The Dissertation Committee for Jessica Lowell Henderson
certifies that this is the approved version of the following dissertation:

Learning and Validating Clinically Meaningful Phenotypes from Electronic Health Data

Committee:

Joydeep Ghosh, Co-Supervisor

William H. Press, Co-Supervisor

Peter Mueller

Robert van de Geijn

David Paydarfar

Learning and Validating Clinically Meaningful Phenotypes from Electronic Health Data

by

Jessica Lowell Henderson

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

August 2018

Dedicated to my husband Justin, my family, and the rest of my incredible support network.

Acknowledgments

I would like to thank my advisors, Joydeep Ghosh and Bill Press, for the guidance and support they have provided me throughout my graduate career. Thank you, Dr. Press, for agreeing to be my core advisor in ICES and for being my advocate within the program. You helped me carve out my place in the CSEM program, and I will always be grateful. Thank you, Dr. Ghosh, for taking a chance on me as a grad student. I had wandered around quite a bit before finding your lab, but you gave me a place and the opportunity to tackle a problem I found fascinating and rewarding to work on. Thank you for all of your guidance and for your patience with me. I have learned to think about the big picture from you and how that should shape the day-to-day life of a researcher. Additionally, thank you to my committee members, Dr. Peter Mueller, Dr. Robert van de Geijn, and Dr. David Paydarfar for encouraging me. Your probing questions have helped me strengthen my work. Furthermore, I would like to acknowledge the financial support I have received from NSF grant SCH 1417697 that has enabled me to pursue the research contained in this dissertation.

I would also like to thank all of my collaborators. To Joyce Ho, thank you for introducing me to tensors and the wonderful world of computational phenotyping. I have had so much fun collaborating with you, and I have learned so much. Thank you to Abel Kho and Josh Denny for providing guidance on the clinical side of my work and to Bradley Malin and Jimeng Sun for giving excellent feedback on ideas and last-minute drafts. Thank you to the students who let me bounce ideas off of them or who put up with me when I hit walls along the way, Shalmali Joshi, Avradeep Bhowmilk, and Rajiv Khanna. Thank you to Woody Austin for always being willing to think about tensors with me.

Finally, to the family and friends who make up my support network, thank you for your encouragement, humor, and unconditional love and belief that I could do this. To my partner and fellow adventurer who has been there every step of the way, Justin Henderson, thank you for always being the one in the hat.

Learning and Validating Clinically Meaningful Phenotypes from Electronic Health Data

Publication No. ____________________

Jessica Lowell Henderson, Ph.D.
The University of Texas at Austin, 2018

Supervisors: Joydeep Ghosh
William H. Press

The ever-growing adoption of electronic health records (EHR) to record patients' health journeys has resulted in vast amounts of heterogeneous, complex, and unwieldy information [Hripcsak and Albers, 2013]. Distilling this raw data into clinical insights presents great opportunities and challenges for the research and medical communities. One approach to this distillation is called computational phenotyping. Computational phenotyping is the process of extracting clinically relevant and interesting characteristics from a set of clinical documentation, such as that which is recorded in electronic health records (EHRs). Clinicians can use computational phenotyping, which can be viewed as a form of dimensionality reduction where a set of phenotypes form a latent space, to reason about populations, identify patients for randomized case-control studies, and extrapolate patient disease trajectories. In recent years, high-throughput computational approaches have made strides in extracting potentially clinically interesting phenotypes from data contained in EHR systems.

Tensor factorization methods have shown particular promise in deriving phenotypes. However, phenotyping methods via tensor factorization have the following weaknesses: 1) the extracted phenotypes can lack diversity, which makes them more difficult for clinicians to reason about and utilize in practice, 2) many of the tensor factorization methods are unsupervised and do not utilize side information that may be available about the population or about the relationships between the clinical characteristics in the data (e.g., diagnoses and medications), and 3) validating the clinical relevance of the extracted phenotypes requires domain training and expertise. This dissertation addresses all three of these limitations. First, we present tensor factorization methods that discover sparse and concise phenotypes in unsupervised, supervised, and semi-supervised settings. Second, via two tools we built, we show how to leverage domain expertise in the form of publicly available medical articles to evaluate the clinical validity of the discovered phenotypes. Third, we combine tensor factorization and the phenotype validation tools to guide the discovery process to more clinically relevant phenotypes.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Increasingly, health service providers record their interactions with patients in electronic health record (EHR) systems. The narrative of a patient's health through time, as told through lab results, medication and diagnosis codes, and clinical notes, unwinds alongside other patients' stories in these vast EHRs systems. By combining, transforming, and extracting the key features of these parallel narratives, researchers and clinicians have the potential to make a difference in the individual stories of their patients. However, there are several challenges to transforming the unwieldy information contained in EHR systems into actionable insights. For one, EHR systems are heterogeneous, both in terms of the types of information they contain (e.g., continuous, natural language, count) and when compared to one another (i.e., EHRs at different facilities can differ a good deal). Additionally, the data contained within them are incomplete (sometimes not randomly), noisy, vast, and complex [Hripcsak and Albers, 2013]. Despite these challenges, researchers have shown that analysis of datasets extracted from EHR systems can shed light on clinical questions and help improve patient care [Jensen et al., 2012].

One approach to distilling the information contained in EHR databases into actionable insights is to construct phenotypes based on the EHRs of groups of patients. A computational, EHR-based phenotype is a set of algorithmically derived characteristics extracted from an EHR system that defines a clinically interesting set of patients [Hripcsak and Albers, 2013; Richesson et al., 2016]. Examples of computational phenotypes can be seen in Figure 3.2. Once derived, these phenotypes can help clinicians reason about the populations they serve and also help identify patients for case and controls for randomized controlled trials.

Traditionally, constructing computational phenotypes has been an iterative process performed by panels of domain experts. This approach is time-consuming and only produces one phenotype at a time. Recently, machine learning researchers have shown computational methods can be used to extract clinically relevant phenotypes from EHR databases in a high-throughput, automatic manner. From a clinician's point of view, automatically-extracted phenotypes must fit the following requirements: 1) the extracted phenotypes must be concise and different from one another, and 2) the phenotypes must be clinically relevant and interesting. The focus of this dissertation is to develop machine learning tools centered around generating clinically relevant phenotypes from information captured in electronic health records and to build means to validate the extracted phenotypes automatically. To this end, we have formulated 1) unsupervised, semi-supervised, and supervised phenotyping algorithms and 2) two phenotype validation frameworks.

1.1 Novel Phenotyping Algorithms

We use tensor factorization to automatically derive phenotypes in a high-throughput manner. We develop unsupervised, supervised, and semi-supervised tensor factorization models that result in interpretative and discriminative factors. Tensors, which are generalizations of vectors and matrices to higher dimensions, are ideal for capturing the multidimensional relationships inherent in EHR count and continuous data [Kolda and Bader, 2009]. For example, two patients may receive the same medication to treat different disorders, which is information that can be stored easily in a tensor. We build tensors from patient-level diagnosis, medication, and procedure codes and then use CANDECOMP/PARAFAC (CP) decomposition to factor the tensor. CP decomposition is a generalization of Singular Value Decomposition (SVD) with some important caveats. Whereas SVD on a matrix results in a decomposition of rank-one matrices, CP decomposition expresses a tensor as the sum of rank-one tensors (Figure 2.1 shows a cartoon of the tensor factorization process). In our clinical application, each rank-one tensor can be interpreted as a potential phenotype.

Pioneering work by Ho et al. [2014a] in 2014 demonstrated that phenotyping via tensor factorization results in a large number of phenotypes that a panel of domain experts judged to be clinically relevant and useful. While Ho et al. [2014a]'s method resulted in sparse factors, this sparsity was introduced through manual thresholding after the factorization had been performed. In later work, Ho et al. [2014b] introduced a factorization formulation that automatically resulted in sparse features, but clinicians critiqued the derived phenotypes were too similar to one another.

We formulated the following three categories of tensor factorization-based methods to extract candidate phenotypes from healthcare datasets:

- unsupervised and diversity-encouraging,
- semi-supervised incorporating insights from a proxy for domain knowledge, and
- supervised or semi-supervised using knowledge of patient disease status.

The first type of tensor factorization method is an unsupervised method that can be applied to general populations in order to understand overall characteristics and overriding groups within patient populations. The model that falls under this category, Granite [Henderson et al., 2017c], was developed based on clinician's critiques that previous phenotyping tensor factorization models were not producing phenotypes that were concise and diverse enough. Granite incorporates similarity penalties to encourage diversity and uses simplex projection to induce sparsity. Chapter 3 describes Granite's formulation and shows potential as a phenotype extraction tool on both simulated data and real EHR data.

In Granite, we observed experimentally that the clinical relevance of the phenotypes degraded with the demand for diversity. To increase the number of clinically relevant phenotypes, the second category of algorithm uses auxiliary information as a proxy for domain expertise to guide the tensor factorization process to more clinically relevant phenotypes. To remove the need for domain expert-provided supervision, we used information about the relationships between medications and diagnoses provided by a phenotype validation tool (summarized in Section 1.2). Specifically, we encode information about diagnoses and medications into a cannot-link constraint matrix and incorporate it into the Granite framework. This model, called PIVETed-Granite [Henderson et al., 2018c], shows potential for extracting sparse, diverse, and interpretable phenotypes. Additionally, we present a model that generalizes PIVETed-Granite to situations where cannot-link constraints can be useful but the side information is not as trusted. This model, called CP tensor decomposition with Cannot-Link Inter-mode Constraints (CP-CLIC) [Henderson et al., 2018d], leverages the information learned during the decomposition process to propose cannot-link constraints and then rejects or accepts them based on evidence from the auxiliary information. We formulate CP-CLIC for a family of loss functions and show its potential use on EHR and simulated data. While there has been some work done to incorporate semi-supervision and supervision into tensor factorization, those methods either assume knowledge about all modes of the tensor or require domain expertise.

The final category of phenotyping factorization models uses information about patient disease status in supervised and semi-supervised ways in the decomposition process to discover phenotypes that are descriptive of those diseases. The first model in this category, Greedy Angular Multiway Array

Iterative Decomposition (gamAID) [Henderson et al., 2017b], is a supervised model that focuses on populations of patients with a specified disease who are at risk for developing other diseases. We show gamAID's formulation and results on a publicly available electronic health record dataset. The second model in this category, Phenotyping through Semi-Supervised Tensor Factorization (PSST) [Henderson et al., 2018a], constructs semi-supervised constraints using patient disease status of a subset of patients in a population to encourage phenotype class membership that is limited to patients with the condition. We show PSST's formulation and analyze the phenotypes resulting from applying PSST to an EHR-tensor constructed from de-identified patient-level data.

1.2 Computational Validation of Phenotypes

Once automatic, high-throughput methods have extracted a set of potential phenotypes, their clinical relevance must be validated. Volunteer domain experts annotate the candidate phenotypes as clinically relevant or not, but this task can be vague, time-consuming, and subject to the personal experiences of the domain expert. To aid in the candidate phenotype validation step, we built a framework called PheKnow–Cloud [Henderson et al., 2017a] and subsequently refined the framework with a tool called Phenotype Instance Verification and Evaluation Tool (PIVET) [Henderson et al., 2018a]. These tools generate clinical relevance evidence sets for candidate phenotypes based on the analysis of a publicly available corpus of medical articles. We present these frameworks and discuss the key differences between them. We show the potential of using this approach to help clinicians and researchers assess the clinical relevance of proposed phenotypes. In particular, we show how to incorporate the insights provided by PIVET into the tensor factorization process to increase the number of clinically meaningful phenotypes.

The rest of the dissertation is organized as follows. Chapter 2 covers the necessary mathematical background and work related to computational phenotyping and constrained tensor factorization. Chapter 3 presents Granite and shows how diversity and sparsity constraints can be included in the tensor factorization formulation to produce interesting, different, and succinct phenotypes. Chapter 4 investigates situations where we have information about the disease status of patients with two models, gamAID (Section 4.1) and PSST (Section 4.2). Chapter 5 discusses the phenotype validation tools, the prototype tool PheKnow–Cloud (Section 5.1) and the next-generation tool PIVET (Section 5.2). Chapter 6 shows, with the model PIVETed-Granite, how to merge tensor factorization with a phenotype validation tool via semi-supervised cannot-link constraints to guide the decomposition process to potentially more clinically meaningful phenotypes than unsupervised methods (Section 6.1). Finally, it also describes how to adapt the PIVETed-Granite formulation to situations and applications where the auxiliary information may be noisy (Section 6.2).

Chapter 2

Background

2.1 Mathematical Background

2.1.1 Tensors

A tensor is a generalization of a matrix to a multidimensional array. Each element of a tensor represents an n-way interaction. The number of dimensions, which are also called modes, is the order of a tensor (e.g., a third order tensor could capture the relationship between a document, term, and author). Vectors are tensors of order one, matrices are tensors of order two, and an n-order tensor has n dimensions. In this dissertation, we primarily consider tensors of order three, where the 3-way interaction is between patients, diagnoses, and medications or patients, diagnoses, and procedures. Tensors can be decomposed into a product of matrices or a combination of matrices and smaller tensors. Tensor factorization utilizes information in the multiway structure to produce factors that are concise and potentially more interpretable than the raw input data. Additionally, tensor factorization can identify components, even with relatively small amounts of observations [Kolda and Bader, 2009].

We use bold-faced lowercase letters to indicate vectors (e.g., $\boldsymbol{a}$, where

a_i is the ith entry of $\boldsymbol{a}$), bold-faced uppercase letters to indicate matrices (e.g., $\mathbf{A}$, where a_{ij} is the i,jth element of $\mathbf{A}$), and bold-faced script letters to indicate tensors with dimension greater than two (e.g., $\boldsymbol{\mathcal{X}}$ where $x_{\vec{i}}$ is the tensor element with index $\vec{i}$). The nth matrix in a series of matrices is denoted with a superscript integer in parentheses (e.g., $\boldsymbol{A}^{(n)}$ is the nth matrix in a series of matrices).

A colon (:) is used to denote a dimension of a tensor that is held fixed ($\mathbf{A}_{:j}$ denotes the jth column of matrix $\mathbf{A}$). A fiber of a tensor is formed when all elements of the tensor are fixed but one. In a third order tensor, $\boldsymbol{\mathcal{X}}$, examples of fibers are $x_{i:j}, x_{:jk}, x_{ij:}$. Arranging the fibers into the columns of a matrix is called the matricization of $\boldsymbol{\mathcal{X}}$ and is denoted $\mathbf{X}_{(n)}$, where n denotes which mode is being held fixed.

The definition for the algebraic operations used in this dissertation are provided below.

Definition 1 *The Khatri-Rao product of two matrices $\mathbf{A} \odot \mathbf{B}$ of sizes $I_A \times R$ and $I_B \times R$ respectively, produces a matrix $\mathbf{Z}$ of size $I_A I_B \times R$ such that $Z = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \cdots & \mathbf{a}_R \otimes \mathbf{b}_R \end{bmatrix}$, where $\otimes$ represents the Kronecker product.*

Definition 2 *The Kronecker product of two vectors $\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \mathbf{b} & a_2 \mathbf{b} \cdots a_{I_A} \mathbf{b} \end{bmatrix}^T$*

Definition 3 *The element-wise multiplication (and division) of two same-sized matrices $\mathbf{A} * \mathbf{B}$ ($\mathbf{A} \oslash \mathbf{B}$) produces a matrix $\mathbf{Z}$ of the same size such that the element $c_{\vec{i}} = a_{\vec{i}} b_{\vec{i}}$ ($c_{\vec{i}} = a_{\vec{i}} / b_{\vec{i}}$) for all $\vec{i}$.*

Definition 4 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ *is an N-way rank one tensor if it can be expressed as the outer product of N vectors,* $\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$*, where each element* $x_{\vec{i}} = x_{i_1, i_2, \cdots, i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$.

2.1.2 Tensor Decompositions

Many tensor decomposition models exist and a complete review of all the techniques is beyond the scope of this proposal. My work uses the CANDECOMP / PARAFAC (CP) decomposition [Carroll and Chang, 1970; Harshman, 1970], a common tensor factorization model. CP decomposition factorizes the original tensor $\mathcal{X}$ as a sum of R rank-one tensors and can be expressed as follows (see Figure 2.1):

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}^{(1)}_r \circ \ldots \circ \mathbf{a}^{(N)}_r = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \ldots; \mathbf{A}^{(N)}]\!] \tag{2.1}$$

The latter representation is shorthand notation with the weight vector $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_R]$ and the factor matrix $\mathbf{A}^{(n)} = [\mathbf{a}^{(n)}_1 \cdots \mathbf{a}^{(n)}_R]$.

Definition 5 *The* rank *of a tensor* $\mathcal{X}$ *is the smallest number of rank-one tensors that can be summed to equal* $\mathcal{X}$.

Finding the rank of a tensor with an order greater than 2 is NP-hard [Kolda and Bader, 2009]. It is common to choose the rank R of a decomposition through a grid search over possible values and setting R to be the rank that results the smallest objective function. Unlike matrices, the decompositions of tensors with order greater than two are unique (up to scaling and permutation).

Standard CP decomposition is formulated as a least squares approximation, called CP alternating least squares (CP-ALS). In CP-ALS, the data are assumed to follow a Gaussian distribution, which makes it well-suited for continuous data [Kolda and Bader, 2009]. This assumption also results in simpler algorithms, and the Alternating Direction Method of Multipliers (ADMM) technique can be readily applied for distributed computation. However, since the kind of EHR data considered in this work is based on counts, a better match is the nonnegative CP alternating Poisson regression (CP-APR) model developed by Chi and Kolda [2012], wherein the objective is to minimize the KL divergence (i.e., data follows Poisson distribution).

When computing a decomposition, it can be handy to use the following identities of matricization:

$$[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]_{(n)} = \boldsymbol{\lambda}\mathbf{A}^{(n)}(\mathbf{A}^{(-n)})^{\intercal}$$

where

$$\mathbf{A}^{(-n)} \equiv \mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}$$

2.1.3 Notational Conveniences in Tensor Computations

All of the algorithms for minimizing the objective functions, f, between the observed values, $\mathbf{x}$, and model parameters, $\mathbf{z}$, detailed in this dissertation use some form of gradient descent. Here we introduce some notational conveniences to aid in the gradient descent process.

