71	km	+ mi	-0.24	life	- languages
-1.79	century	3.98	language	+ languages	-0.22	century	- species
-1.82	john	2.96	china	+ chinese	-0.20	city	- species
-1.88	called	2.62	plants	+ plant	-0.16	war	- species
-1.89	city	2.52	deg	+ temperatures	-0.12	city	- sq
-1.92	world	2.52	music	+ musical	-0.09	century	- june
-1.95	life	2.41	spanish	+ spain	-0.09	war	- languages
-2.04	united	2.15	novel	+ novels	-0.08	war	- example
-2.13	system	2.11	art	+ painting	-0.07	city	- theory
-2.13	university	2.09	poetry	+ poet	-0.07	city	- common
-2.14	family	2.07	agricultural	+ agriculture	-0.07	city	- system
-2.15	time	2.05	war	+ civil	-0.07	city	- called
-2.16	war	2.00	literature	+ literary	-0.07	century	- cells
-2.19	include	1.86	french	+ france	-0.06	war	- cells
-2.19	english	1.84	german	+ germany	-0.05	american	- eng
-2.25	water	1.78	culture	+ cultural	-0.04	city	- english
-2.25	history	1.75	china	+ asia	-0.04	century	- president
-2.26	de	1.74	india	+ asia	-0.04	american	- ft
-2.27	form	1.71	system	+ systems	-0.04	art	- america
-2.33	major	1.68	city	+ york	-0.04	city	- found
-2.34	national	1.68	west	+ east	-0.04	city	- form
-2.35	french	1.59	africa	+ african	-0.03	city	- development
-2.35	william	1.59	deg	+ mm	-0.03	war	- form
-2.37	art	1.51	southern	+ northern	-0.03	war	- usually
-2.38	found	1.50	architecture	+ building	-0.03	called	- deg
-2.40	name	1.48	style	+ architecture	-0.03	war	- forms
-2.40	modern	1.44	body	+ blood	-0.02	city	- time
-2.43	music	1.44	role	+ played	-0.02	city	- united
-2.43	power	1.43	sea	+ ocean	-0.02	war	- human
-2.44	king	1.43	cells	+ blood	-0.02	century	- july
-2.44	social	1.41	science	+ scientific	-0.02	century	- party
-2.45	british	1.41	century	+ centuries	-0.02	war	- theory
-2.46	usually	1.38	population	+ sq	-0.02	american	- king
-2.47	charles	1.37	social	+ society	-0.01	city	- family
-2.48	south	1.36	italian	+ renaissance	-0.01	century	- south
-2.49	law	1.36	music	+ opera	-0.01	called	- eng
-2.50	north	1.36	ocean	+ pacific	-0.01	city	- life
-2.50	repr	1.36	cause	+ disease	-0.01	american	- deg
-2.52	species	1.35	cities	+ urban	-0.01	american	- east
-2.52	theory	1.34	war	+ army	-0.01	american	- city
-2.54	human	1.33	united	+ countries	-0.01	war	- water
-2.55	ft	1.31	animals	+ animal	-0.01	city	- usually
-2.55	black	1.30	church	+ christian	-0.01	american	- cells
-2.56	government	1.30	art	+ museum	-0.01	form	- university
-2.57	west	1.29	education	+ schools	0.00	city	- body
-2.58	york	1.28	programs	+ program	0.00	city	- process
-2.58	church	1.27	deg	+ temperature	0.00	century	- american
-2.58	school	1.26	world	+ war	0.00	ft	- english
-2.59	development	1.25	party	+ leader	0.00	city	- cells
-2.59	common	1.25	government	+ federal			

Table D.4: Top Triples for Grolier Dataset

Top Positive Triples						
0.31	american	+	city	+	york
0.28	city	+	population	+	center
0.25	population	+	deg	+	mm
0.20	major	+	population	+	persons
0.16	ft	+	sea	+	level
0.15	american	+	south	+	america
0.15	deg	+	sq	+	consists
0.14	city	+	deg	+	july
0.13	war	+	civil	+	union
0.12	population	+	deg	+	elected
0.12	american	+	united	+	english
0.11	war	+	congress	+	program
0.11	population	+	sq	+	persons
0.10	american	+	french	+	british
0.10	language	+	includes	+	languages
0.10	world	+	war	+	japanese
0.09	major	+	time	+	changes
0.09	century	+	world	+	laws
0.09	life	+	human	+	stage
0.09	north	+	south	+	president
0.09	city	+	population	+	university
0.08	century	+	world	+	war
0.08	km	+	mi	+	discovered
0.08	war	+	united	+	received
0.08	city	+	river	+	historical
0.08	century	+	history	+	short
0.07	century	+	english	+	story
0.07	war	+	south	+	union
0.07	american	+	war	+	congress
0.07	world	+	united	+	david
0.07	century	+	history	+	active
0.07	century	+	history	+	wide
0.07	war	+	united	+	america
0.07	war	+	army	+	june
0.07	city	+	km	+	population
0.07	government	+	national	+	rise
0.07	century	+	form	+	appeared
0.06	war	+	united	+	caused
0.06	century	+	world	+	separate
0.06	united	+	people	+	continued
0.06	city	+	united	+	urban
0.06	century	+	time	+	applied
0.06	world	+	united	+	building
0.06	world	+	south	+	iron
0.06	world	+	population	+	rate
0.06	city	+	university	+	center
0.06	century	+	time	+	studied
0.06	american	+	war	+	people
0.06	north	+	km	+	fish
0.06	world	+	william	+	series

Top Negative Triples					
-0.26	city	-	population	-	york
-0.12	km	-	mi	-	america
-0.11	american	-	km	-	mi
-0.06	km	-	mi	-	york
-0.05	war	-	north	-	example
-0.04	km	-	mi	-	social
-0.04	km	-	mi	-	family
-0.03	km	-	mi	-	own
-0.03	km	-	mi	-	style
-0.03	city	-	population	-	style
-0.03	km	-	mi	-	theory
-0.03	city	-	center	-	sq
-0.02	km	-	mi	-	law
-0.02	km	-	mi	-	human
-0.02	km	-	mi	-	example
-0.02	city	-	population	-	greek
-0.01	km	-	mi	-	water
-0.01	called	-	km	-	mi
-0.01	england	-	language	-	languages
-0.01	km	-	mi	-	greek
-0.01	time	-	km	-	mi
0.00	km	-	mi	-	english
0.00	mi	-	population	-	america
0.00	population	-	america	-	sq
0.00	war	-	km	-	mi
0.00	american	-	city	-	center

Appendix E

Supplementary Material for Poisson Review, Chapter 5

E.1 Supplementary Datasets and Results

We describe and give results for a crime statistics dataset and the 20 Newsgroup dataset. As mentioned in the paper, these datasets behave similarly to the the BRCA and Classic3 datasets respectively but we include them here for completeness and for additional evidence of the observations described in the paper. The dataset statistics can be seen in Table 5.1. The results for the crime statistics can be seen in Fig. E.1 and the 20 Newsgroup results can be seen in Fig. E.2.