The objective function, f, can be represented as a scalar-valued function of the parameter vector $\mathbf{y}$ [Acar et al., 2011a], where $\mathbf{y}$ represents either the vectorization of the factor matrices or the weights.

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix} \mathrm{vec}(\lambda\mathbf{A}^{(1)} & \sigma\mathbf{u}^{(1)}) \\ \mathrm{vec}(\mathbf{A}^{(2)} & \mathbf{u}^{(2)}) \\ \vdots & \\ \mathrm{vec}(\mathbf{A}^{(N)} & \mathbf{u}^{(N)}) \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{vec}(\hat{\mathbf{A}}^{(1)}) & \cdots & \mathrm{vec}(\hat{\mathbf{A}}^{(N)}) \end{bmatrix}^{\mathsf{T}}\end{aligned}$$

Then, the gradients of the objective function f can be formed by vectorizing the partial derivatives with respect to each component of the parameter vector $\mathbf{y}$:

$$\nabla f(\mathbf{y}) = \begin{bmatrix} \mathrm{vec}\left(\frac{\partial f}{\partial \hat{\mathbf{A}}^{(1)}}\right) & \cdots & \mathrm{vec}\left(\frac{\partial f}{\partial \hat{\mathbf{A}}^{(N)}}\right) \end{bmatrix}^{\mathsf{T}}$$

For notation purposes, we can represent the matricized form of the tensor decomposition as:

$$[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]_{(n)} = \boldsymbol{\lambda}\mathbf{A}^{(n)}(\mathbf{A}^{(-n)})^{\mathsf{T}}$$

where

$$\mathbf{A}^{(-n)} \equiv \mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}$$

It is useful to note that each element in the approximation tensor, $z_{\vec{i}}$, can be rewritten as follows:

$$\begin{aligned} z_{\vec{i}} &= \sigma u_{i_1}^{(1)} u_{i_2}^{(2)} \cdots u_{i_N}^{(N)} + \sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)} \\ &= \sigma \left(\prod_{m \neq n} u_{i_m}^{(m)} \right) u_{i_n}^{(n)} + \sum_{r=1}^{R} \lambda_r \left(\prod_{m \neq n} a_{i_m r}^{(m)} \right) a_{i_n r}^{(n)} \end{aligned}$$

2.1.4 Bregman Divergences

Fitting a CP decomposition involves minimizing an objective function between the tensor $\mathfrak{X}$ and a model tensor $\mathfrak{Z}$. The objective function is usually chosen based on assumptions about the underlying distribution of the data and then augmented with constraints to deliver solutions of a desired form. Least squares approximation, the most popular formulation, assumes a Gaussian distribution and is well-suited for continuous data [Kolda and Bader, 2009]. For count data, it may be more appropriate to use nonnegative CP alternating Poisson regression (CP-APR) developed by Chi and Kolda [2012], wherein the objective is to minimize the KL divergence (i.e., data follows Poisson distribution). The least squares approximation and KL divergence are both examples of Bregman divergences, a generalized measure of distance Bregman [1967]. Other common Bregman divergences and their gradients are listed in Table 2.1. While this dissertation focuses primarily on loss functions that use KL divergence, in Chapter 6, we show a method that can be generalized to other loss functions.

Table 2.1: Bergman divergence loss functions and their derivatives.

Bregman Divergence	**Negative Log-Likelihood**	**Matricized Gradient (i.e., $\frac{\partial \mathcal{L}(\mathfrak{Z} \mid \mathfrak{X})}{\partial \mathbf{A}^{(n)}}$)**
Mean-squared	$\frac{1}{2}(x_{\vec{i}} - z_{\vec{i}})^2$	$(\mathbf{Z}_{(n)} - \mathbf{X}_{(n)})\mathbf{A}^{(-n)}$
Exponential	$x_{\vec{i}} z_{\vec{i}} - \log z_{\vec{i}}$	$(\mathbf{X}_{(n)} - \mathbb{1} \oslash \mathbf{Z}_{(n)})\mathbf{A}^{(-n)}$
Poisson	$z_{\vec{i}} - x_{\vec{i}} \log z_{\vec{i}}$	$(\mathbb{1} - \mathbf{X}_{(n)} \oslash \mathbf{Z}_{(n)})\mathbf{A}^{(-n)}$
Boolean	$\log(z_{\vec{i}} + 1) - x_{\vec{i}} \log z_{\vec{i}}$	$(\mathbb{1} \oslash (\mathbf{Z}_{(n)} + \mathbb{1}) - \mathbf{X}_{(n)} \oslash \mathbf{Z}_{(n)})\mathbf{A}^{(-n)}$

2.2 EHR-Based Phenotyping

In the past, domain experts have manually derived phenotypes, but this is a laborious, time–consuming process [Carroll et al., 2011; Chen et al., 2013; Hripcsak and Albers, 2013]. Recent efforts have focused on using machine learning techniques to automatically extract candidate phenotypes from sets of electronic health records with minimal supervision [Ho et al., 2014a,b; Hu et al., 2015; Wang et al., 2015; Yu et al., 2015].

Nonnegative tensor factorization (NNTF) on tensors constructed from EHR data is one way to perform high-throughput phenotyping. Tensors have the capability to capture complex relationships that exist in healthcare. Tensors can, for example, handle that one medication could be used to treat different diseases. For example, Metformin, commonly used to treat diabetes, has also shown promise in treating the symptoms of polycystic ovary syndrome [Lashen, 2010]. Figure 2.1 shows an example of the phenotyping process using NNTF. The input to the model is a tensor composed of three modes, patients, their diagnoses, and their medications. The output is a weighted sum of rank-one tensors. Each rank-one tensor is formed by taking the outer product of three factor vectors that are found by solving an optimization problem for each of the three modes. These factor vectors can be organized into factor matrices by mode, which is depicted in the lower part of Figure 2.1 (note: the weights, $\boldsymbol{\lambda}$, have been absorbed into the patient factor matrix). Factor matrices are a convenient way to keep track of the modes in a decomposition.

Using NNTF to perform high-throughput computational phenotyping was initially proposed through a method called *Limestone*, which showed that NNTF could computationally extract candidate phenotypes, a surprisingly large number of which were deemed clinically relevant by medical experts [Ho et al., 2014a]. Limestone obtains phenotypes by decomposing the EHR tensor using the CP-APR algorithm and post-processing the factors to remove probabilistically unlikely elements [Ho et al., 2014a]. However, one of Limestone's drawbacks is that it relies upon post-processing to create more sparsity in the phenotypes. A subsequent algorithm called *Marble* addressed this weakness in Limestone by directly adding a global offset tensor and employing a new inference method to encourage sparsity and stability in the phenotypes [Ho et al., 2014b]. Marble decomposed the EHR tensor into an interaction tensor (the sum of the first R rank-one tensors) and a bias tensor (the $(R+1)^{th}$ rank-one tensor). The bias or offset tensor is strictly positive, which makes it possible for terms in the other rank-one components to be zero thereby creating sparse factors. This bias tensor combined with a user-specified sparsity threshold and a projection step results in sparse factors. The factor matrices in Figure 2.1 come from a Marble decomposition of a patient × diagnosis × medication tensor (the first five phenotypes of this fit are shown in Figure

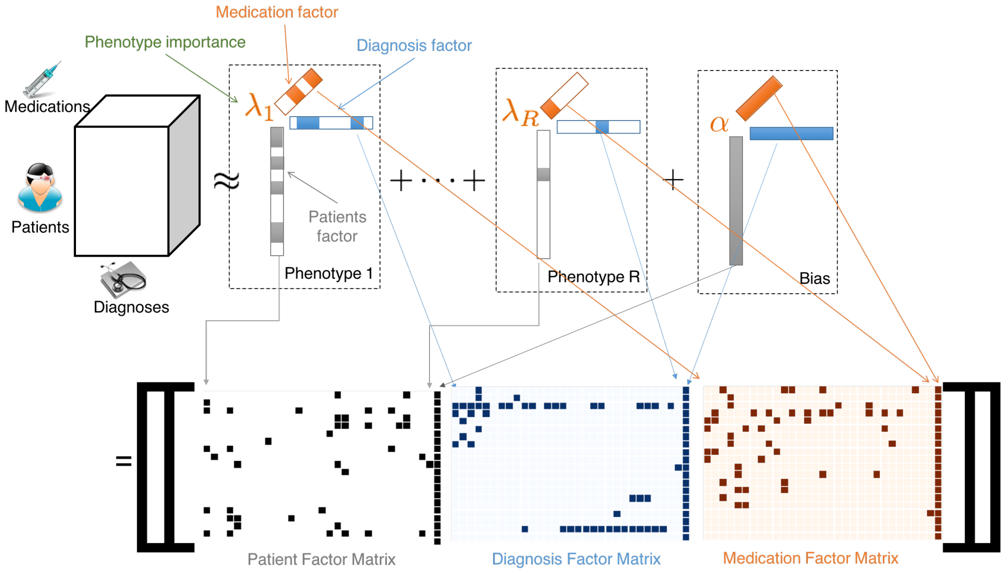


Figure 2.1: Overview of phenotyping via tensor decomposition process. A tensor is constructed of patient-level data is decomposed into the weighted sum of rank-one tensors based on the minimization of an objective function. Each rank-one tensor, formed by taking the outer product of factor vectors, constitutes a phenotype.

3.2). Summing across columns gives the number of phenotypes that contain a particular diagnosis, medication, or patient. For example, the third row of the diagnosis factor matrix is "Major Symptoms, Abnormalities," which appears in the majority of the phenotypes. Domain experts were critical of the fact that while Marble produces interpretable, concise phenotypes, there was too much similarity across phenotypes.

Several other NNTF models have been proposed to achieve automatic, high-throughput computational phenotyping [Perros et al., 2015; Wang et al., 2015; Hu et al., 2015]. Taking a different approach, Perros et al. [2015] introduced a sparse Hierarchical Tucker Factorization, which uses a network of tensors. The authors showed how it could be applied to extracting diagnosis phenotypes out of EHR data using the hierarchical structure in ICD-9 codes. Rubik imposes pairwise constraints on the vectors in the factor matrices, but these constraints result in solutions with near orthogonal vectors [Wang et al., 2015]. While this approach provides high-level insights into a patient population, it may smooth over more nuanced medical realities. Additionally, Rubik used constraints provided by domain experts. However, this approach may not always be feasible because domain expertise may not always be available in the tensor factorization. Hu et al. [2015] used a Bayesian NNTF approach to decompose an EHR count tensor. However, this model does not induce sparsity and diversity in those phenotypes.

High-throughput phenotyping has also been achieved with other machine learning and data mining techniques. Joshi et al. [2016] had success applying weakly supervised matrix factorization to clinical notes to generate phenotypes when the conditions were known a priori, while others have used matrix factorization on the micro (patient) and macro (population) to derive sparse phenotypes from longitudinal EHR data [Zhou et al., 2014]. Other methods have delivered insights using topic modeling approaches. Some topic modeling methods focused solely on diagnosis codes [Chen et al., 2015] and others on heterogeneous data (e.g., diagnosis, laboratory results, clinical notes) [Pivovarov et al., 2015]. Further investigations have applied deep learning to raw EHR data with success, but their methods require supervision, which is not always available in data sources or may be too restrictive for a phenotyping task [Henao et al., 2015; Che et al., 2015; Kale et al., 2015]. While the work above delivers insight into patient populations, only Zhou et al. [2014] focuses on creating concise phenotypes, and none of the above generate diverse phenotypes. For clinicians, diversity is important to discover rare phenotypes in a patient population as well as in features in predictive models. Moreover, diverse phenotypes are likely easier to implement, as a clinician may find it difficult to rank-order or apply phenotypes that have substantial overlap.

2.3 Constrained Tensor Decomposition Methods

2.3.1 Semi-supervised learning in tensor factorization

Semi-supervised learning (SSL) is a hybrid of supervised and unsupervised learning where there is a (small) portion of labeled data and unlabeled data. The assumption in SSL is that the unlabeled data provides information about the distribution of the examples that are useful. One class of approaches, transductive SSL, is useful in situations where we know something about the relationships between observations and wish to incorporate that information into the learning process [Sammut and Webb, 2010]. In particular, semi-supervised clustering introduces the notion that there are pairs of data points that must be clustered together, or *must-link*, and pairs that must not be clustered together in the same cluster, or *cannot-link*. While tensor factorization is similar to clustering, relatively few tensor decomposition methods incorporate semi-supervision. Peng introduced must-link and cannot-link constraints for the least squares objective function (data follows Gaussian distribution) [Peng, 2010]. Peng [2010] incorporated cannot-link and must-link constraints into a non-negative tensor factorization but only put the constraints on individual factor matrices and did not put constraints between the factor matrices.

Few tensor factorization methods incorporate between-mode constraints even in the non-medical domain. Davidson et al. [2013] used inter-mode constraints in supervised and semi-supervised ways to discover network structure in spatio-temporal fMRI datasets. However, their intermode constraints required domain expertise to construct. Narita et al. [2012] used within-mode and between-mode regularization terms to constrain similar objects to have similar factors in 3-mode tensors (i.e., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$). This method requires between-mode constraints on all of the modes, whereas in Chapter 6, we show how to construct cannot-link constraints for subsets of the modes, which makes our approach more flexible and adaptable to a variety of different situations. Additionally, for three modes, Narita et al. [2012]'s method requires the formation of $I_1 I_2 I_3 \times I_1 I_2 I_3$ matrix.

2.3.2 Constrained and supervised tensor factorization methods

Some CP tensor decomposition methods have included constraints in their fitting processes with the goal of tailoring the results to the needs of the applications in question. Carroll et al. [1980] used domain knowledge to put linear constraints on the factor matrices. Formulated for spatiotemporal datasets, CP-ORTHO requires orthogonality of the resulting factor vectors within each mode Afshar et al. [2017].

In the medical domain, a handful of tensor factorization methods have used domain-expert provided constraints or supervised to derive phenotypes. As mentioned in Section 2.2, Rubik introduced a combination of pairwise constraints on the vectors in the factor matrices and guidance matrices to improve the meaningfulness of the factors [Wang et al., 2015]. Rubik's guidance matrices, which encode information that is already known, attempt to induce classes that have minimal overlap by guiding the non-patient modes using domain knowledge. However, by focusing on guiding non-patient modes, Rubik's approach may leave out clinically interesting phenotypes. Furthermore, specifying a priori which elements should appear together may limit the amount of knowledge discovery possible in the decomposition process. Kim et al. [2017b] proposed a supervised tensor factorization method where patient outcome information guides the tensor decomposition to discover phenotypes that are good predictors of these outcomes for unseen patients as well as to generate distinct phenotypes. However, this work requires complete knowledge of the outcomes for each patient in the cohort. Furthermore, they use preprocessing methods to ensure all terms in a phenotype are similar and cohesive. Like the guidance provided in Rubik, this approach could smooth over novel phenotypes important to understanding a condition.

2.4 Phenotype Validation Background

In this section, we describe the background necessary for understanding the two phenotype validation tools we have developed. Both tools use text and co-occurrence analysis of a corpus from Pubmed. We describe the corpus, various analysis methods, and how others have utilized it for knowledge discovery in the medical domain.

2.4.1 PubMed

PubMed Central (PMC) is an online collection comprising over 3 million biomedical and biological articles gathered from thousands of journals [NCBI Resource Coordinators, 2018]. PMC is maintained and curated by the National Library of Medicine (NLM) at the US National Institute of Health[1].

In regard to phenotypes, researchers tend to use PubMed as an exploratory tool to discover new phenotypes rather than as a resource to validate candidate phenotypes. Boland et al orchestrated one of the few studies that used PubMed as a validation tool. They mined EHRs for patients with predefined disease codes and then compared the birth month and the disease of these patients with a group of control patients who did not have the disease codes present in their EHRs. They found a relationship between certain diseases and birth months in the case group [Boland et al., 2015]. They validated their results against papers retrieved from PubMed that mentioned disease and birth month. This study was novel in that it demonstrated PubMed could be utilized to provide feedback for and validation of results produced through automatic means.

[1] `https://www.ncbi.nlm.nih.gov/pmc/about/faq/`

More commonly, researchers use PubMed as tool to generate hypotheses and discover phenotypes and other biomedical issues [Ananiadou et al., 2006; Jensen et al., 2006]. Multiple software packages such as LitInspector [Frisch et al., 2009], PubMed.mineR [Rani et al., 2015], ALIBABA [Plake et al., 2006], as well as python packages such as Pymedtermino [Lamy et al., 2015] and Biopython [Cock et al., 2009] have been developed to help researchers extract and visualize PubMed. Other researchers have built tools to rank search results, discover topics and relationships within search results, visualize search results, and improve user interaction with PubMed [Lu, 2011].

2.4.2 Text Mining PubMed

Jensen et al. [2006] give a thorough overview of how PubMed can be harnessed for information extraction and entity recognition. Natural language processing (NLP) techniques form one approach to mining the literature. Some researchers have used NLP techniques on PubMed to discover disease-gene associations [Kim et al., 2017a], and others have used PubMed in concert with additional data sources to generate phenotypes [Alnazzawi et al., 2015]. Collier et al. [2015] used NLP techniques in conjunction with association rule mining to discover phenotypes using PubMed. However, none of these approaches have sought to use PubMed as a validation tool for data-driven phenotypes.

Co-occurrence analysis, which is what PheKnow-Cloud and PIVET are built on, is more widely used because it is simple to implement and interpret. Researchers have applied co-occurrence strategies to generate phenotypes. Some have performed co-occurrence analysis on PubMed to study links between diseases [Rajpal et al., 2014], which can be viewed as a simple type of phenotype discovery. Others have explored relationships between phenotypes and genotypes [Pletscher-Frankild et al., 2015; Xu et al., 2016]. In contrast to this work, our approach uses phenotypes as the starting point and performs co-occurrence analysis over the PMC corpus as a means of assessing their validity. We assume these phenotypes were induced over other sources (e.g., EHRs) and not from PMC. Co-occurrence analysis has the drawback of not being able to explicitly model the type of relationship that exists between two or more terms (e.g., negative or positive). However, we require the terms within a phenotype be positively related to one another, which aligns with the findings of publication bias research.

Publication bias is the tendency for the academic publishing ecosystem (e.g., researchers, reviewers, and editors) to submit and publish articles that show positive relationships between the entities being studied. The nonrandom omission of results that is not based on the quality of the methodology but on the direction of the results is a well-studied area of research and has been shown to have a negative effect on research in many cases [Hopewell et al., 2009; Dickersin, 1990; Song et al., 2010, 2013; Ekmekci, 2017; Dwan et al., 2008][19-24]. In general, publication bias introduces risks to researchers and to the general public to which research is applied (via policies and treatment decisions). However, in PheKnow-Cloud and PIVET, this bias is a strength rather than a drawback because the current focus of PheKnow-Cloud and

PIVET is on the presence of relationships within the user-supplied candidate phenotypes. Furthermore, as co-occurrence analysis does not attempt to infer information about the type of relationship or any causal information, the presence of publication bias allows for the assumption that when two phrases occur together, it may imply that a relationship exists [Dickersin, 1990; Easterbrook et al., 1991; Stern and Simes, 1997].

Chapter 3

Granite: Diverse, Sparse High-Throughput Phenotyping via Tensor Factorization

3.1 Introduction

This chapter describes Granite, a novel nonnegative tensor factorization model to fit count data, that produces diverse, sparse, and interpretable candidate phenotypes in an unsupervised manner [Henderson et al., 2017c]. Granite deviates from Marble [Ho et al., 2014b], a state-of-art model in 2014, in several key aspects: (i) it introduces a flexible penalized angular regularization term on the factors to promote diversity, (ii) it utilizes a simplex projection to calculate the factors and ℓ_2-regularization to achieve better sparsity control, and (iii) it develops an effective projected gradient descent-based approach to solve for the interaction and bias factors simultaneously. The penalized angular regularization term is flexible so users can encode different amounts of diversity in each mode. We illustrate the efficacy of our model on simulated data and real EHR data.

3.2 Problem Formulation

Let $\boldsymbol{\mathcal{X}}$ denote an $I_1 \times I_2 \times \cdots \times I_N$ tensor of count (nonnegative integer) data and $\boldsymbol{\mathcal{Z}}$ represent a same-sized tensor where each element $z_{\vec{i}}$ contains the optimal Poisson parameters of the observed tensor $x_{\vec{i}}$. The Granite optimization problem is defined as the following:

$$
\begin{aligned}
\min(f(\boldsymbol{\mathcal{X}})) \equiv \min(\sum_{\vec{i}}(z_{\vec{i}} - x_{\vec{i}} \log z_{\vec{i}}) && (3.1)\\
+\frac{\beta_1}{2}\underbrace{\sum_{n=1}^{N}\sum_{r=1}^{R}\sum_{p=1}^{r}(\max\{0, \frac{(\mathbf{a}_p^{(n)})^\intercal \mathbf{a}_r^{(n)}}{||\mathbf{a}_p^{(n)}||_2 ||\mathbf{a}_r^{(n)}||_2} - \theta_n\})^2}_{\text{angular regularization}} && (3.2)\\
+\frac{\beta_2}{2}\underbrace{\sum_{n=1}^{N}\sum_{r=1}^{R}||\mathbf{a}_r^{(n)}||_2^2)}_{\ell_2\text{regularization}} && (3.3)\\
\text{s.t } \boldsymbol{\mathcal{Z}} = [\![\sigma; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!] + [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!] && (3.4)\\
\sigma > 0, \lambda_r \geq 0,\ \forall r &&\\
\mathbf{A}^{(n)} \in [0,1]^{I_n \times R}, \mathbf{u}^{(n)} \in (0,1]^{I_n \times 1},\ \forall n &&\\
||\mathbf{a}_r^{(n)}||_1 = ||\mathbf{u}^{(n)}||_1 = 1,\ \forall n. && (3.5)
\end{aligned}
$$

Minimizing the objective function, f, in (3.1, 3.2, 3.3) results in the tensor $\boldsymbol{\mathcal{Z}}$. As shown in Equation (3.4), $\boldsymbol{\mathcal{Z}}$ consists of two terms: (i) rank-one bias tensor with positive weight and factor vectors, σ and $\mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(1)}$, and (ii) rank R interaction tensor with nonnegative weight vector and factor matrices, $\boldsymbol{\lambda}$ and $\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}$. The rank R interaction tensor is composed of the weighted sum of rank-one tensors. Each rank-one tensor is constructed from N stochastic vectors (elements sum to 1 and are nonnegative), which is consistent with the existing CP Poisson tensor decompositions. We now discuss key features of the Granite approach in more detail.

3.2.1 Promoting Intra-Phenotype Diversity

To encourage diversity between the rank-one tensors, Granite introduces a penalty term to the objective function, shown in Equation (3.2). The penalized angular regularization term reduces the occurrence of overlapping elements in the interaction factor matrices $\mathbf{A}^{(n)}$ by penalizing decompositions where the factor vectors are too correlated, measured by the cosine of the angle between the vectors. Two vectors that are orthogonal will yield a cosine similarity of 0, while two identical vectors will result in a 1. This penalty is adapted from [Acar et al., 2014], which introduced angular constraints to yield a structure-revealing data fusion model that is robust to overfactoring. However, our model relaxes the angular constraint and softly imposes diversity via the regularization penalty. This results in the flexibility to allow for overlapping phenotypes in the scenario where it truly exists.

It is also important to note that *only vectors whose cosine angle with other vectors are greater than θ_n are penalized.* Thus, our model does not necessarily encourage orthogonal factor components unless $\theta_n = 0$, which would result in the same constraints as in [Wang et al., 2015]. Since θ_n is specific to each mode, our model can impose different levels of diversity on each mode.

A user may want to focus on extracting a few, diverse diagnoses but be less concerned with the similarity between the vectors of the patient mode.

3.2.2 Promoting Inter-Phenotype Sparsity

Granite uses ℓ_2-regularization (see Equation 3.3) and simplex projection (see Section 3.3.1) to achieve sparsity. Experimentally, ℓ_2-regularization term encourages the terms in the factor matrix vectors to be small. In Granite, the terms are projected back into feasible space using simplex projection onto a ball of diameter s and then are ℓ_1 renormalized. Adjusting the size of parameter s determines the number of non-zero terms in the factor vectors. The ℓ_2-regularization term along with the simplex projection act like an Elastic Net [Zou and Hastie, 2005] regularization to drive terms in the interaction tensors to 0.

3.2.3 Capturing the Baseline

The bias tensor, carried over from the Marble framework, captures the general features of the tensor and provides the stability necessary for elements in the factor vectors to be driven to zero. The bias tensor encapsulates the general characteristics of a patient population while the R rank-one interaction tensors reflect the key features of subgroups of the patient population.

Algorithm 1: Detailed Granite algorithm

```
Data: $\mathcal{X}, R, \mathbf{s}, \boldsymbol{\theta}$
Result: $[\![\sigma; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!], [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]$
for $k = 1, 2, \cdots, K$ do
    # Update parameters $\hat{\mathbf{A}}^{(n)}$
    Calculate $\nabla f(\hat{\mathbf{A}}^{(n)})$ for $n = 2, \ldots, N$ using Eqs. (3.11, 3.12)
    #Simplex projection with s
    Update $\hat{\mathbf{A}}^{(n)}$ for $n = 2, \ldots, N$ with projected gradient descent
    line search and simplex projection
    # Update parameter $\hat{\mathbf{A}}^{(1)}$
    Calculate $\nabla f(\hat{\mathbf{A}}^{(1)})$ using Eqs. (3.11, 3.12)
    Update $\hat{\mathbf{A}}^{(1)}$ with gradient descent and nonnegative projection
    Eqs. (3.6, 3.7)
    # Standard stopping criteria
    if $||\mathbf{y}^{+} - \mathbf{y}||_F <$ convergenceTol then
        break
    end
end
```

3.3 Algorithm

The Granite algorithm minimizes the objective function f to solve for the bias and interaction factor matrices simultaneously through projected gradient descent. The approach is different than Marble. Specifically, Marble combines an alternating minimization approach, where each mode has a multiplicative update with a sequential unconstrained minimization approach. Not only have gradient descent approaches been shown to have faster convergence compared to the alternating minimization approach [Acar et al., 2011a], but the projected gradient step avoids the problem of zeroing out components too early in the multiplicative updates. Furthermore, solving for the bias and interaction terms simultaneously avoids a potential problem where subtracting the best rank-one approximation may actually increase the tensor rank [Stegeman and Comon, 2010]. We note that although the work of Hansen et al. [2015] obtained better speed and accuracy of CP decomposition of Poisson data using bound-constrained Newton methods, the angular regularization term results in complications for second-order optimization.

Granite combines the interaction and bias vectors for each factor matrix, such that for mode n, the combined factor matrix is $\hat{\mathbf{A}}^{(n)} = \begin{bmatrix}\mathbf{A}^{(n)} & \mathbf{u}^{(n)}\end{bmatrix}$. Our preliminary experiments showed that absorbing the weights, $\boldsymbol{\lambda}$ and σ, into one of the modes cut down on computation time as well as increased the stability of the results. Without loss of generality, the first mode is chosen to be $\hat{\mathbf{A}}^{(1)} = \left[\left(\lambda \mathbf{A}^{(1)}\right) \quad (\sigma \mathbf{u}^{(1)})\right]$.

3.3.1 Projection

Projected gradient descent is used to ensure the solution lies in the feasible space (i.e., non-negative or positive). For the first mode, $\mathbf{A}^{(1)}$ and $\mathbf{u}^{(1)}$, the projection function is simply the standard projection on the nonnegative and positive orthant respectively:

$$P_{\mathbf{A}}(\mathbf{A}^{(1)}) = \max\{0, \mathbf{a}_r^{(1)}\}, \tag{3.6}$$

$$P_{\mathbf{u}}(\mathbf{u}^{(1)}) = \max\{\epsilon, \mathbf{u}^{(1)}\}, \epsilon \text{ arbitrarily small and positive.} \tag{3.7}$$

Projection of the other bias vectors for the other modes occurs identically to Equation 3.7.

Algorithm 2: Projected Gradient Descent Line Search

```
$t = t_{\text{init}}$    # Initialize the step size
Calculate $\nabla f(\mathbf{y})$
$F_t(\mathbf{y}) = \frac{1}{t}(\mathbf{y} - P_\Omega(\mathbf{y} - t\nabla f(\mathbf{y}))$
# Perform line search to find a good step size
while $f(\mathbf{y} - tF_t(\mathbf{y})) > f(\mathbf{y})$ do
    $t = \hat{\beta}_{\text{line}} t$
    $F_t(\mathbf{y}) = \frac{1}{t}(\mathbf{y} - P_\Omega(\mathbf{y} - t\nabla f(\mathbf{y})))$
end
$\mathbf{y}^+ = P_\Omega(\mathbf{y} - t\nabla f(\mathbf{y}))$
```

Projection for the interaction factor components $\mathbf{a}_r^{(g)}$ other than the first mode uses the Euclidean projection onto the ℓ_1-ball of diameter s [Duchi et al., 2008], which is described by the following optimization problem:

$$\begin{aligned} &\min_a \frac{1}{2}||a - b||_2^2 \\ &\text{s.t.} \sum a_i = s, a_i \geq 0. \end{aligned} \tag{3.8}$$

When $s = 1$, this is projection onto the probabilistic (or canonical) simplex. However, Granite takes advantage of the properties of the simplex projection and decreases s to a number less than 1, which results in even more sparse solutions. The subsequent result is then renormalized to meet the stochastic constraints. The detailed Granite algorithm is presented in Algorithm 2. A greedy approach has been suggested for efficient sparse projections onto the simplex [Becker et al., 2013], but is not scalable for large dimensions.

In Algorithm 2, we select an appropriate step size, t, using backtracking line search by iteratively shrinking the step size by $\hat{\beta}_{\text{line}}$ to ensure the following condition is met:

$$\begin{aligned} f(\mathbf{y} - tF_t(\mathbf{y})) &> f(\mathbf{y}), \\ \text{where } F_t(\mathbf{y}) &= \frac{1}{t}(\mathbf{y} - P_\Omega(\mathbf{y} - t\nabla f(\mathbf{y}))). \end{aligned}$$

Note that Equation (3.8) is the projection function, $P_\Omega(\cdot)$, in Algorithm 2. Although computing the objective function can be expensive, this ensures that our algorithm converges to a local minimum based on the standard convergence analysis of the proximal gradient method.

3.3.2 Partial derivatives of the objective function

Using the notational conveniences introduced in Section 2.1.3, we derive the partial derivatives for the factor vectors and penalty terms.

We first compute the gradient for the angular regularization term. We denote the cosine similarity penalty between two vectors using the function $g(\mathbf{a}_r^{(n)}, \mathbf{a}_p^{(n)})$. For convenience, we drop the n, r terms and introduce $\mathbf{b} = \mathbf{a}_p^{(n)}$ for $p \neq r$ and let $g(\mathbf{a}, \mathbf{b})$ denote the cosine similarity between two vectors $\mathbf{a}$, $\mathbf{b}$, where $g(\mathbf{a}, \mathbf{b}) = (\frac{\mathbf{b}^\intercal \mathbf{a}}{||\mathbf{b}||_2 ||\mathbf{a}||_2} - \theta_n)$. The gradient for the angular term is then

$$\begin{aligned} \frac{\partial g(\mathbf{a}, \mathbf{b})}{\partial \mathbf{a}} &= \frac{\mathbf{b}||\mathbf{a}||_2^2 - < \mathbf{b}, \mathbf{a} > \mathbf{a}^\intercal}{||\mathbf{b}||_2 ||\mathbf{a}||_2^3} \\ \frac{\partial (\max\{0, g(\mathbf{a}, \mathbf{b})\})^2}{\partial \mathbf{a}} &= (\max\{0, g(\mathbf{a}, \mathbf{b}))\}) \frac{\partial g(\mathbf{a}, \mathbf{b})}{\partial \mathbf{a}} \end{aligned}$$

The partial of the KL divergence step with respect to $\mathbf{a}_r^{(n)}$ is straightforward:

$$\frac{\partial \sum (z_{\vec{i}} - x_{\vec{i}} \log z_{\vec{i}})}{\partial \mathbf{a}} = [1 - \mathbf{X}_{(n)} \oslash \mathbf{Z}_{(n)}]\mathbf{a}_r^{(-n)}$$

The partial derivatives with respect to the factor matrices are the following:

$$
\begin{aligned}
\frac{\partial f}{\partial \mathbf{a}_r^{(n)}} &= [1 - \mathbf{X}_{(n)} \oslash \mathbf{Z}_{(n)}]\mathbf{a}_r^{(-n)} && (3.9)\\
&+ \beta_1 \sum_{p \neq r} \left(\max\{0, g(\mathbf{a}_r^{(n)}, \mathbf{a}_p^{(n)})\}\right) \frac{\partial g(\mathbf{a}_r^{(n)}, \mathbf{a}_p^{(n)})}{\partial \mathbf{a}_r^{(n)}} && (3.10)\\
&+ \ \beta_2 \ \mathbf{a}_r^{(n)} && (3.11)\\
\frac{\partial f}{\partial \mathbf{u}^{(n)}} &= [1 - \mathbf{X}_{(n)} \oslash \mathbf{Z}_{(n)}]\mathbf{u}^{(-n)} && (3.12)
\end{aligned}
$$

3.3.3 Membership of Existing Factors

Granite also computes a membership vector for a new axis, where the other modes are fixed with the already learned factors. The membership vector is defined as the convex combination of existing tensor factors, where the r^{th} element denotes the probability the entity exhibits characteristics consistent with the r^{th} tensor factors. For example, new patients can be projected onto the computational phenotypes to obtain a *phenotype membership* vector where each element represents the probability the patient has the phenotype. It is important to note that the membership vector is not equivalent to the factor matrix because the *stochastic constraints are on the row* and not the column. The ability to take new patients and obtain their phenotype membership can be used in several ways. For one, predictive models can be trained on phenotypes associated with a subset of population and then applied to other subsets.

Algorithm 3: Membership Calculation

```
Randomly initialize $\mathbf{B}$
for $k = 1, 2, \cdots, k_{\max}$ do
    Calculate $\nabla f(\mathbf{B})$
    Update $\mathbf{B}^{+} = P_{\mathbf{B}}(\mathbf{B} - t\nabla f(\mathbf{B}))$
    if $|f(\mathbf{B}^{+}) - f(\mathbf{B})| <$ convergenceTol then
        break
    end
end
$\hat{\mathbf{A}}^{(1)} =$ normalize rows($\mathbf{B}$)
```

Without loss of generality, we assume the 1st mode is the new axis (e.g., patients). Given a new tensor $\tilde{\boldsymbol{\mathcal{X}}}$, we want to find $\tilde{\mathbf{A}}^{(1)}, \tilde{\mathbf{u}}^{(1)}$ that provides the best approximation given $\mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}$ are fixed. We observe that this is almost equivalent to gradient descent where the partial derivatives of the other factors are zero except that the membership vector is obtained by normalizing the entries of $\tilde{\mathbf{A}}^{(1)}$ across the row instead of the columns. To solve for the optimal $\tilde{\mathbf{A}}^{(1)}, \tilde{\mathbf{u}}^{(1)}$, the same projected gradient descent approach described in Section 3.3.1 is taken with the projection onto the nonnegative orthant and the angular and ℓ_2 regularization penalties set to zero (minimizing the KL divergence only). Once $\tilde{\mathbf{A}}^{(1)}$ is calculated, the rows are normalized to sum to 1. Algorithm 3 details the calculation of the membership vector, $\mathbf{B}$. We define $\mathbf{B}$ using the factor matrices with a vector $\boldsymbol{\alpha_m}$, where m is the number of dimensions in the first axis. $\mathbf{B} = \boldsymbol{\alpha_m}\mathbf{I}\begin{bmatrix}\tilde{\mathbf{A}}^{(1)} & \tilde{\mathbf{u}}^{(1)}\end{bmatrix}$.