1. **Crime count dataset (Medium counts, medium overdispersion):** Aggregated crime counts from LAPD during the years 2012-2015.[1] The original dataset contains 151 types of crime counts such as "Burglary" and "Vandalism". This dataset exhibits a wide range of mean values with weak correlation and weak overdispersion.
2. **20 Newsgroup text dataset (Low counts, medium overdispersion):** Standard text corpus for document classification with almost 1000 forum posts from 20 different newsgroups. [2]

[1] `https://data.lacity.org/A-Safe-City/Crimes-2012-2015/s9rj-h3s6`. We removed year 2013 and November of 2015 which both clearly had a different distribution than other years likely due to different classification systems.

[2] `http://qwone.com/~jason/20Newsgroups/` We slightly modified this dataset by removing words that were merely for structure or were clearly outliers: "line", "subject", "organ", "re", "post", "host", "nntp", and "maxaxaxaxaxaxaxaxaxaxaxaxaxax". The raw dataset contained very strong outliers, e.g. the word "1" had a mean of 0.58 and a standard deviation of 5.89 but had a maximum value of 344—more than 50 standard deviations away from the mean. Thus, for each variable, we truncated the values beyond the the 99.5th percentile to be the 99.5th percentile; thus, at most 0.5% of values were truncated per variable.

Table E.1: Dataset Statistics

(Per Variable ⇒)			Means			Dispersion Indices			Spearman's ρ		
Dataset	p	n	Min	Med	Max	Min	Med	Max	Min	Med	Max
Crime LAPD	10	1035	25	39	118	1.4	2.1	6.2	-0.19	0.14	0.52
	100	1035	0.06	0.77	118	0.91	1.3	16	-0.49	0.02	0.78
20News	10	18846	0.36	0.5	1.4	0.59	1.7	6.2	-0.37	0.03	0.67
	100	18846	0.07	0.15	1.4	0.83	1.9	6.2	-0.37	0.05	0.67

E.2 Implementation Details

E.2.1 Copulas Paired with Poison Marginals

As stated in the paper, we estimated the copula-based models using the Inference Function for Margins (IFM) method via the distributional transform. More specifically, we first estimated the Poisson marginal distributions. Then, we computed the distributional transform to map the data from the discrete domain to the continuous domain, i.e. $u = (F(x) + F(x-1))/2$ where $F(\cdot)$ is the Poisson CDF.[3] Finally, we estimate the copula distribution using either the `copulafit` function in MATLAB or the `RVineStructureSelect` function in the `VineCopula`[4] R package for the Gaussian and vine copulas respectively. For the vine copula, the vine structure and bivariate copulas were automatically selected in the `RVineStructureSelect` function; we allowed the following six bivariate copulas and their rotations: Gaussian copula, Student's t copula, Clayton copula, Gumbel copula, Frank copula, and Joe copula.

[3]We chose the DT transform because it is computationally and conceptually the simplest of estimation methods even though more complex methods exist for a small number of samples [Nikoloulopoulos, 2016].

[4]`https://cran.r-project.org/web/packages/VineCopula/index.html`

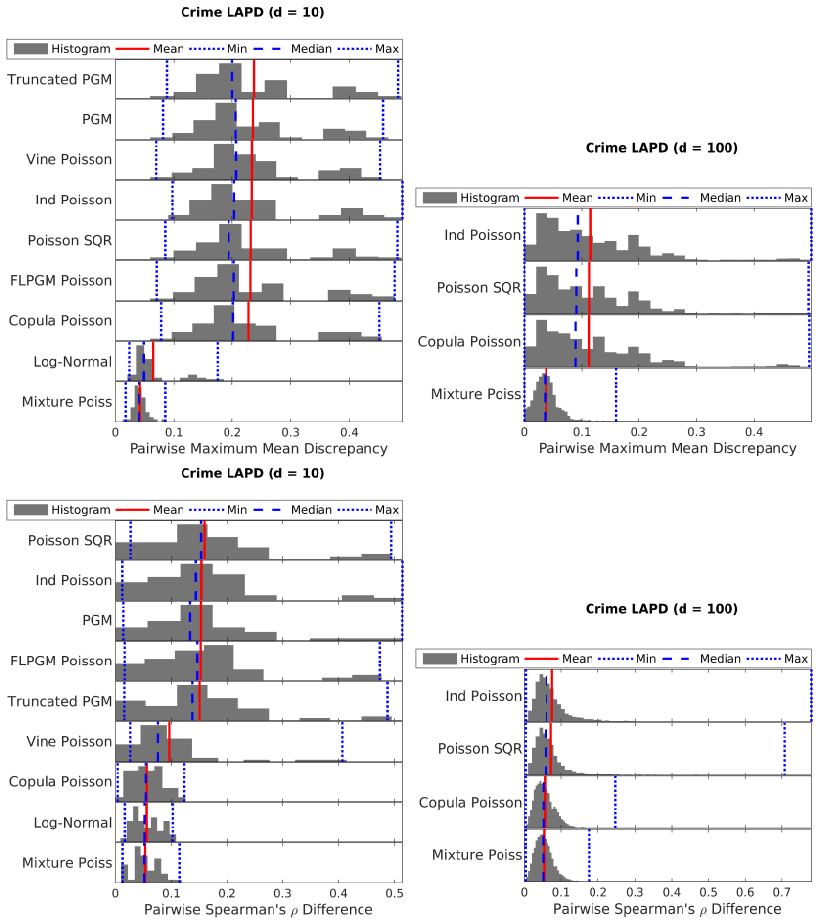


Figure E.1: The results for the LAPD crime statistics dataset with medium count values and medium overdispersion behave similarly to the results from the BRCA dataset described in the paper.

E.2.2 Mixture Models

For the finite mixture of independent Poissons, we initialized the EM algorithm with the best of 10 k-means clusterings and set the maximum number of EM iterations to one hundred. Note that even in high dimensions, the EM algorithm usually converged in under 20 iterations. For the log-normal mixture, we used 1000 iterations and 400 burn-in iterations for the MCMC algorithm.