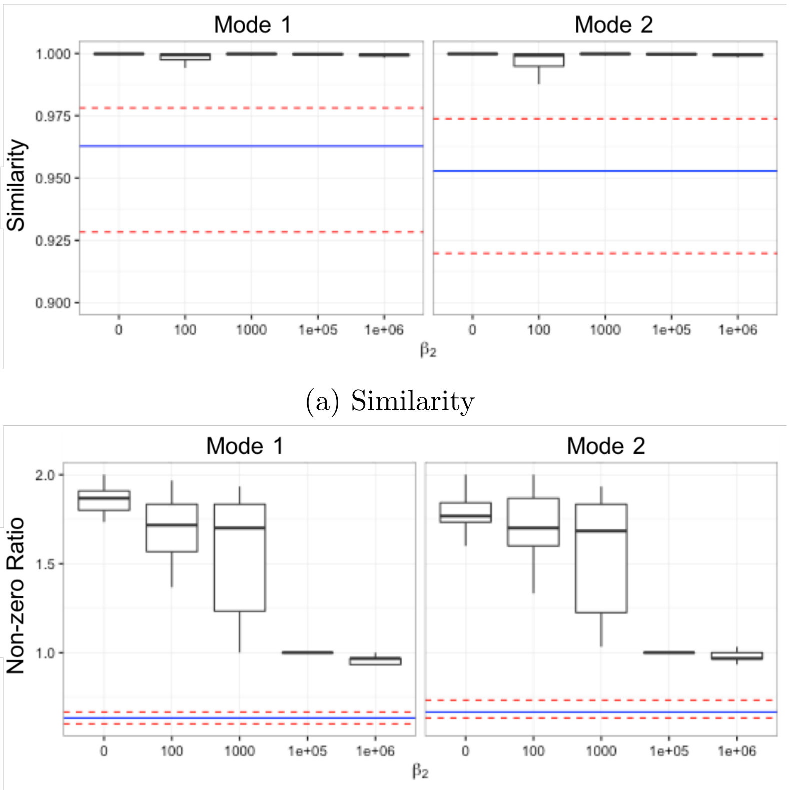
3.4 Experiments

3.4.1 Simulated Tensors

In this section, we evaluate Granite's performance against a simulated dataset where the actual tensor factors are known. This allows us to demonstrate the recovery properties of Granite in a controlled environment and explore the effects of algorithmic choices.

Specifically, we consider a third-order tensor of size $40 \times 20 \times 20$ with rank of 5 (i.e., $R = 5$). We generate the model $\mathbfcal{Z} = [\![\sigma; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!] + [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]$. Both the weights and bias factor vectors are straightforward, as the sampling occurs in the nonnegative and positive orthants respectively. We simulate the vectors in each interaction factor matrix $\mathbf{A}^{(n)}$ by sampling non-zero element indices according to a specified sparsity pattern. We then randomly sample along the simplex for the non-zero indices, rejecting vectors that are too similar to those already generated (i.e., their normalized cosine angle is greater than θ_n). Finally, each tensor element x_{ijk} is sampled from the Poisson distribution with the parameter set to z_{ijk}.

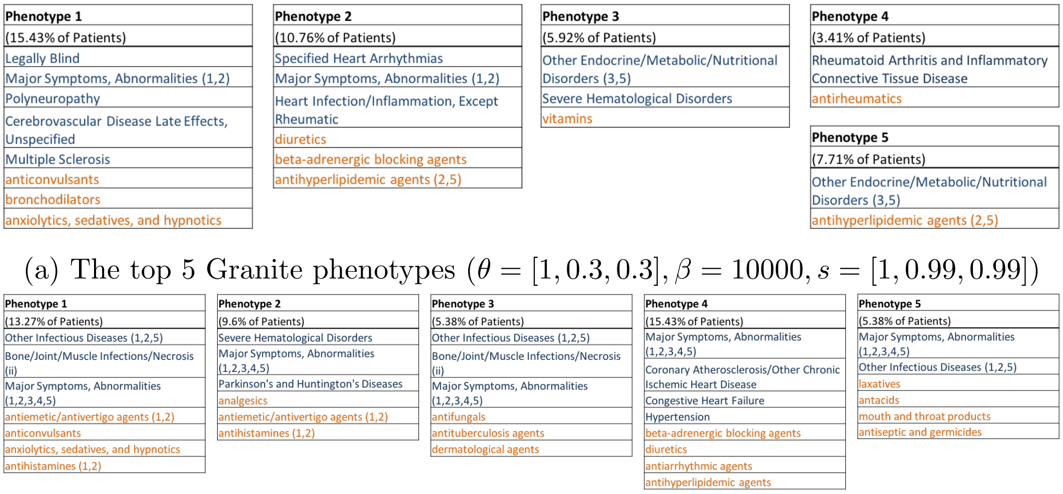
Our algorithm is evaluated on 50 simulated tensors where we set the cosine similarity to .3 and $\beta_1 = 1000$ and varied β_2 for each run. In addition, we fixed the sparsity parameter to project onto the simplex ($s = 1$) for the first mode and $s = .3$ for the second and third modes. The results are evaluated using 1) the non-zero ratio between the computed solution and the actual solution and 2) the cosine angle between vectors in the simulated and fit tensors. The cosine angle between the two vectors, a component of the factor



(a) Similarity

(b) Non-zero Ratio

Figure 3.1: Similarity (top) and non-zero ratio (bottom) between the fit latent factors, calculated by Granite and Marble, and the true latent factors for the second and third mode. The boxes represent Granite's performance, and the median and the the 25% and 75% percentiles of Marble's performance are designated by the blue and red dotted lines, respectively.



Phenotype 1
(15.43% of Patients)
Legally Blind
Major Symptoms, Abnormalities (1,2)
Polyneuropathy
Cerebrovascular Disease Late Effects, Unspecified
Multiple Sclerosis
anticonvulsants
bronchodilators
anxiolytics, sedatives, and hypnotics

Phenotype 2
(10.76% of Patients)
Specified Heart Arrhythmias
Major Symptoms, Abnormalities (1,2)
Heart Infection/Inflammation, Except Rheumatic
diuretics
beta-adrenergic blocking agents
antihyperlipidemic agents (2,5)

Phenotype 3
(5.92% of Patients)
Other Endocrine/Metabolic/Nutritional Disorders (3,5)
Severe Hematological Disorders
vitamins

Phenotype 4
(3.41% of Patients)
Rheumatoid Arthritis and Inflammatory Connective Tissue Disease
antirheumatics

Phenotype 5
(7.71% of Patients)
Other Endocrine/Metabolic/Nutritional Disorders (3,5)
antihyperlipidemic agents (2,5)

(a) The top 5 Granite phenotypes ($\theta = [1, 0.3, 0.3], \beta = 10000, s = [1, 0.99, 0.99]$)

Phenotype 1
(13.27% of Patients)
Other Infectious Diseases (1,2,5)
Bone/Joint/Muscle Infections/Necrosis (ii)
Major Symptoms, Abnormalities (1,2,3,4,5)
antiemetic/antivertigo agents (1,2)
anticonvulsants
anxiolytics, sedatives, and hypnotics
antihistamines (1,2)

Phenotype 2
(9.6% of Patients)
Severe Hematological Disorders
Major Symptoms, Abnormalities (1,2,3,4,5)
Parkinson's and Huntington's Diseases
analgesics
antiemetic/antivertigo agents (1,2)
antihistamines (1,2)

Phenotype 3
(5.38% of Patients)
Other Infectious Diseases (1,2,5)
Bone/Joint/Muscle Infections/Necrosis (ii)
Major Symptoms, Abnormalities (1,2,3,4,5)
antifungals
antituberculosis agents
dermatological agents

Phenotype 4
(15.43% of Patients)
Major Symptoms, Abnormalities (1,2,3,4,5)
Coronary Atherosclerosis/Other Chronic Ischemic Heart Disease
Congestive Heart Failure
Hypertension
beta-adrenergic blocking agents
diuretics
antiarrhythmic agents
antihyperlipidemic agents

Phenotype 5
(5.38% of Patients)
Major Symptoms, Abnormalities (1,2,3,4,5)
Other Infectious Diseases (1,2,5)
laxatives
antacids
mouth and throat products
antiseptic and germicides

(b) The top 5 Marble phenotypes ($\gamma = [0, 0.15, 0.15], \alpha = 10000$)

Figure 3.2: The Granite and Marble phenotypes with the highest weights (i.e., largerst λ_is) for R = 30

match score [Chi and Kolda, 2012], is used to quantify the similarity between the computed solution and the actual factor representation. We use the Hungarian method is to compute the optimal pairing between each approximated rank-one tensor and the corresponding "true" rank-one representation.

Figure 3.1a shows a boxplot of the similarity scores between the calculated latent factors and the true latent factors for the second and third mode, where the blue line represents the median performance of Marble and the red dotted lines are the 25% and 75% percentiles of Marble's similarity scores. Overall, Granite is able to recover the true latent representation better than Marble, with similarity scores above 0.95.

Figure 3.1b illustrates the non-zero ratio (i.e., (# of non-zeros in fitted factor vectors)/(# of non-zeros in actual factor vectors)) with the blue line denoting the median non-zero ratio of Marble. Granite's non-zero ratio improves as β_2 increases and the algorithm is able to recover the original sparsity pattern. While Marble's non-zero ratio is lower overall, it is below the original sparsity pattern and Granite outperforms Marble in terms of recovering the original tensor. Thus, Granite is able to capture the simulated latent factors while maintaining sparse solutions.

3.4.2 EHR-Count Tensor Experiments

3.4.2.1 Dataset Description

The Synthetic Derivative (SD) is a large, de-identified Electronic Medical Record (EMR) database at the Vanderbilt University Medical Center (VUMC) [Roden et al., 2008]. Among other pieces of patient information, the SD contains inpatient and outpatient billing codes and medication codes of nearly 2 million patients. In the work of Ritchie et al. [2010], domain experts manually developed algorithms that use diagnosis and medication codes to identify case and control statuses for patients within the SD for certain conditions.

We focus on the patients identified as case and controls for resistant hypertension. For each patient in the tensor, we include five years of data from the last diagnosis they received. We construct the count tensor from medication and diagnosis records. Since individual International Classification of Diseases (ICD-9) diagnosis codes capture information at a fine-grained level specialized for billing purposes, we use CMS's Hierarchical Condition Categories (HCC) to group the diagnosis codes.[1] Additionally, we aggregate medications based on Medical Subject Headings (MeSH) pharmacological actions provided by the RxClass REST API, a product of the US National Library of Medicine.[2] It is important to note that a medication may have several uses and, therefore, belong to multiple categories. The resulting tensor is 1394 patients by 177 diagnoses by 149 medications and thus has over 36 million cells. Of these patients, 33% of the patients were labeled as resistant hypertension cases and 67% were labeled as controls.

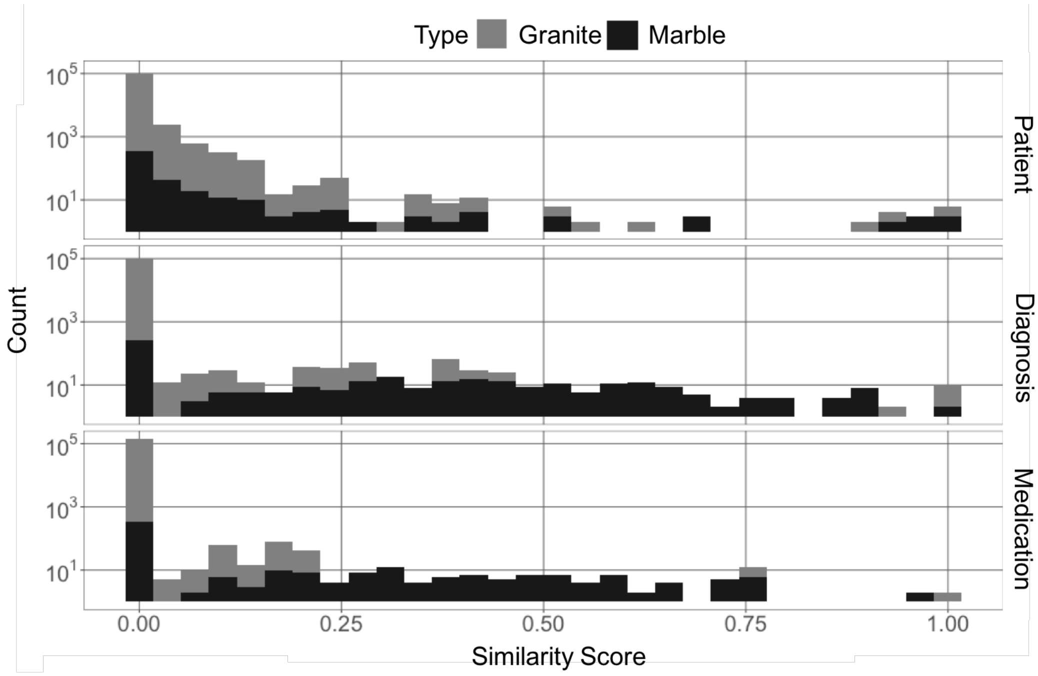


Figure 3.3: Cosine similarity within factor matrices for Granite ($s = [1, .99, .99], \theta = [1, .35, .35], \beta_1 = 10000, \beta_2 = 10000$) and Marble ($\gamma = [0, 0.15, 0.15], \alpha = 10000$) with R = 30 (counts are shown on a log scale).

[1] http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

[2] https://rxnav.nlm.nih.gov/RxClassAPIs.html

3.4.2.2 Results

We evaluate Granite against other dimensionality reduction techniques in the following three ways: 1) we quantitatively compare Granite-generated phenotypes with Marble-generated phenotypes to demonstrate the desirable qualities of Granite, 2) we use annotations from a domain expert to analyze the clinical relevance of the phenotypes, and 3) we use the phenotypes generated in an unsupervised manner in a supervised classification task to demonstrate the predictive power of Granite.

First, we compare phenotypes generated with Granite with those generated by Marble. Figure 3.2 shows the Granite- (top) and Marble-generated phenotypes (bottom) associated with the largest weights, $\boldsymbol{\lambda}$. The numbers in parentheses next to the items indicate in which phenotypes the items appear. For example, "Other infectious diseases" is labeled "(1, 3, 5)" because it is repeated in Phenotypes 1, 3, and 5 in the Marble-generated phenotypes. Overall, Granite produces more diverse phenotypes, which is illustrated in Figure 3.3. Using a log scale for the counts, Figure 3.3 shows histograms of the cosine similarity scores between vectors by mode. Here, the angular penalty term for the Granite decomposition was set to .35, and in the histogram, it can be seen that the vectors in the Granite factor matrices have cosine scores between 0 (completely perpendicular) and .4 (a small number of common terms) in the diagnosis mode and 0 and .25 in the medication mode. Note that since the angular penalty was set to 1 for the patient mode, there is less diversity in this mode, which may be preferable from a clinical perspective. In contrast, the cosine similarity scores for Marble-generated factor vectors are more widely dispersed, especially in the diagnosis and procedure mode. This indicates there is more overlap using Marble.

Experimentally we found Granite-generated phenotypes can cover a range of sizes for patient groups. Tables 3.2 and 3.3 show the phenotypes extracted using Granite, where * denotes features that were related to case patients and the † denotes features related to control patients according to our predictive model (discussed later in this section). Most phenotypes capture small parts of the population, demonstrating the potential for our algorithm to uncover rare phenotypes.

Next, we examine the clinical relevance of the generated phenotypes. A domain expert graciously annotated the Granite- and Marble-generated phenotypes as "clinically relevant", "possibly clinically relevant", and "not clinically relevant." Overall, Granite generated fewer clinically relevant phenotypes than Marble, but we found that the clinical relevance of Granite-generated phenotypes was highly correlated with the weight associated with the phenotype (i.e., higher λ_r means more likely to be relevant). On the other hand, Marble-generated phenotypes did not exhibit this relationship. Figure 3.4 shows the Receiver Operator Curve based on using the λ weight associated with the phenotype to classify that phenotype as clinically meaningful or not. This analysis suggests there is a trade-off between diversity and clinical relevance. By encouraging diverse solutions through the angular penalty term, Granite is more likely to find less relevant phenotypes that correspond to smaller weights, and

Table 3.1: AUC using R = 30, Granite's parameters are set to $s = [1, .99, .99], \theta = [1, .35, .35], \beta_1 = 10000, \beta_2 = 1000$, Marble's parameters are set to $\alpha = 10000, \gamma = [0, .15, .15]$.

Method	Average AUC	Std. Dev.	Average Non-Zeros Per Phenotype (Modes 1 and 2)
Granite	0.7298	0.0243	**4.6300** (w/o bias)
Marble	0.7197	0.0190	5.3330 (w/o bias)
CP-APR	0.7405	0.0117	111.0000
CP-ALS	0.6765	0.0234	113.1522
NMF	0.7203	0.0315	NA

in practice, these phenotypes can be discarded. Moreover, the discriminative power of Granite and its ability to generate sparse and diverse phenotypes make it a useful tool for clinicians.