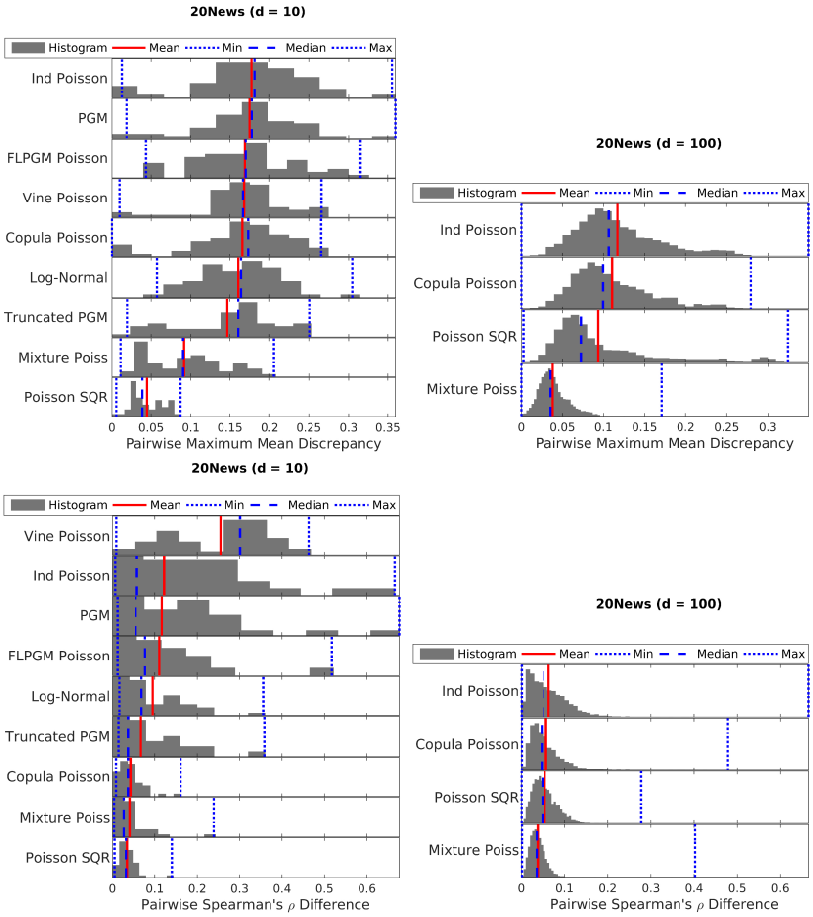


Figure E.2: The results for the 20 Newsgroup dataset with low count values and medium overdispersion behave very similarly to the results from the Classic3 dataset described in the paper.

E.2.3 Conditional Models

We set the truncation value R to the 99th percentile of the non-zeros in the training dataset. This helped avoid expensive computations if there was one or two very large outliers since each gradient iteration requires R exponential evaluations per observation. We tested 10 regularization parameters of log-spaced points between $\lambda_{\max}$ and $0.0001\lambda_{\max}$ where $\lambda_{\max}$ is the max value of the off diagonals of the training data empirical second moment matrix. In

the case of the Poisson SQR, we set $\lambda_{\max}^{\text{SQR}} = \sqrt{\lambda_{\max}}$ because the sufficient statistics are square roots of the original sufficient statistics. Essentially, this initially estimates an independent model and slowly moves toward a highly dependent model by reducing the regularization parameter.

E.3 Sampling Details

For the models based on pairing copulas with Poisson marginals, we first sampled from the copula either using the `copularnd` MATLAB function in the case of the Gaussian copula or `RVineSim` from the VineCopula R package in the case of the vine copulas; then, we transformed the copula samples to the discrete domain using the Poisson marginal CDFs. For the mixture models, sampling is also straightforward; we sampled the Poisson mean from the finite mixture or log-normal distribution and then sampled a Poisson variable given this mean. For the PGM, TPGM and Poisson SQR models, we used Gibbs sampling with 5,000 iterations. Because the Poisson SQR conditionals are non-standard, we implemented the Gibbs iterations using two steps of Metropolis-Hastings rejection sampling. For the FLPGM models, we used the annealed importance sampling routines provided by the authors of [Inouye et al., 2015] with 100 annealing steps. Overall, the copula-based and mixture models have direct sampling routines whereas the conditional models have natural procedures for Gibbs sampling.

Appendix F

Visualization Algorithmic Experiments and Extra Example Visualizations

F.1 Algorithm Phases Figure

The main phases of the visualization algorithm can be viewed in Figure F.1.

F.2 Algorithm Experiment Figures

Figure F.2 shows that the visualization either has too much empty space or distorts the underlying graph function if the target number of violated constraints is too large or too small. For Figure F.2, all the phases are executed except for the last strong gravity phase, which would distort the comparison because it is meant merely as a final clean-up phase.

We compare our reverse simulated annealing algorithm to Wordle's spiral algorithm in Figure F.3 and show that our algorithm performs better both quantitatively in terms of objective function and qualitatively but again place the figure in the supplementary materials because of space constraints.

F.3 More Example Visualizations

Several more example visualizations for the natural science text dataset, airport delay times and daily stock returns can be seen in Figure F.4, Figure F.5, and Figure F.6 respectively.

Figure F.1: Each phase of the layout algorithm is important for a compact though meaningful layout (via graph layout). The phases are (from left to right, top to bottom): random initialization, unconstrained simulated annealing, font scaling, feasible projection onto non-overlap set via reverse simulated annealing, constrained simulated annealing and finally a strong gravity phase of constrained simulated annealing. See subsection 12.5.1 for dataset description.

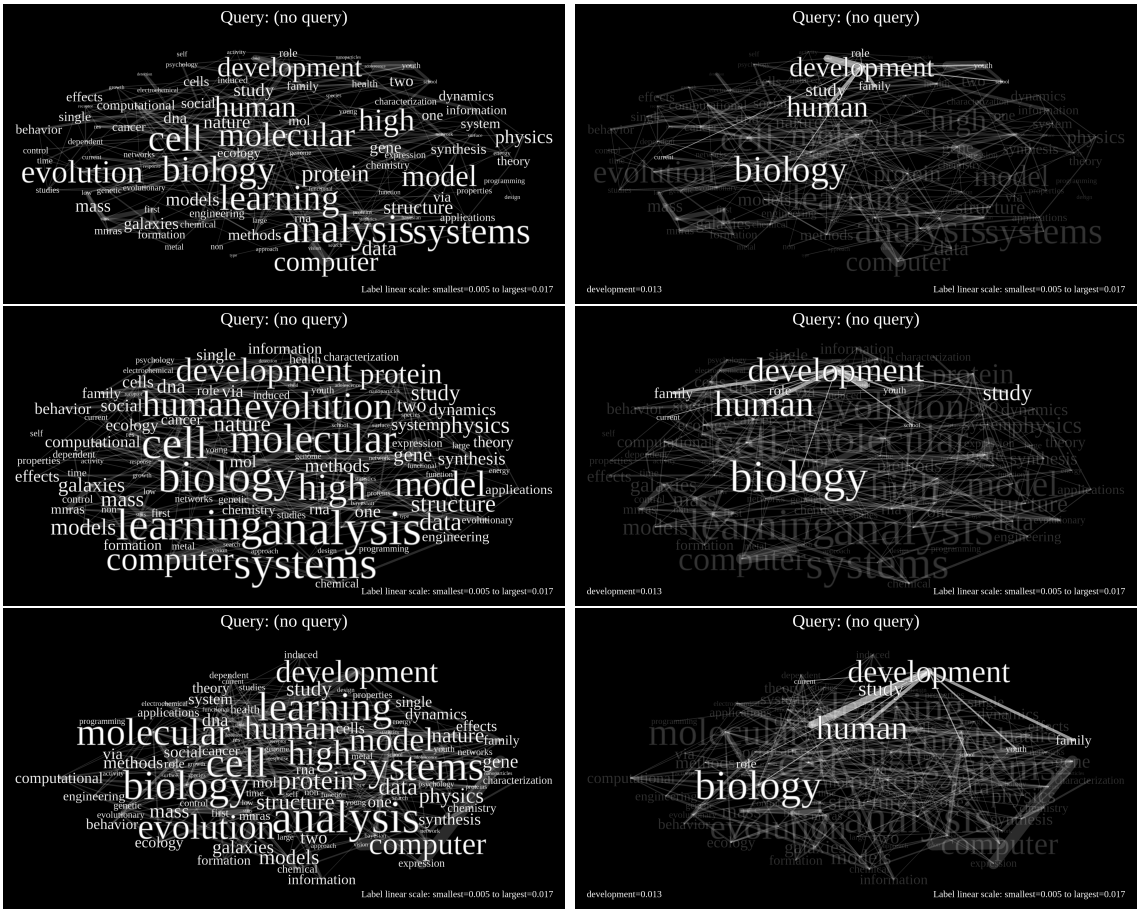


Figure F.2: During font scaling, if the number of violated constraints is too small (top), the visualization will not be compact such that there will be empty space. However, if the number of violated constraints is too large (bottom), the projection phase significantly impairs the graph layout optimization. We select a value between these two extremes (middle). The lower quality of projection can be seen by highlighting the word "development" and noticing that the $m = 2p$ visualization (bottom) puts it farther from "child" (with thick edge) and "human". The number of violated constraints is $p/2$, p and $2p$ from top to bottom, and the graph optimization values (lower is better) are -8838, -8710 and -7317 from top to bottom. We did not run the final strong gravity phase in order to better measure the effect of font scaling. See subsection 12.5.1 for dataset description.

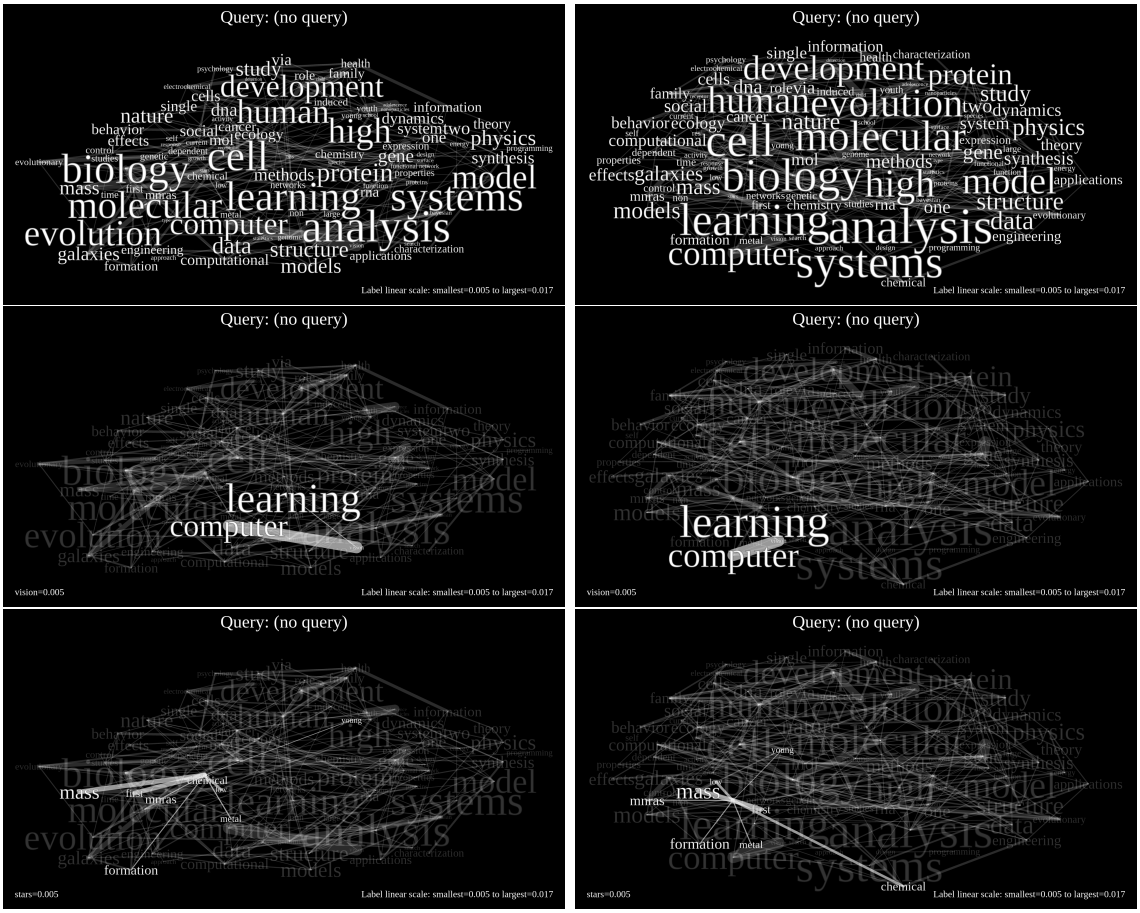


Figure F.3: Projecting the labels onto the feasible set (i.e. no overlaps) using the standard spiral technique (left) does not perform as well as projecting using reverse simulated annealing on the underlying graph optimization function (right). Quantitatively, the optimization values after projection (lower is better) were -8391 for the spiral and -8571 for reverse simulated annealing. Qualitatively, when highlighting the words "vision" (middle) and "stars" (bottom), the spiral technique (left) shows longer thick edges than the reverse annealing (right). Furthermore, the spiral technique (top left) shows more empty space than the reverse annealing (top right). See subsection 12.5.1 for dataset description.