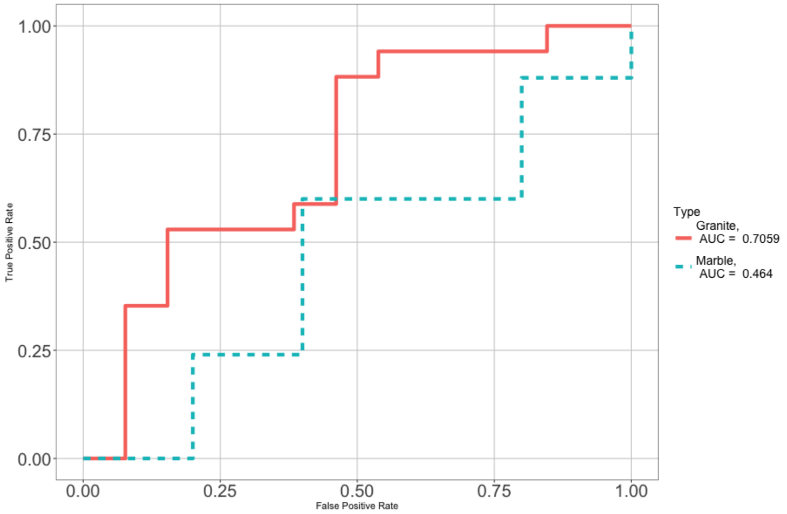
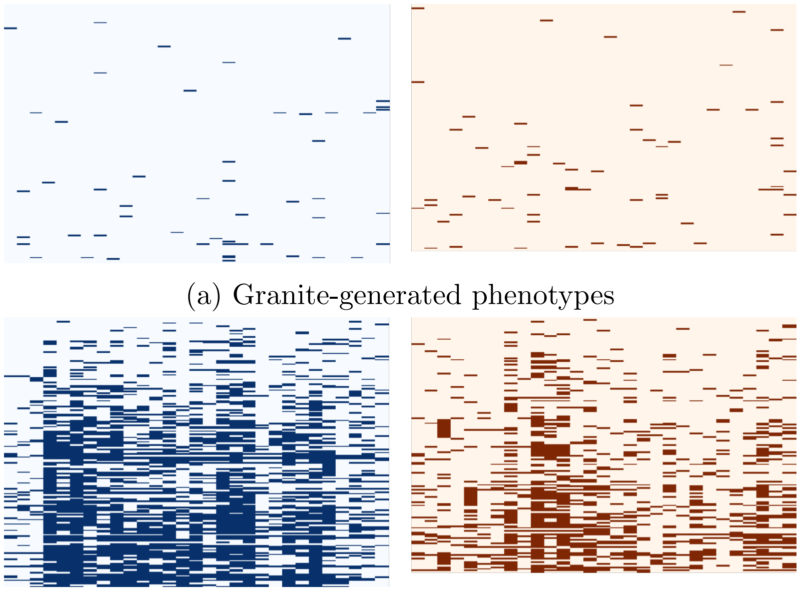


Figure 3.4: ROC for Granite and Marble where classification task was to predict which phenotypes are clinically significant based on λ weight.

Finally, we compare Granite's predictive performance to Marble, CP-APR with nonnegative constraints, CP-ALS with nonnegative constraints, and



(a) Granite-generated phenotypes

(b) CP-APR-generated phenotypes

Figure 3.5: Heatmap of non-zero elements in factors of diagnosis (dark blue) and medication (dark orange) modes generated by Granite and CP-APR phenotypes (Granite used $\theta = [1, 0.3, 0.3], \beta = 10000, s = [1, 0.99, 0.99]$.)

Non-negative Matrix Factorization (NMF) using a classification task to predict resistant hypertension patients. It is important to note that the derived features for these methods are obtained through unsupervised learning (i.e., phenotypes are not adapted to fit the classification model). For the five methods, we fix the number of computational phenotypes at thirty ($R = 30$), based on analysis of the log-likelihood, and derive computational phenotypes from the constructed tensor. We performed a grid search on parameters for Granite and Marble in order to obtain a good tradeoff between sparsity and diversity. We then train ℓ_1-regularized logistic regression models on phenotypes from each of the aforementioned methods. Note for NMF, phenotypes are derived from

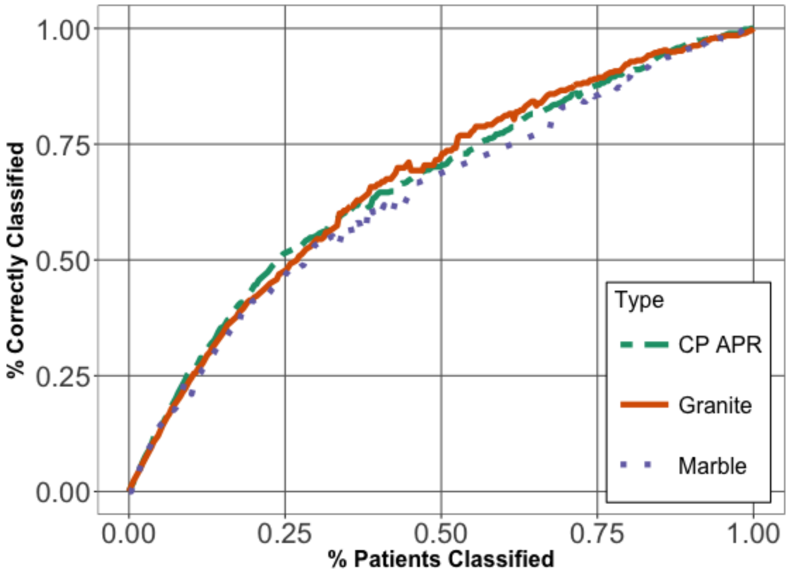


Figure 3.6: Cumulative gains chart for predicting hypertension case and controls. The solid line denotes Granite's performance.

a matricized version of the tensor (i.e., $\mathbf{W}$ are the features where $\mathbf{X} \approx \mathbf{W}\mathbf{H}^{\intercal}$). We ran the model on five 80-20 train-test splits, generated using stratified random sampling, with the features derived from the training dataset only. For CP-ALS, CP-APR, Marble, and Granite, the phenotype membership matrix is the feature matrix, and for NMF, the patient loadings matrix is the feature matrix. The optimal LASSO parameter for the regression model is learned via 10-fold cross-validation in the SD population.

Table 4.3 shows the area under the receiver operating characteristic curve (AUC) for the different methods and the average number of non-zero entries in the diagnosis and medication modes per phenotype. It can be seen that Granite has a higher predictive performance than CP-ALS, Marble, and NMF.

The low performance of CP-ALS might indicate that the Poisson assumption is important. Since CP-APR is not restricted by sparsity constraints, it is able to capture more of the population and unsurprisingly has the best AUC. However, CP-APR-generated phenotypes are not sparse. Figure 3.5 shows the number of nonzero terms in the medication and diagnosis modes for Granite-generated phenotypes (Figure 3.5a) and CP-APR-generated phenotypes (Figure 3.5b). The large number of medication and diagnoses codes per phenotype (111 total codes on average) of CP-APR makes the generated phenotypes harder to interpret than the substantially more concise Granite-generated phenotypes (4.6300 total codes on average). Therefore, we can conclude Granite phenotypes are discriminative, with sparsity and diversity, which we believe should make this method more attractive than its competitors.

To look more closely at the important features in the classification task, we return to Tables 3.2 and 3.3 where features most predictive of cases and controls are indicated by * and †, respectively. It is interesting to note that in addition to "hypertension" appearing in the most predictive of features of case patients (Phenotype 9), comorbidities of hypertension, like diabetes (Phenotype 23) [Long and Dagogo-Jack, 2011] and angina pectoris (Phenotype 21) [Richardson and Hill, 1979], also appear to be predictive. Figure 3.6 shows a cumulative gains chart of the three methods. All five methods perform similarly on smaller proportions of the population, but, as the percent of patients classified increases, Granite is more discriminative. Granite's diverse phenotypes are expected to be more useful to clinicians because it will reduce the time needed to sift through Marble's repetitive phenotypes and CP-APR's and CP-ALS's lengthy phenotypes to discover clinically interesting features of a population.

3.5 Conclusion

In this chapter, we presented Granite, a diverse and sparse Poisson nonnegative model to fit EHR count data. Our algorithm provides an unsupervised methodology to achieve high-throughput phenotyping. The model generates multiple concise and interpretable computational phenotypes with minimal supervision but also yields high diversity factors with minimal overlapping elements between the phenotypes.

The experimental results on simulated and real EHR data demonstrate the conciseness, interpretability, diversity, and predictive power of Granite-derived phenotypes. Granite can also be used to rapidly characterize, predict, and manage a large number of diverse diseases, thereby promising a novel, data-driven solution that can benefit the entire population. Despite its merits, there are certain limitations to Granite. Particularly, a domain expert who reviewed Granite phenotypes found fewer clinically relevant phenotypes than those derived by Marble. The domain expert noted there could be mismatches between diagnoses and medications in the phenotypes. Thus, requiring diversity comes at the cost of degrading the clinical meaningfulness of phenotypes. In Chapter 6 we show how to leverage the domain expertise contained in a corpus of publicly available to guide the phenotyping process to more clinically

Table 3.2: Granite phenotypes ranked by λ_r, * denotes the phenotypes most predictive of being a hypertension case, † denotes the phenotypes most predictive of being a control. Diagnoses are orange (capitalized), and medications are blue (uncapitalized) (Part 1).

Phenotype 1 (15.43% of Patients)
Legally Blind
Major Symptoms, Abnormalities
Polyneuropathy
Cerebrovascular Disease Late Effects, Unspecified
Multiple Sclerosis
anticonvulsants
bronchodilators
anxiolytics, sedatives, and hypnotics

Phenotype 2 (10.76% of Patients)
Specified Heart Arrhythmias
Major Symptoms, Abnormalities
Heart Infection/Inflammation, Except Rheumatic
diuretics
beta-adrenergic blocking agents
antihyperlipidemic agents

Phenotype 3 (5.92% of Patients)
Other Endocrine/Metabolic/Nutritional Disorders
Severe Hematological Disorders
vitamins

Phenotype 4 (3.41% of Patients)
Rheumatoid Arthritis and Inflammatory Connective Tissue Disease
antirheumatics

Phenotype 5 (7.71% of Patients)
Other Endocrine/Metabolic/Nutritional Disorders
antihyperlipidemic agents

Phenotype 6 (0.72% of Patients)*
Lymphoma and Other Cancers
antiviral agents

Phenotype 7 (0.54% of Patients)
Severe Hematological Disorders
antiemetic/antivertigo agents

Phenotype 8 (2.24% of Patients)
Major Symptoms, Abnormalities
antifungals

Phenotype 9 (3.5% of Patients)*
Cardio-Respiratory Failure and Shock
Hypertension
antiarrhythmic agents

Phenotype 10 (0.36% of Patients)
Major Symptoms, Abnormalities
Other Infectious Diseases
antituberculosis agents

Phenotype 17 (8.61% of Patients)
Pelvic Inflammatory Disease and Other Specified Female Genital Disorders
Osteoporosis and Other Bone/Cartilage Disorders
bronchodilators
anticonvulsants
vitamins
laxatives
antacids

Phenotype 18 (1.08% of Patients)
Severe Hematological Disorders
antiviral agents
antiparkinson agents
analgesics
GI stimulants
anticoagulants
chelating agents
antimetabolites

Phenotype 19 (0.72% of Patients)
Lung and Other Severe Cancers
mouth and throat products

Phenotype 20 (0.45% of Patients)†
Quadriplegia
mouth and throat products

Phenotype 21 (9.42% of Patients)*
Angina Pectoris/Old Myocardial Infarction
antianginal agents
diuretics
antiplatelet agents
nutraceutical products

Phenotype 22 (16.5% of Patients)
Precerebral Arterial Occlusion and Transient Cerebral Ischemia
Coronary Atherosclerosis/Other Chronic Ischemic Heart Disease
Urinary Tract Infection
Coagulation Defects and Other Specified Hematological Disorders
Major Symptoms, Abnormalities
Hypertension
Pressure Pre-Ulcer Skin Changes or Unspecified Stage
Other Endocrine/Metabolic/Nutritional Disorders
Diabetes with No or Unspecified Complications
hormones/antineoplastics
tetracyclines
immunostimulants
antihyperlipidemic agents

Phenotype 23 (0.72% of Patients)*
Diabetes with No or Unspecified Complications
nutraceutical products

Table 3.3: Granite phenotypes ranked by λ_r, * denotes the phenotypes most predictive of being a hypertension case, † denotes the phenotypes most predictive of being a control. Diagnoses are orange (capitalized), and medications are blue (uncapitalized) (Part 2).

Phenotype 11 (0.54% of Patients)*
Opportunistic Infections
immunosuppressive agents
immunostimulants
antiviral agents
antidepressants

Phenotype 12 (3.86% of Patients)
Major Symptoms, Abnormalities
Disorders of the Vertebrae and Spinal Discs
prolactin inhibitors
antiarrhythmic agents

Phenotype 13 (2.15% of Patients)
Diabetes with No or Unspecified Complications
bronchodilators
laxatives
antihistamines

Phenotype 14 (1.35% of Patients)
Other Endocrine/Metabolic/Nutritional Disorders
antiviral agents

Phenotype 15 (0.45% of Patients)†
Major Head Injury
anxiolytics, sedatives, and hypnotics
antiarrhythmic agents

Phenotype 16 (0.54% of Patients)†
Colorectal, Bladder, and Other Cancers
otic preparations
adrenal cortical steroids

Phenotype 24 (11.03% of Patients)
Uncompleted Pregnancy With Complications
Drug/Alcohol Psychosis
Rheumatoid Arthritis and Inflammatory Connective Tissue Disease
Attention Deficit Disorder
macrolide derivatives
ophthalmic preparations

Phenotype 25 (9.15% of Patients)†
Traumatic Amputation
ophthalmic preparations
local injectable anesthetics
miscellaneous uncategorized agents

Phenotype 26 (7.89% of Patients)
Hemiplegia/Hemiparesis
hormones/antineoplastics
immunostimulants
anticonvulsants

Phenotype 28 (0.09% of Patients)
Severe Hematological Disorders
uterotonic agents

Phenotype 29 (1.17% of Patients)†
Other Eye Disorders
Poisonings and Allegic Reactions
Other Infectious Diseases
Other Endocrine/Metabolic/Nutritional Disorders
medical gas

Phenotype 30 (0.9% of Patients)
Acute Myocardial Infarction
antidiarrheals

meaningful phenotypes.

Chapter 4

Patient-Disease-Status-Aware Phenotyping

In some situations, we may have access to information about the disease status of sets of patients. Incorporating this information into the tensor decomposition process can result in the discovery of phenotypes that give more nuanced views of that disease or other diseases within the population. We present two models, Greedy Angular Multiway Array Iterative Decomposition (gamAID) [Henderson et al., 2017b] and Phenotyping through Semi-Supervised Tensor Factorization (PSST) [Henderson et al., 2018a], that include this information in the fitting process. The first, gamAID, starts with two classes of patients, and in an alternating manner, accumulates phenotypes that are representative of the class in question and distinct from the other class. The second, PSST, uses patient disease status (case and control), to encourage membership in the phenotypes to be either majority case or control patients. PSST accomplishes this through cannot-link constraints on the patient factor matrix.

4.1 Greedy Angular Multiway Array Iterative Decomposition (gamAID)

4.1.1 Introduction

Over time, populations of patients with the same disease have different disease trajectories. Some patients may stay stable in their conditions, and other patients may develop other diseases as the result of the first disease. Being able to identify early signs of a diverging disease trajectory can be key to managing a patient's care. An example of this is diabetes, which can cause kidney damage with varying degrees of severity. This damage, called diabetic nephropathy, is a type of Chronic Kidney Disorder (CKD) and is found in 23% of diabetes patients. The presence of both CKD and diabetes in a patient can result in complications of care. For example, reduced kidney function inhibits the amount of insulin the kidneys can remove from a person's blood, which makes the process of controlling a diabetic patient's glycemic levels more challenging. Therefore, being able to identify early signs of CKD in diabetes patients can help mitigate complications of simultaneously managing diabetes and CKD [Cavanaugh, 2007].

To explore the early signs of complicating diseases, we developed Greedy Angular Multiway Array Iterative Decomposition (gamAID) [Henderson et al., 2017b], an exploratory, supervised nonnegative tensor factorization method that iteratively extracts phenotypes from tensors constructed from medical count data. gamAID discovers what phenotypes differentiate two groups of patients that are similar at time y_t but different from one another at time

y_{t+1}. gamAID is a tensor decomposition model similar to Granite in terms of objective function but different in terms of application. Our goal was to accumulate computational phenotypes that are representative of each patient class that are “different” from phenotypes discovered in the other patient class. In this work, we demonstrated the potential of the method on diabetes patients who do and do not develop CKD and compared its performance against another dimensionality reduction technique, Fisher's Linear Discriminant Analysis (LDA).

4.1.2 Methods

Greedy Angular Multiway Array Iterative Decomposition (gamAID), is an exploratory, supervised non–negative tensor factorization method for uncovering distinctive phenotypes that can differentiate patients with or without a disease. Our goal is to accumulate computational phenotypes that are representative of each patient class that are “different” from phenotypes discovered in the other patient class. Given the binary labels representing whether or not a diabetic patient is diagnosed with CKD in the year following the observed records, we construct three types of tensors to which gamAID will fit decompositions. The first tensor, $\boldsymbol{\mathcal{X}}_{(01)}$ contains EHR count data from both classes of patients. We then split $\boldsymbol{\mathcal{X}}_{(01)}$ along the patient mode to form $\boldsymbol{\mathcal{X}}_{(0)}$ and $\boldsymbol{\mathcal{X}}_{(1)}$ so that they only count data specific to the class in question (i.e., class 0 or class 1). gamAID fits one of three tensors $\boldsymbol{\mathcal{Z}}_{(01)}$, $\boldsymbol{\mathcal{Z}}_{(0)}$, and $\boldsymbol{\mathcal{Z}}_{(1)}$ based on what step it is in. These fit tensors are the same size as their respective observation tensor, and each element $z_{\vec{i}}$ contains the optimal Poisson parameter for the observed tensor $x_{\vec{i}}$. We constrain the fit tensors to share all but one of the same factor vectors along the non–patient factors (i.e., diagnosis, procedure). Thus, the discovered phenotypes can be used to uncover higher–order interactions, which can then be used as distinguishing characteristics for improved prediction and understanding. Given the patient classes are similar to one another, the decomposition $\mathcal{Z}_{(01)}$, fit on $\mathcal{X}_{(01)}$, captures some features held in common between the two classes.