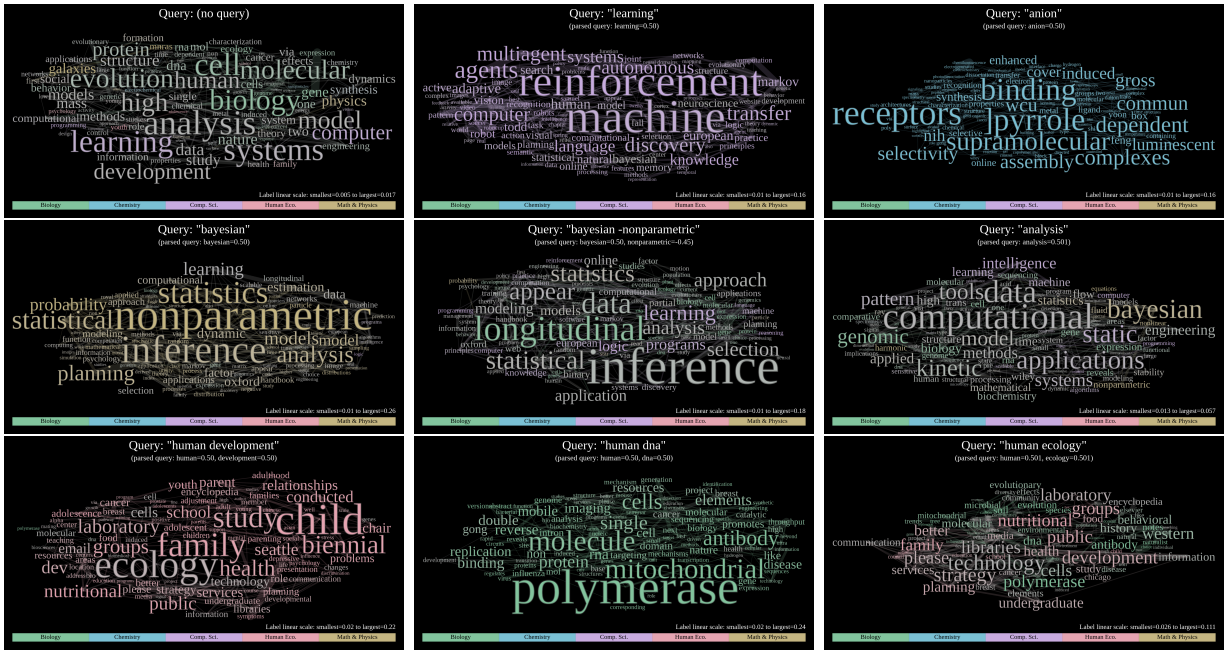


Figure F.4: These 'what if' visualizations demonstrate how the query intuitively manipulates the underlying probabilistic model and displays related variables (in this case words) in an interpretable fashion. All of these visualizations differ significantly from the visualization with no query. Some words only occur in one subject, such as "electrochemical" in chemistry; these retain the same color in all visualizations. Other words are used in multiple subjects; for example, "model" is used in many subjects and thus appears gray with no query, but becomes yellow when using the query "bayesian". Note that queries can be multiple words, affording very different viewpoints such as the bottom three visualizations. In addition, queries can include 'negative' queries with the minus sign such as "bayesian -nonparametric"; these condition on a high chance of "bayesian" occurring but a low chance of "nonparametric" occurring.

Figure F.5: These visualizations show that long airport delay times often occur during the winter and possibly autumn months likely due to weather delays. The no query visualization readily shows that the Chicago airports (MDW and ORD) have long delays in general. Querying on "ORD(IL) JFK(NY)" means that Chicago and New York have long delays and the yellow color suggests that other high delays likely belong to winter days. The negative query of "-ORD(IL) -MDW(IL) -JFK(NY)" means that neither the Chicago or New York airports have long delays and thus there is no cold weather delays at least in the midwest and northeast; however, distant California airports, namely ACV and SFO, may have long delays.

Figure F.6: The query "joy", an energy company, demonstrates that if one energy company is doing well, many other energy companies also do well (e.g. cnx and dnr). However, if technology companies are doing well as suggested by the "msft amd" query, other technology stocks perform well (e.g. nvda). Similar patterns exist for health care stocks when querying "alxn -joy" since alxn is a pharmaceutical company. Finally, when querying "++hig", for Hartford Financial Services, other financial companies do well, and the visualization shows three clusters of financial stocks centered around Lincoln National Corporation, JP Morgan Chase and KeyBank.

Bibliography

K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

J. Aguero-Valverde and P. P. Jovanis. Bayesian multivariate Poisson lognormal models for crash severity modeling and site ranking. *Transportation Research Record: Journal of the Transportation Research Board*, 2136(-1):82–91, 2009.

J. Aitchison and C. Ho. The multivariate Poisson-log normal distribution. *Biometrika*, 76 (4):643–653, 1989.

N. Aletras and M. Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22, 2013.

G. I. Allen and Z. Liu. A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *Bioinformatics and Biomedicine, 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

G. I. Allen and Z. Liu. A local Poisson graphical model for inferring networks from sequencing data. *IEEE Trans. on Nanobioscience*, 12(3):189–198, 2013.

P. M. E. Altham. Two generalizations of the binomial distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 27(2):162–167, 1978.

A. G. Arbous and J. Kerrich. Accident statistics and the concept of accident-proneness. *Biometrics*, 7(4):340–432, 1951.

O. Banerjee, L. El Ghaoui, and A. D'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR*, 9:485–516, 2008.

L. Barth, S. I. Fabrikant, S. G. Kobourov, A. Lubiw, M. Nöllenburg, Y. Okamoto, S. Pupyrev, C. Squarcella, T. Ueckerdt, and A. Wolff. Semantic word cloud representations: Hardness and approximation algorithms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8392 LNCS: 514–525, 2014a.

L. Barth, S. Kobourov, and S. Pupyrev. Experimental comparison of semantic word clouds. *13th International Symposium, SEA 2014*, pages 237–258, 2014b.

T. Bedford and R. M. Cooke. Vines: A new graphical model for dependent random variables. *Annals of Statistics*, pages 1031–1068, 2002.

J. a. Bennell and J. F. Oliveira. The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415, 2008.

R. S. Berns. *Billmeyer and Saltzman's Principles of Color Technology*. John Wiley and Sons, Hoboken, New Jersey, 3rd editio edition, 2001.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

Y. Bishop, S. Fienberg, and P. Holland. Sampling models for discrete data. In *Discrete Multivariate Analysis: Theory and Practice*, chapter 13, pages 435–456. Springer, 2007.

D. Blei, L. Carin, and D. Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6):55–65, nov 2010.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3(1):993–1022, 2003.

D. M. Blei, J. D. Lafferty, and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

D. Borland and R. M. Taylor. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007.

J. Boyd-Graber, C. Fellbaum, D. Osherson, and R. Schapire. Adding dense, weighted connections to wordnet. In *Proceedings of the Third International WordNet Conference*, pages 29–36, 2006.

J. R. Bradley, S. H. Holan, and C. K. Wikle. Computationally efficient distribution theory for Bayesian inference of high-dimensional dependent count-valued data. *arXiv preprint arXiv:1512.07273*, 2015.

J. Campbell. The Poisson correlation function. *Proceedings of the Edinburgh Mathematical Society*, 4(01):18–26, 1934.

J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

U. Cherubini, E. Luciano, and W. Vecchiato. *Copula Methods in Finance*. John Wiley and Sons, 2004a.

U. Cherubini, E. Luciano, and W. Vecchiato. Simulation of market scenarios. In *Copula Methods in Finance*, chapter 6. Wiley, 2004b.

S. Chib and R. Winkelmann. Markov chain Monte Carlo analysis of correlated count data. *Journal of Business & Economic Statistics*, 19(4):428–435, 2001.

R. T. Clemen and T. Reilly. Correlations and copulas for decision and risk analysis. *Management Science*, 45(2):208–224, 1999.

M. Collins and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *NIPS*, pages 617–624, 2001.

R. J. Cook, J. F. Lawless, and K.-A. Lee. A copula-based mixed Poisson model for bivariate recurrent events under event-dependent censoring. *Statistics in Medicine*, 29(6):694–707, 2010.

W. Cui, Y. Wu, S. Liu, F. Wei, and M. Zhou. Context preserving dynamicword cloud visualization. *IEEE Pacific Visualisation Symposium*, pages 121–128, 2010.

C. Czado, E. C. Brechmann, and L. Gruber. Selection of vine copulas. In *Copulae in Mathematical and Quantitative Finance*, pages 17–37. Springer, 2013.

S. Demarta and A. J. McNeil. The t copula and related copulas. *International Statistical Review*, 73(1):111–129, 2005.

M. Denuit and P. Lambert. Constraints on concordance measures in bivariate discrete data. *Journal of Multivariate Analysis*, 93(1):40–57, 2005.

M. Dwass and H. Teicher. On infinitely divisible random vectors. *Annals of Mathematical Statistics*, pages 461–470, 1957.

T. Dwyer. Scalable, versatile and simple constrained graph layout. *Computer Graphics Forum*, 28(3):991–998, 2009.

K. El-Basyouny and T. Sayed. Collision prediction models using multivariate Poisson-lognormal regression. *Accident Analysis & Prevention*, 41(4):820–828, 2009.

J. Feinberg. Wordle. In J. Steele and N. Iliinsky, editors, *Beautiful Visualization: Looking at data through the eyes of experts*, chapter 3, pages 37–58. O'Reilly Media, Inc., 2010.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, jul 2008.

P. Gambette and J. Véronis. Visualising a text with a tree cloud. In *Classification as a Tool for Research*, pages 561–569. Springer, 2010.

C. Genest and J. Nešlehová. A primer on copulas for count data. *ASTIN Bulletin*, 37(2): 475–515, 2007.

A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities*, volume 195. Springer, 2009.

P. Gopalan, J. M. Hofman, and D. M. Blei. Scalable recommendation with Poisson factorization. *arXiv preprint*, pages 1–10, 2013.

M. J. Greenacre. *Biplots in practice*. Fundacion BBVA, 2010.

A. Gretton. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl:5228–35, apr 2004.

J. Gwizdka and P. Bakelaar. Navigating one million tags. *Proceedings of the American Society for Information Science and Technology*, 46(1):1–7, 2009.

F. Hadiji, A. Molina, S. Natarajan, and K. Kersting. Poisson dependency networks: Gradient boosted models for multivariate count data. *Machine Learning*, 100(2-3):477–507, 2015.

S. W. Han and H. Zhong. Estimation of sparse directed acyclic graphs for multivariate counts data. *Biometrics*, 2016.

M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Map Reader: Theories of Mapping Practice and Cartographic Representation*, 40(1):261–268, 2011.

A. Heinen and E. Rengifo. Multivariate autoregressive modeling of time series count data using copulas. *Journal of Empirical Finance*, 14(4):564–583, 2007.

A. Heinen and E. Rengifo. Multivariate reduced rank regression in non-Gaussian contexts, using copulas. *Computational Statistics and Data Analysis*, 52(6):2931–2944, 2008.

T. Hofmann. Probabilistic latent semantic analysis. In *UAI'99 Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

P. Holgate. Estimation for the bivariate Poisson distribution. *Biometrika*, 51(1-2):241–287, 1964.

C. Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. *Nips*, pages 1–9, 2011.

C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *JMLR*, 15:2911–2947, 2014.

R. Ihaka. Colour for presentation graphics. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing Vienna Austria*, (Dsc):1–18, 2003.

D. Inouye, P. Ravikumar, and I. Dhillon. Capturing semantically meaningful word dependencies with an admixture of Poisson MRFs. In *Advances in Neural Information Processing Systems*, volume 4, pages 3158–3166, 2014a.

D. I. Inouye, P. Ravikumar, and I. S. Dhillon. Admixture of Poisson MRFs: A topic model with word dependencies. In *ICML*, 2014b.

D. I. Inouye, P. Ravikumar, and I. S. Dhillon. Fixed-length Poisson MRF: Adding dependencies to the multinomial. In *NIPS*, pages 3195–3203, 2015.

D. I. Inouye, P. Ravikumar, and I. S. Dhillon. Square root graphical models: Multivariate generalizations of univariate exponential families that permit positive dependencies. In *ICML*, 2016a.