Like Granite (Chapter 3), gamAID uses angular constraints to encourage diversity between factor vectors of each mode by penalizing any pair of vectors that are similar. This helps gamAID find phenotypes that are different from previously discovered phenotypes. It is important to note that since $\mathcal{X}_{(01)}$ consists of count data, it is not possible to standardize the tensor by subtracting off the mean and dividing by the standard deviation. Thus, a bias term, $\mathbf{u}^{(n)}$, is added in Equation (6.16) to capture the baseline state of the data. Each factor matrix $\mathbf{A}^{(n)}$ can be projected onto a sparse simplex denoted by s (shown in Equation (4.4)), which provides a tunable parameter to alter the number of elements in the resulting factors. The optimization problem that is solved for each separate tensor $\boldsymbol{\mathcal{X}}_{(d)}$, where $d \in \{0, 1, 01\}$, is the following:

$$f(\boldsymbol{\mathcal{Z}}_{(d)}) = \min(\sum_{\vec{i}}(z_{\vec{i}(d)} - x_{\vec{i}(d)} \log z_{\vec{i}(d)}) \tag{4.1}$$

$$+ \frac{\beta}{2} \sum_{n=1}^{N} \sum_{r=1}^{R} \sum_{p=1}^{r} (\frac{(\mathbf{a}_p^{(n)})^\intercal \mathbf{a}_r^{(n)}}{||\mathbf{a}_p^{(n)}||_2 ||\mathbf{a}_r^{(n)}||_2})^2 \tag{4.2}$$

$$\text{s.t } \boldsymbol{\mathcal{Z}}_{(d)} = [\![\sigma; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!] + [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!] \tag{4.3}$$

$$||\mathbf{a}_r^{(n)}||_1 = s, \ 0 < s \leq 1, \ \forall n \tag{4.4}$$

$$||\mathbf{u}^{(n)}||_1 = 1, \ \forall n. \tag{4.5}$$

gamAID uses a greedy algorithm to iteratively build up a tensor decomposition of size R by fitting rank one tensors only using one class at a time. The algorithm, which is illustrated in Figure 4.1, fits a rank one tensor that is "best fit" relative to the class and the rank-one tensors we have already accumulated. The first step is to fit the best rank one tensor $\boldsymbol{\mathcal{Z}}_{(01)}$ to $\boldsymbol{\mathcal{X}}_{(01)}$ based on the optimization problem described above ($\boldsymbol{\mathcal{Z}}_{(01)} = \lambda_1 \mathbf{a}_1^{(1)} \circ \mathbf{a}_1^{(2)} \circ \mathbf{a}_1^{(3)}$). We then choose one class, $\boldsymbol{\mathcal{X}}_{(1)}$, and minimize the optimization problem with respect to $\boldsymbol{\mathcal{X}}_{(1)}$ to fit a rank-two decomposition, with the first rank one tensor set to the one learned in the previous steps ($\boldsymbol{\mathcal{Z}}_{(1)} = \lambda_1 \mathbf{a}_1^{(1)} \circ \mathbf{a}_1^{(2)} \circ \mathbf{a}_1^{(3)} + \lambda_2 \mathbf{a}_2^{(1)} \circ \mathbf{a}_2^{(2)} \circ \mathbf{a}_2^{(3)}$). gamAID then switches classes and minimizes the optimization problem with respect to $\boldsymbol{\mathcal{X}}_{(0)}$ to fit a rank-three decomposition based on the two rank one tensors learned previously. gamAID continues to switch classes until the user-specified number of phenotypes R. The patient mode for each class needs to be refit each time as the membership to phenotypes might be redistributed for a given patient, given a new set of phenotypes. The pseudocode for the

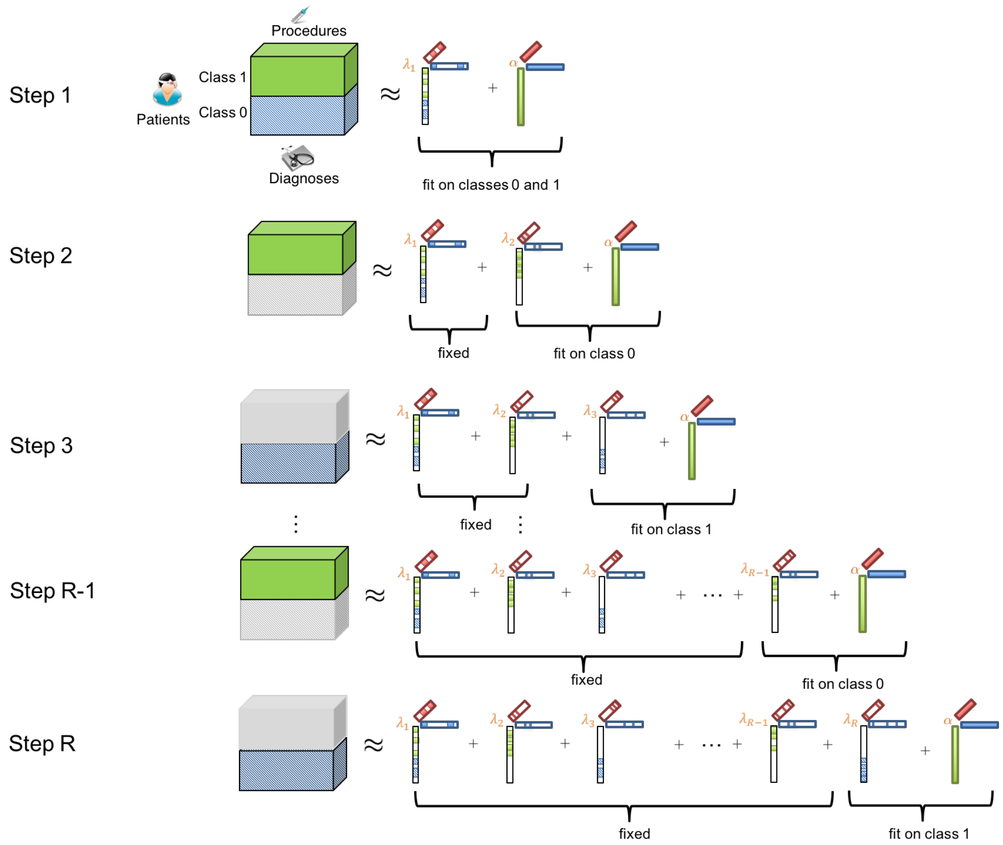


Figure 4.1: Illustration of the gamAID process. gamAID greedily accumulates phenotypes by fitting tensors specific to each class and holding the previously fixed tensors fixed.

algorithm is shown in Algorithm 4.

Algorithm 4: Pseudocode for the gamAID algorithm

```
Data: $\mathcal{X}, \mathcal{X}_{(1)}, \mathcal{X}_{(0)}, K$
Result: $[\![\sigma; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!], [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \mathbf{A}^{(2)}; \mathbf{A}^{(3)}]\!]$
for $r = 1, 2, \cdots, R$ do
    if $r == 1$ then
        Solve the optimization problem (Equations (4.1)- (4.5)) for
        $\mathcal{X}_{(01)}$, the tensor corresponding to both class 0 and class 1
        patients
    end
    if $r$ is even then
        Solve the optimization problem (Equations (4.1)- (4.5)) for
        $\mathcal{X}_{(1)}$, the tensor corresponding to class 1 patients
    end
    if $r$ is odd and $r > 1$ then
        Solve the optimization problem (Equations (4.1)- (4.5)) for
        $\mathcal{X}_{(0)}$, the tensor corresponding to class 0 patients
    end
end
```

At the end of the gamAID process, the diagnosis and procedure modes are fixed and the combined patient factor matrix is learned by minimizing the objective function once more. The final step is to normalize across the rows of the patient factor matrix. We can interpret the normalized values as a patient's membership to or loading on a phenotype.

4.1.3 Experiments

4.1.3.1 Data

We demonstrate the potential of the gamAID framework on the publicly available *CMS Linkable 2008-2010 Medicare Data Entrepreneurs' Synthetic*

Public Use File (DE-SynPUF) that the Centers for Medicare and Medicaid Services (CMS) provides.[1] It contains claim records spanning 3 years of data. The records have been synthesized from 5% of the 2008 Medicare population to protect the privacy of the patients. DE-SynPUF contains inpatient, outpatient, carrier, and prescription drug event claims in addition to the beneficiary files. Although the relationships between some of the variables have been altered to minimize re-identification risk, due to the very large size and coverage of the data, the conclusions obtained by population-level models are expected to closely represent those obtained from the unaltered dataset, and thus still provided very valuable clinical insights.

We extracted two classes based on values for different disease flags in the Beneficiary file. Class 1 consists of patients flagged as diabetic in 2009 and 2010, who did not have a chronic kidney disease (CKD) flag in 2009 but did have a CKD flag in 2010. We also refer to this class as "diabetes-CKD." Class 0, which we also refer to as "diabetes only," consists of patients with a diabetes flag in 2009 and 2010 and no CKD flag in 2009 or 2010.

The extracted cohort consists of $1,492$ diabetes-CKD patients and $1,625$ diabetes-only patients. Figure 4.2 shows the difference between the diagnosis counts between diabetes–only and diabetes–CKD patients. For reference, the negative side of the x-axis are diagnoses that appeared more in

[1] The dataset can be downloaded at `https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/index.html`

diabetes–CKD than diabetes–only patients. Our analysis also showed that some diagnoses appear in one class but not the other. We focused on diagnosis and procedures for the two non-patient modes. To build the tensor, we use the 50 diagnosis with the highest counts for each class as well as the diagnoses that appeared much more in one class than the other. We included all procedures associated with these diagnoses.

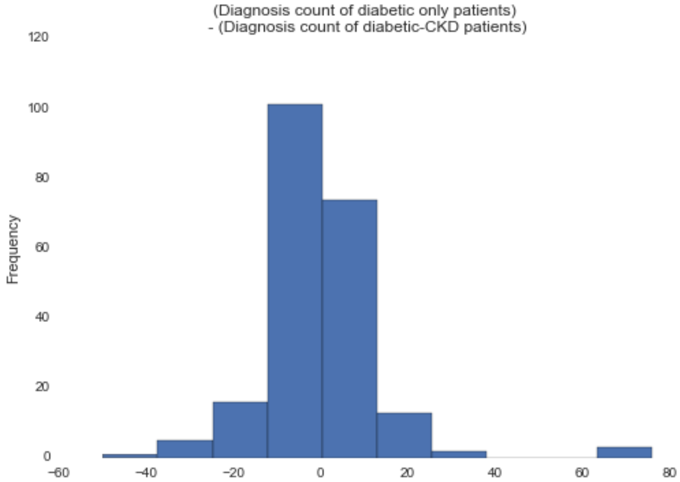


Figure 4.2: Histogram of difference between diagnosis counts between classes.

4.1.3.2 Results

We used the gamAID process to accumulate 9 phenotypes from $\boldsymbol{\mathcal{X}}_{(01)}$ (to fit phenotype 1), $\boldsymbol{\mathcal{X}}_{(1)}$ (to fit phenotypes 2, 4, 6, 8), and $\boldsymbol{\mathcal{X}}_{(0)}$ (to fit phenotypes 3, 5, 7, 9). After finishing the gamAID process, we fixed the diagnosis and procedure modes and fit the patient mode to learn the membership of the patients across the phenotypes. Table 4.1 shows the percentage of patients by class per phenotype and the percentage of patients the phenotype captured overall. Interestingly, phenotypes not fit on one class are dominated by that class (e.g., phenotypes 4 is mostly diabetes-only patients though it was fit on diabetes-CKD patients).

This implies that the patients in both classes are quite similar, which makes intuitive sense. Figure 4.3 depicts a selection of phenotypes for which diabetes-CKD patients were the majority. In the future, we plan to consult domain experts on the clinical relevance of the extracted phenotypes, but based on a literature search, many of the elements of the phenotypes in diabetes-CKD majority phenotypes have been documented as being related to chronic kidney disease. For example, gastrointestinal disorders (phenotypes 1 and 9), heart dysrhythmias (phenotype 1), and abdominal pain (phenotype 1) are commonly found in patients with chronic kidney disorder [De Francisco, 2002; Stadler et al., 2015; Boriani et al., 2015]. Additionally, back issues (phenotype 6) are a symptom of chronic kidney disorder [Hays et al., 1994]. While we can say nothing about causation, it is interesting to see that these phenotypic elements were present in the diabetic-CKD majority phenotypes.

In comparison, we applied Fisher's Linear Discriminant Analysis (LDA) to a matricized $\mathcal{X}_{(01)}$ and to the first 30 components of a PCA decomposition of the matricized $\mathcal{X}_{(01)}$[Venables and Ripley, 2013; Wold et al., 1987]. We then used 5-fold cross-validation to fit the projected vector. Figure 4.4 shows a distribution of observations projected onto the linear discriminant. When Fisher's LDA is fit on the raw matricized tensor (top left), it appears there is good separation between the classes. However, when applied to the test

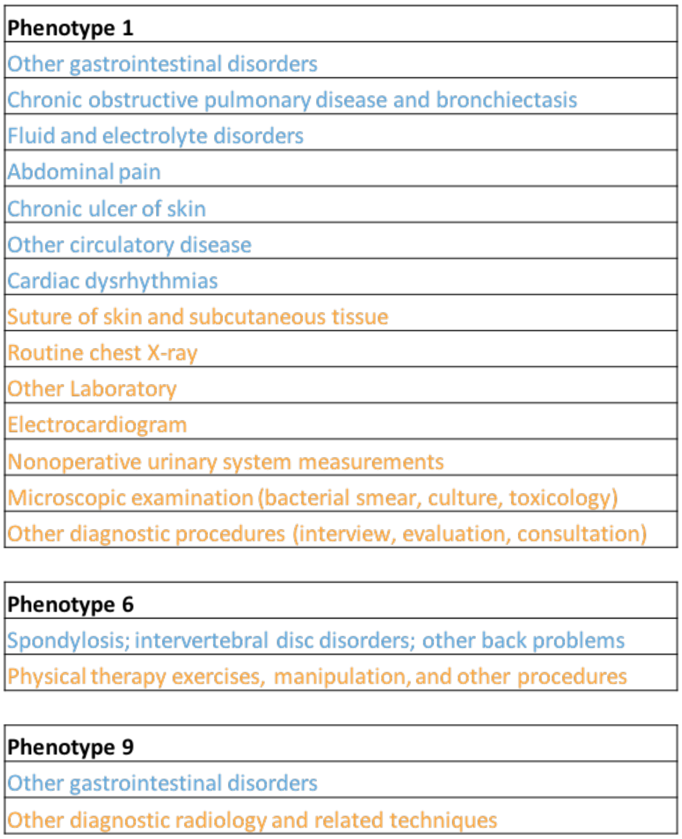
Table 4.1: Percentages of Class Membership by Phenotype

Phenotype	**% Class 1**	**% Class 0**	**% Population Captured**
1	0.52	0.48	0.08
2	0.49	0.51	0.80
3	0.48	0.52	0.10
4	0.48	0.52	0.21
5	0.48	0.52	0.17
6	0.54	0.46	0.09
7	0.00	0.00	0.00
8	0.48	0.52	0.08
9	0.62	0.38	0.01

set (top right), the separation quickly disappears, which suggests overfitting. The training and test distributions of Fisher's LDA applied to the first 30 PCA components look similar (bottom left and right, respectively), but the overlap of the two classes suggests the classes are difficult to separate. Finally, we used the linear discriminant to predict the classes of the test set. This resulted in an average f1-score of .4783 on the raw tensor and .3914 on the PCA components of the tensor. In contrast, a SVM model trained on top of the gamAID decomposition resulted in an average f1-score of .5106. Thus, while this is a difficult classification problem, gamAID shows an improvement over other methods.

4.1.4 Conclusion

We presented an exploratory greedy, iterative approach called gamAID that extracts phenotypes in a supervised manner from a population consisting of diabetes patients without CKD and diabetes patients who will transition to a



Phenotype 1
Other gastrointestinal disorders
Chronic obstructive pulmonary disease and bronchiectasis
Fluid and electrolyte disorders
Abdominal pain
Chronic ulcer of skin
Other circulatory disease
Cardiac dysrhythmias
Suture of skin and subcutaneous tissue
Routine chest X-ray
Other Laboratory
Electrocardiogram
Nonoperative urinary system measurements
Microscopic examination (bacterial smear, culture, toxicology)
Other diagnostic procedures (interview, evaluation, consultation)

Phenotype 6
Spondylosis; intervertebral disc disorders; other back problems
Physical therapy exercises, manipulation, and other procedures

Phenotype 9
Other gastrointestinal disorders
Other diagnostic radiology and related techniques

Figure 4.3: A subset of phenotypes resulting from the gamAID process.

CKD diagnosis in the future. We showed that this method has the potential to tease out phenotypes of diverging disease populations and paired with a simple classifier can identify patients at-risk for CKD. However, gamAID has a few drawbacks. The biggest one is that the types of phenotypes that result from the fit are heavily dependent on which class the algorithm fits first. We addressed this weakness by developing a framework that fits a decomposition on both classes simultaneously. This framework, Phenotyping through Semi-

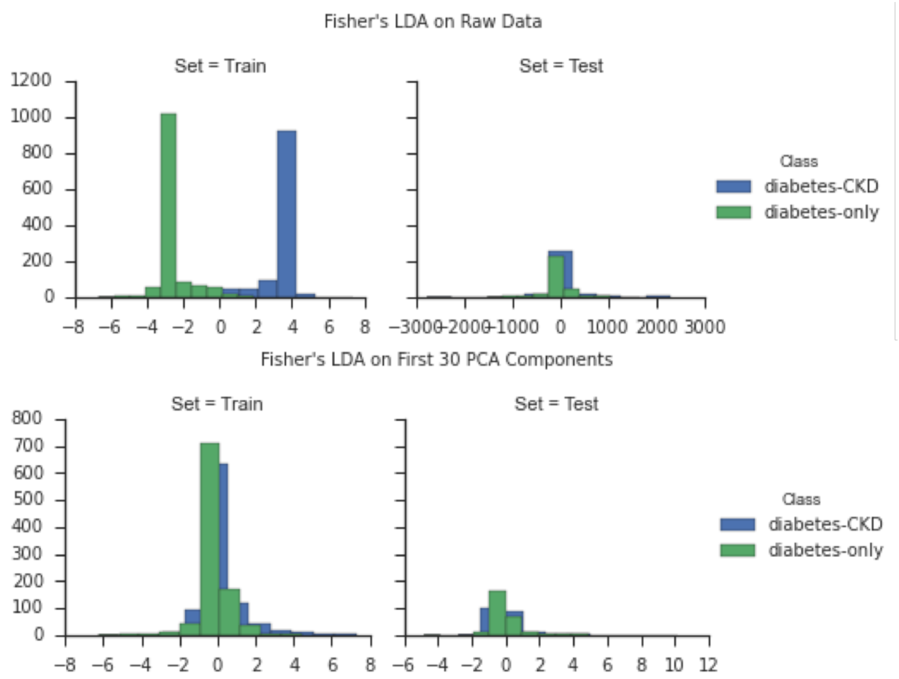


Figure 4.4: LDA distribution of projected data (raw and first thirty components of data transformed by PCA)

Supervised Tensor Factorization (PSST), which is detailed in the next section, imposes constraints on the patient factor matrix to encourage class membership to be limited to one class per phenotype. Furthermore, PSST is flexible enough to work on tensors that contain patients who do not fit the case or control specification.