D. I. Inouye, P. Ravikumar, and I. S. Dhillon. Generalized root models: Beyond pairwise graphical models for univariate exponential families. *arXiv preprint arXiv:1606.00813*, 2016b.

D. I. Inouye, E. Yang, G. I. Allen, and P. Ravikumar. A review of multivariate distributions for count data derived from the Poisson distribution. *WIREs Computational Statistics (arXiv preprint:1609.00066)*, 2017.

A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. In *AISTATS*, pages 378–387, 2010.

H. Joe and J. J. Xu. *The Estimation Method of Inference Functions for Margins for Multivariate Models*. Technical report 166, The University of British Columbia, Vancouver, Canada, 1996.

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*, volume 165. Wiley New York, 1997.

E. Kaarik. Imputation by conditional distribution using Gaussian copula. In *Compstat*, pages 1447–1454, 2006.

E. Käärik and M. Käärik. Modeling dropouts by conditional distribution, a copula-based approach. *Journal of Statistical Planning and Inference*, 139(11):3830–3835, 2009.

M. S. Kaiser and N. Cressie. Modeling Poisson variables with positive spatial dependence. *Statistics & Probability Letters*, 35(4):423–432, 1997.

T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

K. Kano and K. Kawamura. On recurrence relations for the probability function of multivariate generalized Poisson distribution. *Communications in statistics-theory and methods*, 20(1):165–178, 1991.

D. Karlis. An EM algorithm for multivariate Poisson distribution and related models. *Journal of Applied Statistics*, 30(1):63–77, 2003.

D. Karlis. Models for multivariate count time series. In R. A. Davis, S. H. Holan, R. Lund, and N. Ravishanker, editors, *Handbook of Discrete-Valued Time Series*, chapter 19, pages 407–424. CRC Press, 2016.

D. Karlis and L. Meligkotsidou. Finite mixtures of multivariate Poisson distributions with application. *Journal of statistical Planning and Inference*, 137(6):1942–1960, 2007.

D. Karlis and E. Xekalaki. Mixed Poisson distributions. *International Statistical Review*, 73 (1):35–58, 2005.

K. Kawamura. The structure of multivariate Poisson distribution. *Kodai Mathematical Journal*, 2(3):337–345, 1979.

H. Kazianka. Approximate copula-based estimation and prediction of discrete spatial data. *Stochastic Environmental Research and Risk Assessment*, 27(8):2015–2026, 2013.

H. Kazianka and J. Pilz. Copula-based geostatistical modeling of continuous and discrete data including covariates. *Stochastic Environmental Research and Risk Assessment*, 24 (5):661–673, 2010.

S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

J. D. Kort. *Pricing multi-asset financial products with tail dependence using copulas*. PhD thesis, Delft University of Technology, 2007.

A. Krishnamoorthy. Multivariate binomial and Poisson distributions. *Sankhyā: the Indian Journal of Statistics*, pages 117–124, 1951.

F. Krummenauer. Limit theorems for multivariate discrete distributions. *Metrika*, 47(1): 47–69, 1998.

D. M. B. Lafferty and J. D. Correlated topic models. In *Advances in Neural Information Processing Systems 18*, pages 147–154, 2006.

K. K. Lai and J. W. M. Chan. Developing a simulated annealing algorithm for the cutting stock problem. *Computers & Industrial Engineering*, 32(1):115–127, 1997.

J. Lau, K. Grieser, D. Newman, and T. Baldwin. Automatic labeling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1536–1545, 2011.

J. H. Lau, P. Cook, D. McCarthy, D. Newman, T. Baldwin, and L. Computing. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for computational Linguistics (EACL 2012)*, pages 591–601, 2012.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Lecture Notes in Computer Science*, volume 4029, pages 548–562, 2006.

J. Lee and T. Hastie. Structure learning of mixed graphical models. In *Aistats 16*, volume 31, pages 388–396, 2013.

Y.-Y. Lee, C.-C. Lin, and H.-C. Yen. Mental map preserving graph drawing using simulated annealing. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, volume 60, pages 179–188, 2006.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):34, 2012.

S. Loukas and C. Kemp. On computer sampling from trivariate and multivariate discrete distributions: Multivariate discrete distributions. *Journal of Statistical Computation and Simulation*, 17(2):113–123, 1983.

J. Ma, K. M. Kockelman, and P. Damien. A multivariate Poisson-lognormal regression model for prediction of crash counts by severity, using Bayesian methods. *Accident Analysis & Prevention*, 40(3):964–975, 2008.

L. Madsen. Maximum likelihood estimation of regression parameters with spatially dependent discrete data. *Journal of Agricultural, Biological, and Environmental Statistics*, 14(4):375–391, 2009.

L. Madsen and Y. Fang. Joint regression analysis for discrete longitudinal data. *Biometrics*, 67(3):1171–1175, 2011.

D. Magatti, S. Calegari, D. Ciucci, and F. Stella. Automatic labeling of topics. In *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, pages 1227–1232, 2009.

D. Mahamunulu. A note on regression in the multivariate Poisson distribution. *Journal of the American Statistical Association*, 62(317):251–258, 1967.

X.-l. Mao, Z.-y. Ming, Z.-j. Zha, T.-s. Chua, H. Yan, and X. Li. automatic labeling hierarchical topics. In *Cikm2012*, pages 2383–2386, 2012.

A. K. McCallum. Mallet: A machine learning for language toolkit, 2002.

C. McNaught and P. Lam. Using Wordle as a supplementary research tool. *The qualitative report*, 15(3):630–643, 2010.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, pages 1436–1462, 2006.

J. C. Milton, V. N. Shankar, and F. L. Mannering. Highway accident severities and the mixed logit model: An exploratory empirical analysis. *Accident Analysis and Prevention*, 40:260–266, 2008.

D. Mimno and D. Blei. Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 227–237, 2011.

D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, 2011.

A. M'Kendrick. Applications of mathematics to medical problems. *Proceedings of the Edinburgh Mathematical Society*, 44:98–130, 1925.

R. Nallapati, A. Ahmed, W. Cohen, and E. Xing. Sparse word graphs: A scalable algorithm for capturing word correlations in topic models. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 343–348, 2007.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 215–224, 2010.

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

A. K. Nikoloulopoulos. Copula-based models for multivariate discrete response data. In P. Jaworski, F. Durante, and W. Härdle, editors, *Copulae in Mathematical and Quantitative Finance*, number July. 2013a.

A. K. Nikoloulopoulos. On the estimation of normal copula discrete regression models using the continuous extension and simulated likelihood. *Journal of Statistical Planning and Inference*, 143(11):1923–1937, 2013b.

A. K. Nikoloulopoulos. Efficient estimation of high-dimensional multivariate normal copula models with discrete spatial responses. *Stochastic Environmental Research and Risk Assessment*, 30(2):493–505, 2016.

A. K. Nikoloulopoulos and D. Karlis. Modeling multivariate count data using copulas. *Communications in Statistics-Simulation and Computation*, 39(1):172–187, 2009.

S. Nikolova, J. Boyd-Graber, C. Fellbaum, and P. R. Cook. Better vocabularies for assistive communication aids: Connecting terms using semantic networks and untrained annotators. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 171–178, 2009.

A. Noack. Modularity clustering is force-directed layout. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 79(2):1–8, 2009.

A. Panagiotelis, C. Czado, and H. Joe. Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499):1063–1072, 2012.

E. Park and D. Lord. Multivariate Poisson-lognormal models for jointly modeling crash frequency by severity. *Transportation Research Record: Journal of the Transportation Research Board*, (2019):1–6, 2007.

R. A. Parsa and S. a. Klugman. Copula regression. *Variance: Advancing theScience of Risk*, 5(1):45–54, 2011.

Q. Pleple. Interactive topic modeling. In *ACL*, pages 248–257, 2013.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, jun 2000.

P. Ravikumar, M. Wainwright, and J. Lafferty. High-dimensional Ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, jun 2010.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

J. Reisinger, A. Waters, and R. J. Mooney. Spherical topic models. In *Graphical Models*, volume 42, pages 1–4, 2010.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7(1):110–120, 1997. ISSN 10505164. doi: 10.1214/aoap/1034625254.

M. Roth. On the multivariate t distribution. Technical report, Linköpings universitet, Linköping, Sweden, 2013.

L. Rüschendorf. Copulas, sklar's theorem, and distributional transform. In *Mathematical Risk Analysis*, chapter 1, pages 3–34. Springer-Verlag Berlin Heidelberg, 2013.

R. Salakhutdinov and G. Hinton. Replicated softmax: An undirected topic model. *NIPS*, 22:1607–1614, 2009.

V. Schmitz. *Copulas and Stochastic Processes*. 2003.

G. Sharma and C. E. Rodríguez-Pardo. The dark side of cielab. *Proc. SPIE 8292, Color Imaging XVII: Displaying, Processing, Hardcopy, and Applications*, 8292:82920D, 2012.

A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231, 1959.

A. Sklar. Random variables, joint distribution functions, and copulas. *Kybernetika*, 9(6): 449–460, 1973.

J. B. Slapin and S.-O. Proksch. A scaling model for estimating time-serial positions from texts. *American Journal of Political Science*, 52(3):705–722, 2008.

R. Srivastava and A. Srivastava. On a characterization of Poisson distribution. *Journal of Applied Probability*, 7(2):497–501, 1970.

K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, 2012.

H. Steyn. On the multivariate Poisson normal distribution. *Journal of the American Statistical Association*, 71(353):233–236, 1976.

M. Steyvers and T. Griffiths. Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2007.

M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.

W. Tansey, O. H. M. Padilla, A. S. Suggala, and P. Ravikumar. Vector-space Markov random fields via exponential families. In *ICML*, 2015.

Y. W. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1556–1581, dec 2006.

H. Teicher. On the multivariate Poisson distribution. *Scandinavian Actuarial Journal*, 1954 (1):1–9, 1954.

P. K. Trivedi and D. M. Zimmer. *Copula Modeling: An Introduction for Practitioners*, volume 1. 2005.

P. Tsiamyrtzis and D. Karlis. Strategies for efficient computation of multivariate Poisson probabilities. *Communications in Statistics-Simulation and Computation*, 33(2):271–292, 2004.

C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21:5—-42, 2011.

F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with Wordle. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1137–1144, 2009.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(12):1–305, 2008.

Y.-W. Wan, G. I. Allen, and Z. Liu. Tcga2stat: simple tcga data access for integrated statistical analysis in r. *Bioinformatics*, 32(6):952–954, 2016.

J. Wang, J. Zhao, S. Guo, C. North, and N. Ramakrishnan. ReCloud: Semantics-based word cloud visualization of user reviews. In *Proceedings of the 2014 Graphics Interface Conference*, pages 151–158. Canadian Information Processing Society, 2014.

W. Y. Wang and Z. Hua. A semiparametric Gaussian copula regression model for predicting financial risks from earnings calls. *ACL*, pages 1155–1165, 2014.

Y. Wang. Characterizations of certain multivariate distributions. *Mathematical Proceedings of the Cambridge Philosophical Society*, 75(02):219–234, 1974.

S. D. Wicksell. *Some Theorems in the Theory of Probability, with Special Reference to Their Importance in the Theory of Homograde Correlation...* 1916.

Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma. Semantic-preserving word clouds by seam carving. In *Computer Graphics Forum*, volume 30, pages 741–750. Wiley Online Library, 2011.

P. Xue-Kun Song. Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000.

I. Yahav and G. Shmueli. On generating multivariate Poisson data in management science applications. *Applied Stochastic Models in Business and Industry*, 28(1):91–102, 2012.

E. Yang, P. Ravikumar, G. Allen, and Z. Liu. Graphical models via generalized linear models. In *NIPS*, pages 1358–1366, 2012.

E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. On Poisson graphical models. In *NIPS*, pages 1718–1726, 2013.

E. Yang, Y. Baker, P. Ravikumar, G. I. Allen, and Z. Liu. Mixed graphical models via exponential families. In *AISTATS*, pages 1042–1050, 2014a.

E. Yang, A. C. Lozano, and P. Ravikumar. Elementary estimators for graphical models. In *NIPS*, 2014b.

E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. Graphical models via univariate exponential family distributions. *JMLR*, 16:3813–3847, 2015.

H. F. Yu, F. L. Huang, and C. J. Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, nov 2011.

A. Zeileis and K. Hornik. Choosing color palettes for statistical graphics. *Research Report Series/Department of Statistics and Mathematics, Wien, Wirtschaftsuniv*, 41(October): 34, 2006.

X. Zhan, H. M. Abdul Aziz, and S. V. Ukkusuri. An efficient parallel sampling technique for multivariate Poisson-lognormal model: Analysis with two crash count datasets. *Analytic Methods in Accident Research*, 8:45–60, 2015.

J. Zhao and D. Meng. Fastmmd: Ensemble of circular discrepancy for efficient two-sample test. *Neural computation*, 27:1345–1372, 2015.