4.2 Phenotyping through Semi-Supervised Tensor Factorization (PSST)

4.2.1 Introduction

In this section, we present Phenotyping through Semi-Supervised Tensor Factorization (PSST), a novel method that uses partial information about a patient's disease status to mitigate the chance that patients with different disease statuses will appear in the same phenotypes [Henderson et al., 2018a]. We posit that the use of a semi-supervised based approach to leverage *known information available only for a subset of the patients* will lead to phenotypes that are descriptive of the interplay between different disorders. We demonstrate the potential of PSST to extract clinically interesting and discriminative phenotypes by focusing on a dataset of 1,622 patients gathered at Vanderbilt University Medical Center (VUMC) where the disease status is known for a subset of patients. Specifically, we construct a tensor that consists of the following four types of patients: cases and controls of resistant hypertension patients and cases and controls of type-2 diabetes patients. We compare PSST with three other tensor-based computational phenotyping methods, two of which are unsupervised and one of which is supervised. The supervised method, which we will refer to as DDP (Discriminative and Distinct Phenotyping), requires complete knowledge of the disease status of all patients, while PSST does not [Kim et al., 2017b]. This investigation demonstrates how using disease status information on one diagnosis (e.g., resistant hypertension or type-2 diabetes) can reveal discriminative phenotypes, even for other diagnoses, that may not be present in fully supervised or unsupervised approaches.

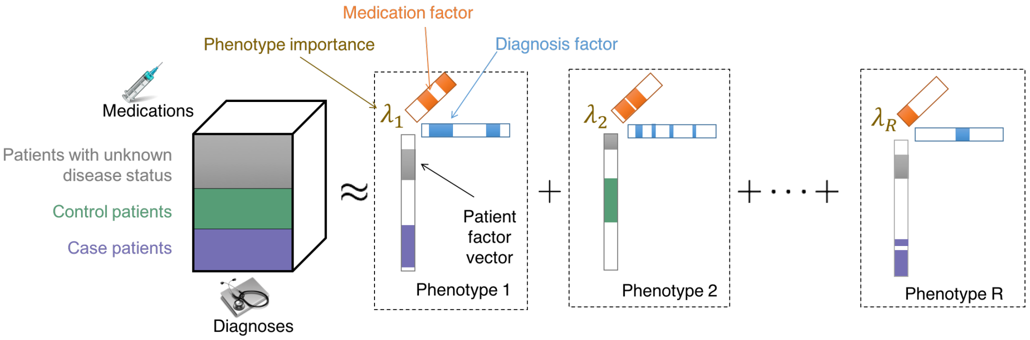


Figure 4.5: An example of phenotyping via tensor factorization. The tensor containing the observed data is pictured as the cube on the left. Each element of the observed tensor corresponds to the number of times a patient received a medication prescription and diagnosis in a set amount of time. A set of rank-one components, formed by taking the outer product of a patient, a diagnosis, and a medication factor vector, is found by minimizing a loss function. The non-zero elements in each component are indicated by colored bars in the factor vectors and consist of the clinical characteristics in that phenotype. The goal of PSST is to use information about the disease status of just a few of the patients within the tensor to encourage patients with different statuses to be in different components, which is indicated by the various colored blocks in the patient factor vectors.

4.2.2 Methods

4.2.2.1 Mathematical Formulation

Like Granite, PSST models the observed data using the Poisson distribution and incorporates angular and ℓ_2 penalties to encourage diversity and control the size of the faactors [Henderson et al., 2017c]. For simplicity, we focus on a 3-mode tensor where the three dimensions are (1) patients, (2) diagnoses, and (3) medications. However, this approach can easily generalize to an N-mode tensor. An observed tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is approximated as the sum of R 3-way rank-one tensors $\mathcal{X} \approx \mathcal{Z} = [\![\boldsymbol{\lambda}; \mathbf{A}; \mathbf{B}; \mathbf{C}]\!]$. PSST introduces a cannot-link matrix on the patient factor matrix ($\mathbf{A}$) to encourage separation in the patients, where different disease statuses are in different phenotypes (e.g., hypertension case patients and hypertension control patients). This notion is illustrated in Figure 4.5. The optimization problem for the observed tensor $\mathcal{X}$ is:

$$
\begin{aligned}
f(\mathcal{X}) = \min(&\sum_{\vec{i}}(z_{\vec{i}} - x_{\vec{i}} \log z_{\vec{i}}) && (4.6)\\
&+ \frac{\beta_1}{2}\sum_{r=1}^{R}\sum_{p=1}^{r}(((\max\{0, \frac{(\mathbf{b}_p)^\intercal \mathbf{b}_r}{||\mathbf{b}_p||_2||\mathbf{b}_r||_2} - \theta\})^2 && (4.7)\\
&+ (\max\{0, \frac{(\mathbf{c}_p)^\intercal \mathbf{c}_r}{||\mathbf{c}_p||_2||\mathbf{c}_r||_2} - \theta\})^2)) && (4.8)\\
&+ \frac{\beta_2}{2}\sum_{r=1}^{R}(||\mathbf{a}_r||_2^2 + ||\mathbf{b}_r||_2^2 + ||\mathbf{c}_r||_2^2) && (4.9)\\
&+ \frac{\beta_3}{2}\text{trace}(\mathbf{A}^\intercal \mathbf{M}\mathbf{A}) && (4.10)\\
\text{s.t } \mathcal{Z} = &[\![\sigma; \mathbf{u}_a; \mathbf{u}_b; \mathbf{u}_c]\!] + [\![\boldsymbol{\lambda}; \mathbf{A}; \mathbf{B}; \mathbf{C}]\!] && (4.11)\\
&||\mathbf{a}_r||_1 = ||\mathbf{b}_r||_1 = ||\mathbf{c}_r||_1 = 1, \mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r \geq 0 && (4.12)\\
&||\mathbf{u}_a||_1 = ||\mathbf{u}_b||_1 = ||\mathbf{u}_c||_1 = 1, \mathbf{u}_a, \mathbf{u}_b, \mathbf{u}_c > 0. && (4.13)
\end{aligned}
$$

For count data, the loss function is the negative log-likelihood between the observed data $\mathbf{x}$ and the model $\mathbf{z}$ parameters (4.6). As introduced in Granite, an angular penalty term (4.7 and 4.8) discourages any factors that are too similar, where similarity is defined as the cosine angle between two factor vectors. Additionally to control the growth of the size of the factors and for computational stability, we include an ℓ_2 penalty term (4.9).

Unlike Granite and Marble, PSST incorporates partial class knowledge to encourage patients with different disease statuses to appear in different phenotypes using a cannot-link semi-supervised penalty term. The cannot-link matrix $\mathbf{M} \in \mathbb{R}^{I_1 \times I_1}$ is constructed such that $m_{i,j} = 1$ only if patients i and j have different disease statuses and is otherwise 0. In (4.10), if patients i and j are in different classes but both belong to phenotype r, then the penalty $a_{ir} \cdot a_{jr}$ is added to the objective function. Thus the term in (4.10) will only contribute to the objective function if two patients have two different disease statuses (e.g., one patient is a case and one is a control) and will be 0 otherwise (e.g., both patients are case, both are control, or one of them is unknown). Figure 4.5 illustrates the impact of this cannot-link term, phenotype 1 and R consist of cases and patients with unknown disease status and phenotype 2 consists of controls and patients with unknown disease status. Since this is a soft penalty, some case and control patients can be in the same phenotype if they are very similar. We use gradient descent to solve the optimization problem.

4.2.3 Experiment Design

4.2.3.1 Dataset and preprocessing

We constructed a tensor from the diagnosis and medication counts of 1,622 patients from the Synthetic Derivative (SD), a de-identified EHR database gathered at the VUMC [Roden et al., 2008]. The SD contains clinical and billing code information for over 2 million inpatient and outpatient interactions. Previously, a panel of domain experts identified sets of characteristics in the form of billing and medical codes of patients as case and control for a set of diseases [Ritchie et al., 2010]. In the present work, we focus on resistant hypertension case and control patients and type-2 diabetes case and control patients. A small subset of these patients are both resistant hypertension and type-2 diabetes case patients (see Table 6.5 for the number of patients in each class).

Table 4.2: Patient disease status (supplied by domain experts) in the VUMC SD dataset used in this study.

Disease Class	**Number of Patients**
Resistant hypertension case	304
Resistant hypertension control	399
Type 2 diabetes case	373
Type 2 diabetes control	452
Type 2 diabetes and resistant hypertension case	94

For each case patient, we counted the medication and diagnosis interactions that occurred two years before they received the diagnosis of the disease (i.e., hypertension or type-2 diabetes). For each control patient, we counted the medication and diagnosis interactions that occurred two years before their last interaction with the VUMC. The diagnosis codes follow the International Classification of Diseases (ICD-9) system and capture information at a high level of detail for insurance purposes. We use PheWAS coding to aggregate the diagnosis codes into broader categories [Denny et al., 2013]. Additionally, we use Medical Subject Headings (MeSH) pharmacological terms provided by the RxClass RESTful API, a product of the US National Library of Medicine, to group the medications into more general categories[2]. These groupings resulted in a tensor with the following dimensions: 1622 patients by 1325 diagnoses by 148 medications.

4.2.3.2 Evaluation metrics

We evaluate PSST along three dimensions: (1) the efficacy of the cannot-link constraint to encourage case and control patients to belong to different phenotypes, (2) the discriminative quality of the resulting phenotypes on an unrelated classification task, and (3) the clinical meaningfulness of the resulting phenotypes.

For the second evaluation metric, we determine if the cannot-link matrix that is used to separate resistant hypertension case and control patients can be used to predict which are the type-2 diabetes case and controls. Likewise, we reversed the two, where a cannot-link matrix is constructed from the type-2 diabetes case and control patients and the resulting phenotypes are used to predict resistant hypertension. For each classification task, we row-normalize the patient factor matrix ($\mathbf{A}$) to obtain a phenotype membership (probability that a patient belongs to each phenotype). Then, using a logistic regression model, we perform a 5-fold cross-validation to evaluate the lift and

[2] https://rxnav.nlm.nih.gov/RxClassAPIs.html

the area under the receiver operating curve (AUC). Lift is the ratio between the results obtained through the predictive model and results obtained without a model. Our hypothesis is that the resistant hypertension cannot-link constraints in PSST will result in phenotypes that uncover latent factors pertinent to type-2 diabetes patients and that type-2 diabetes cannot-link constraints will have a similar effect for identifying hypertension case patients.

To evaluate the clinical meaningfulness, we have enlisted two clinicians to annotate them as clinically relevant or not clinically relevant. To reduce the annotation burden, the classification task results were used to identify highly predictive phenotypes and these were randomly shuffled to avoid biasing the experts.

4.2.3.3 Unsupervised and Supervised Comparison Models

We compared PSST with three other tensor factorization methods: Marble [Ho et al., 2014b], Granite [Henderson et al., 2017c], and DDP [Kim et al., 2017b]. Marble has two sets of parameters relating to the strength of the underlying characteristics (bias term) and the sparsity of the resulting factors. These parameters are tuned to achieve comparable results with respect to the number of non-zero elements per computational phenotype. Granite has both a sparsity-inducing and a diversity-inducing regularization term to yield a sparse set of diverse phenotypes. The Granite parameters (excluding the diversity-inducing term) are tuned to yield the best predictive accuracy. DDP incorporates a logistic regression term, as well as a similarity-based cluster structure, to encourage distinctness. Since this cluster structure requires existing knowledge, we excluded it from our analysis.

4.2.4 Results

We chose $R = 30$ phenotypes for PSST, Marble, and Granite through experimentation. Since DDP was restricted to case and control patients and resulted in a smaller tensor, we found 15 phenotypes resulted in a reasonably good fit.

Efficacy of Cannot-Link Constraints: Class Separation in Patient Factor Matrix. After fitting the PSST decomposition, we analyzed how well it encouraged class separation within the phenotypes, and we compared it to the performance of DDP, Granite, and Marble. For these experiments, we show results for two formulations of PSST and Granite, one with the angular penalty, denoted as "with diversity," and without the angular penalty. In each phenotype, we calculated the percentage of patients who were case and the percentage of patients that were control and then took the difference. For example, a difference of .2 in phenotype k means that one class could have consisted of 40% of the phenotype while the other class consisted of 60% of the phenotype. Figures 4.6 and 4.7 depict histograms of the difference between the percentages within each phenotype for PSST (with and without diversity constraints), Marble, Granite (with and without diversity constraints), and DDP. Figure 4.6 shows that PSST with and without diversity resulted in phenotypes where the majority was either hypertension case (teal bins) or hypertension control (orange bins). Marble and Granite (with and without diversity) resulted in phenotypes that most often consisted of case patients, and DDP resulted in phenotypes that consisted only of case patients.

Similarly in Figure 4.7, PSST with and without diversity constraints results in phenotypes that are either primarily type-2 diabetes case or control patients. Granite with diversity was the only decomposition aside from PSST to derive any phenotypes consisting of a majority control patients. DDP's lack of separation between patient classes is surprising given that it incorporates a logistic regression loss term in its fitting process. In both case studies, the cannot-link constraints in PSST encourage class separation within the phenotypes.

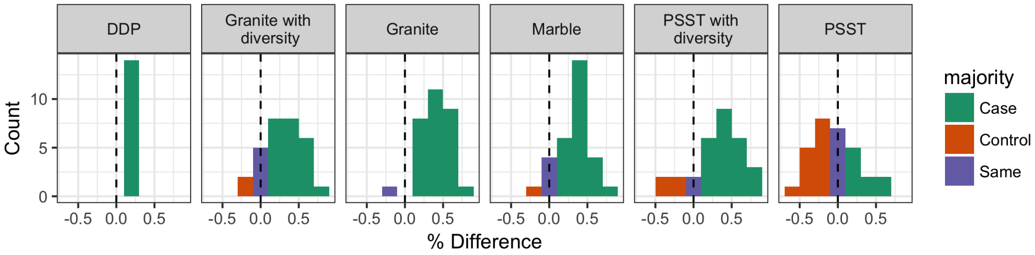


Figure 4.6: Histogram of differences between the percent membership by class for resistant hypertension patients using resistant hypertension cannot-link constraints.

Discriminative Evaluation. Using logistic regression, we compared how well each method discriminates between case and control patients. For PSST, we predict case and control patients that were not used in the cannot-link constraints. Specifically, if a fit was performed with the cannot-link constraints on type-2 diabetes case and control patients, we then predict the resistant hy-

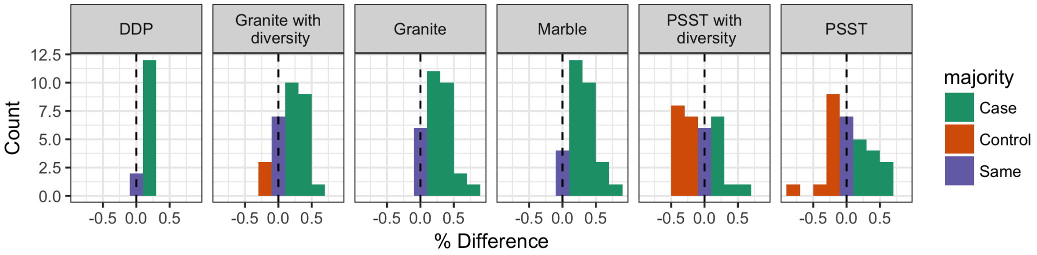


Figure 4.7: Histogram of differences between the percent membership by class for type-2 diabetes patients using type-2 diabetes cannot-link constraints.

pertension case and control patients, and vice versa for cannot-link constraints on resistant hypertension. The features for the logistic regression are the row-normalized patient factor matrix and restricted to the rows corresponding to case and control patients. Table 4.3 shows the AUC values averaged across the five runs for each method for predicting resistant hypertension and type-2 diabetes. As expected, the supervised method DDP outperformed all methods, but PSST had the second highest AUC for each condition. Secondly, there is a tradeoff between diversity constraints (e.g., in PSST and Granite) and the predictive quality of the phenotypes, which was previously noted by Henderson et al. [2017c]. Furthermore, the relatively low AUC values indicate these are difficult classification problems, but the performance of PSST implies incorporating knowledge about a subset of patients can be beneficial.

Figures 4.8 and 4.9 show the lift of the three methods with the highest AUCs in each classification task. When predicting who is a type-2 diabetes case patient (Figure 4.8), DDP has a higher lift than Marble and Granite. On the other hand, when predicting which patients are resistant hypertension case

Table 4.3: AUC for predicting case and control patients using decompositions with cannot-link constraints on the other case and control patients. For example, "Hypertension" below refers to the AUC for predicting hypertension patients when the cannot-link constraints were applied to type-2 diabetes case and control patients.

	Condition	
Method	**Hypertension**	**Type-2 Diabetes**
PSST	0.6618	0.6074
PSST with diversity	0.6275	0.5830
DDP	0.6928	0.6614
Granite	0.6074	0.5528
Granite with diversity	0.5939	0.5745
Marble	0.5919	0.5928

and control in this particular instance (Figure 4.9), PSST consistently has the highest lift. This is surprising given DDP incorporates the resistant hypertension case and control status into fitting the decomposition and has the highest AUC. This indicates that semi-supervision in PSST could be guiding the decomposition toward phenotypes that are meaningful for resistant hypertension patients.

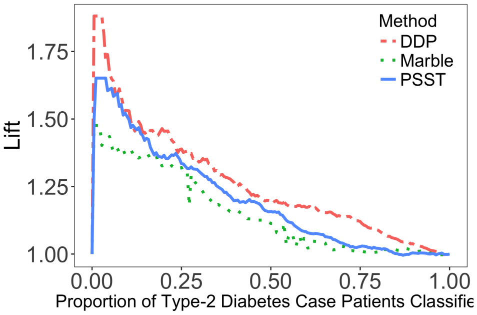


Figure 4.8: Lift curve for type-2 diabetes prediction task.

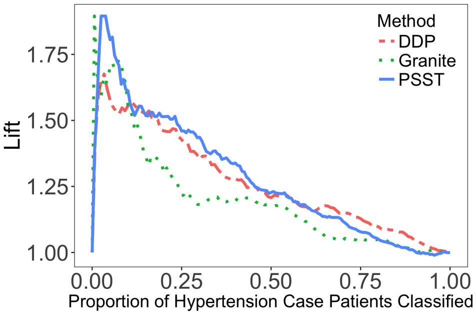


Figure 4.9: Lift curve for resistant hypertension prediction task.

Clinical Relevance Evaluation. As a final step in our analysis, two clinicians annotated the clinical relevance of the phenotypes generated by PSST, Marble, and DDP that were most predictive of being a resistant hypertension case patient. The clinicians assigned each phenotype one of the following labels: 1) clinically meaningful, 2) possibly clinically meaningful, and 3) not clinically meaningful. In total, the clinicians annotated 5 PSST-, 5 Marble-, and 3 DDP-generated phenotypes (DDP had only three positive coefficients). In cases where the annotator's disagreed, we used the label with the lowest clinical relevance score. Using Cohen's Kappa, the inter-rater reliability score was $\kappa = .45$, suggesting the inter-rater agreement was moderate.

Figure 4.10 shows the distribution of the annotations by method. For DDP, 66% of the phenotypes were possibly or not clinically meaningful, suggesting there may be a trade-off between seemingly good predictive quality and clinical relevance. PSST and Marble had the same number of clinically relevant phenotypes, with only 20% deemed not significant. By incorporating semi-supervision through soft constraints, PSST maintains predictive power and interpretative value in this case study.

4.2.5 Discussion

PSST, which only incorporates partial patient information, resulted in phenotypes that had a high degree of separation between case and control patients. The phenotypes extracted by PSST were more predictive of case and control for the two conditions hypertension and type-2 diabetes than two

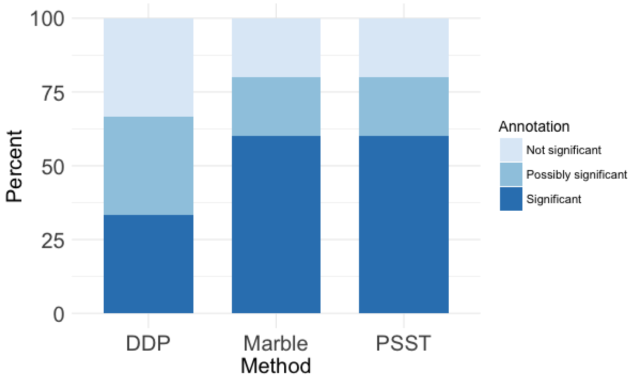


Figure 4.10: Percentage of most predictive phenotypes generated by PSST, Marble, and DDP phenotypes that were clinically significant, possibly clinically significant, not clinically significant.

unsupervised methods. It did not perform quite as well on the prediction task as the supervised method, DDP, but the computational cost in terms of running time to convergence and memory required to use DDP restricts its use. Additionally, in terms of clinical relevance, the phenotypes produced by DDP were not as clinically relevant overall as compared to PSST. This implies that for DDP there may be a trade-off between clinical relevance and predictive power. Furthermore, DDP requires labels for all patients, and the cost of obtaining labels in medical informatics can be high in terms of time and expertise required. Therefore, a semi-supervised method like PSST could help researchers use information available to them without restricting their work to labeled observations.

One major challenge in extracting phenotypes through automatic, machine learning methods is verifying the phenotypes are clinically interesting and meaningful. This validation step is a task that requires domain expertise and time. Furthermore, the phenotypes themselves should be annotated by a panel of experts, and the analysis in the previous section showed that annotators do not agree on the clinical significance of a phenotype at all times. Therefore, it may be beneficial to use a third-party annotator. For this purpose, we developed PheKnow–Cloud, a tool that uses co-occurrence analysis on a publicly available repository of medical articles to calculate a clinical validity score for a supplied phenotypeHenderson et al. [2017a]. PheKnow–Cloud could prove useful for situations where annotators labeled a phenotype as "possibly clinically significant," as they did for a PSST phenotype show in Table 4.4. According to PheKnow–Cloud, this phenotype is likely clinically meaningful, which may lead to further discussion between the annotators.

Table 4.4: Example of phenotype labelled "possibly clinically significant."

Diagnoses	**Medications**
Hyperlipidemia GERD	Angiotensin converting enzyme inhibitors Antihyperlipidemic agents Antiadrenergic agents, centrally acting

4.2.6 Conclusion

We presented Phenotyping through Semi-Supervised Tensor factorization, or PSST, a method that incorporates information from subsets of patients to encourage class separation in patient phenotype membership. Using two case studies, we demonstrated the benefits of integrating partial information into the tensor factorization process to extract phenotypes. We showed the semi-supervised constraints induce considerable class separation between patients with different disease statuses (i.e., case and control) whereas a supervised and two unsupervised methods resulted in little to no class separation. Additionally, PSST may help extract phenotypes that are more descriptive and predictive of patients' disease statuses than purely unsupervised methods, and while PSST did not outperform a supervised method on a prediction task, it did result in phenotypes that were more interpretable than those of the supervised method. Another benefit of PSST is it does not require labels for all observations, and since the cost of obtaining labels can be high, PSST allows for the use of larger datasets for phenotyping tasks than supervised methods.

Chapter 5

Validating Learned Phenotypes

When candidate phenotypes are generated using automatic, unsupervised, and high throughput processes like Granite or PSST, it is necessary to explore their validity, clinical significance, and relevance. To date, these methods are validated by panels of domain experts, which usually consist of clinicians volunteering their time. Albeit less time–consuming than the manual derivation process, the annotating process is still a large time commitment. Annotators are given a set of phenotypes (e.g., 30 candidate phenotypes) at a time, and issues can arise during the phenotype verification process. First, domain expert annotators may disagree on the clinical relevance of a candidate phenotype based on their different experiences as medical professionals. Second, unsupervised methods may generate phenotypes that are unfamiliar to annotators, so they may incorrectly judge a phenotype as clinically insignificant when it is not. Additionally, given that these methods can result in a diverse set of candidate phenotypes, annotators may feel that the objective of phenotype validation is subjective or not well defined.

This section details two frameworks, PheKnow–Cloud and PIVET, that we developed for evaluating the clinical relevance and validity of extracted phenotypes. PheKnow–Cloud, a batch approach, takes as input phenotypes that have been generated in an automatic manner and builds evidence sets and clinical relevance scores based on the analysis of publicly available medical journals. PheKnow–Cloud showed the potential for using medical journals to build evidence sets but its brute force approach is slow and only works for sets of phenotypes. We developed PIVET to be a fast, one-off approach to evaluating phenotypes. PIVET completely refactors each part of the conceptual framework of PheKnow–Cloud to deliver fast performance with comparable discriminative abilities as the original framework.

5.1 PheKnow–Cloud

5.1.1 Introduction

In this section, we discuss PheKnow–Cloud, an interactive tool that uses analysis of publicly available medical journals [Henderson et al., 2017a]. Given a phenotype supplied by the user, PheKnow–Cloud builds sets of evidence for the phenotypes and presents the analysis to the user. Specifically, PheKnow–Cloud leverages the medical expertise within the PubMed Open Access Subset,[1] a publicly available, online database of over one million scientific articles. The tool builds the evidence set by generating co-occurrence counts of phenotypic terms from the articles and uses lift, a metric that summarizes if two or more items co-occur more often than average while accounting for the frequency of the item, as a way to gauge the clinical relevance of the phenotypes

[1] http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

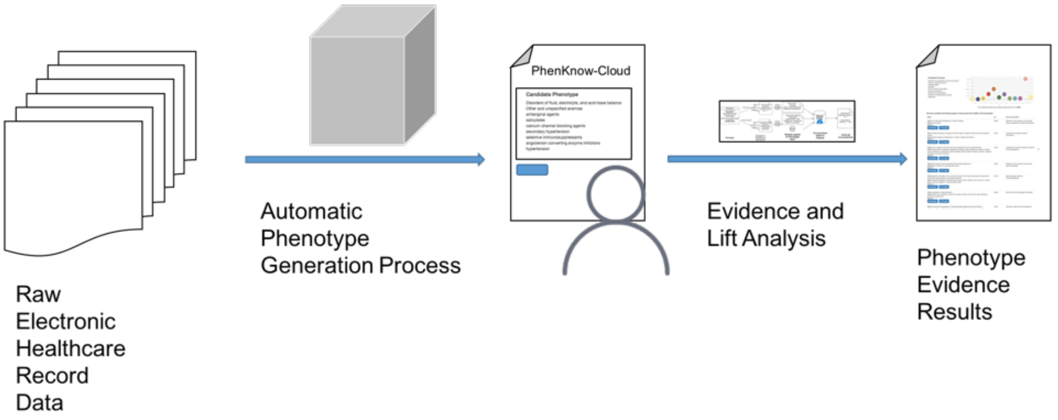


Figure 5.1: The PheKnow–Cloud process.

(see Figure 5.1 for an overview of the process). We summarize the PheKnow–Cloud interface and the process by which the output of PheKnow–Cloud is generated, and then show experimental results to support generating the output in this manner. PheKnow–Cloud builds on preliminary work by Bridges et al. [2016] by substantially improving the evidence set generation process and introducing an interactive interface.

5.1.2 Methods

We first describe the PheKnow–Cloud interface and then discuss the methods that generate each of the features on this interface.

5.1.2.1 PheKnow–Cloud: Front End Process

A standard usage case of PheKnow–Cloud is illustrated in Figure 5.1. First, a user generates phenotypes through an automatic (statistical) method[2]. Then, the user enters a phenotype into the PheKnow–Cloud Welcome Page and starts the analysis process by pressing the “Enter” button. The tool parses the phenotype, and on the backend, uses analysis of PubMed to generate the evidence sets it then presents to the user.

Once PheKnow–Cloud has run the analysis on PubMed, the particulars of which are discussed in the next section, it then presents the user with the results of the analysis on a new page. Figure 5.3 shows a screenshot of the example output on a user-supplied candidate phenotype (with the middle entries of the table omitted for space reasons). The Results page consists of three main parts. The top lefthand corner lists the candidate phenotype the user entered on the Welcome Page. The top righthand corner contains a scatter plot depicting the standard deviations above the median lift of each of the contributing tuples of terms that occurred once or more in the test corpus. This allows the user to understand the distribution of the lifts. Below the plot is the average of the standard deviations above the median of non–zero lifts depicted in the scatterplot. The calculation of these values is discussed in the next section.

The majority of the page consists of a table of information listing a

[2] PheKnow–Cloud does not strictly need an automatically generated phenotype and could potentially be used as a discovery tool.

curated set of articles. This is the crux of PheKnow–Cloud and is the body of evidence that can either give support to the hypothesis that the candidate phenotype is valid or cast doubt on the validity of the hypothesis. The table contains information about the articles that were deemed the most relevant to the phenotype via a process detailed in the next section. The results are sorted by lift in a descending manner, with each row of the table containing information about a PubMed article including the title, author, year, buttons that link to more information, and the tuple of terms that co–occur in the paper. The abstracts of the papers are initially collapsed to allow users to view more articles but can be expanded using the "Abstract" button for users to see if the paper is relevant to the phenotype. Pushing the "Link to the paper" button takes the user to article on PubMed should the user want to examine the article in more detail.

Our framework is flexible and modular to support new metrics and features like sorting and filtering. PheKnow–Cloud can be easily updated with additional refinements to the validation process to help the user better gather evidence for the validity of the supplied candidate phenotype. For example, we plan to allow users to filter on the sets of co-occurring items so they can examine the papers corresponding to those terms. Our tool is built on the new client–server stack, Node.js, Javascript, HTML, and D3.js, which allows us to refine and further develop PhenKnow–Cloud to leverage new interactive visualizations.

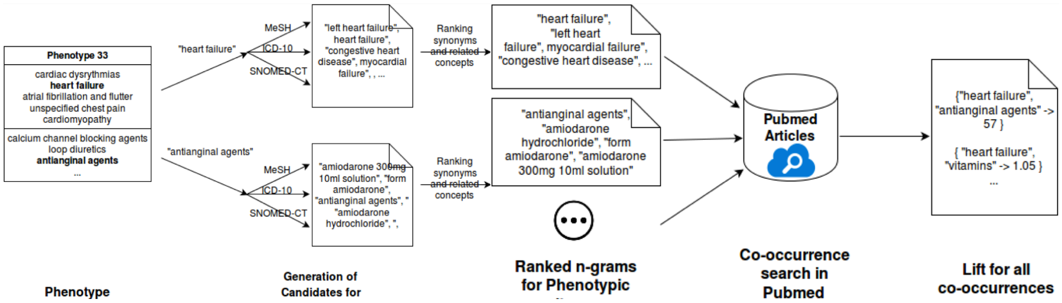
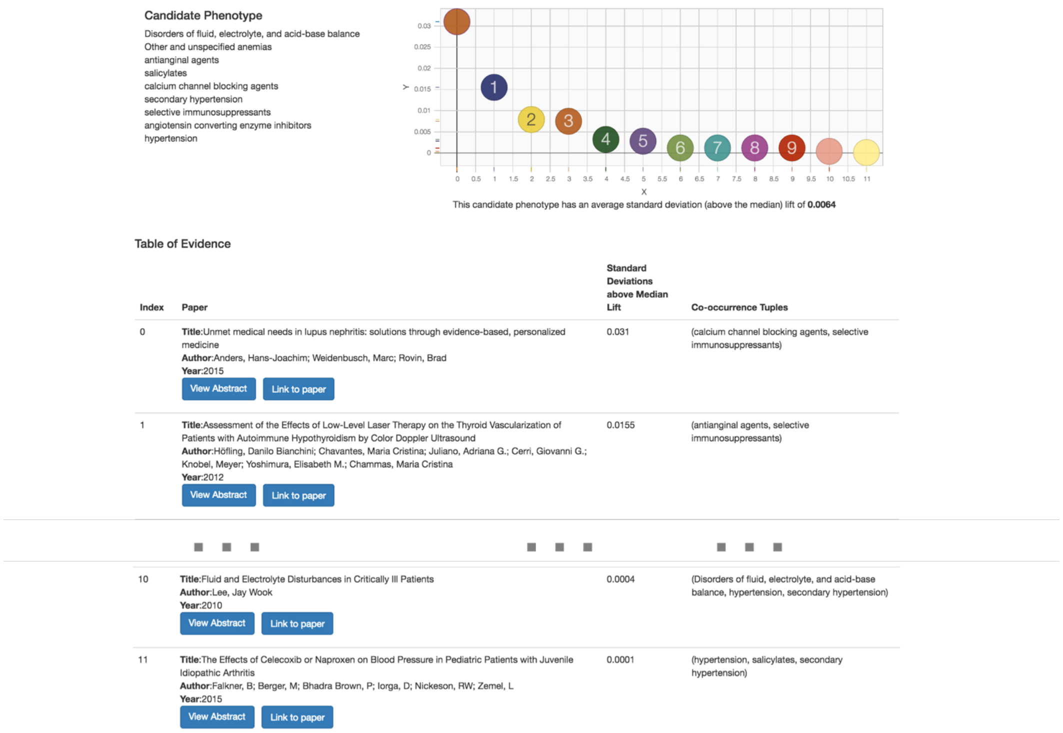

Phenotype 33
cardiac dysrythmias
heart failure
atrial fibrillation and flutter
unspecified chest pain
cardiomyopathy
calcium channel blocking agents
loop diuretics
antianginal agents
...
"heart failure"
"antianginal agents"
MeSH
ICD-10
SNOMED-CT
"left heart failure", heart failure", "congestive heart disease", myocardial failure", , ...
"amiodarone 300mg 10ml solution", "form amiodarone", "antianginal agents", " "amiodarone hydrochloride", ",
Ranking synonyms ~~and related~~ concepts
"heart failure", "left heart failure", myocardial failure", "congestive heart disease", ...
"antianginal agents", "amiodarone hydrochloride", "form amiodarone", "amiodarone 300mg 10ml solution"
Pubmed Articles
{"heart failure", "antianginal agents" -> 57 }
{ "heart failure", "vitamins" -> 1.05 }
...
Phenotype
Generation of Candidates for
Ranked n-grams for Phenotypic
Co-occurrence search in Pubmed
Lift for all co-occurrences

Figure 5.2: Co-occurrence and lift analysis process.



Candidate Phenotype

Disorders of fluid, electrolyte, and acid-base balance
Other and unspecified anemias
antianginal agents
salicylates
calcium channel blocking agents
secondary hypertension
selective immunosuppressants
angiotensin converting enzyme inhibitors
hypertension

This candidate phenotype has an average standard deviation (above the median) lift of **0.0064**

Table of Evidence

Index	Paper	Standard Deviations above Median Lift	Co-occurrence Tuples
0	**Title:**Unmet medical needs in lupus nephritis: solutions through evidence-based, personalized medicine **Author:**Anders, Hans-Joachim; Weidenbusch, Marc; Rovin, Brad **Year:**2015 View Abstract Link to paper	0.031	(calcium channel blocking agents, selective immunosuppressants)
1	**Title:**Assessment of the Effects of Low-Level Laser Therapy on the Thyroid Vascularization of Patients with Autoimmune Hypothyroidism by Color Doppler Ultrasound **Author:**Höfling, Danilo Bianchini; Chavantes, Maria Cristina; Juliano, Adriana G.; Cerri, Giovanni G.; Knobel, Meyer; Yoshimura, Elisabeth M.; Chammas, Maria Cristina **Year:**2012 View Abstract Link to paper	0.0155	(antianginal agents, selective immunosuppressants)
	■ ■ ■	■ ■ ■	■ ■ ■
10	**Title:**Fluid and Electrolyte Disturbances in Critically Ill Patients **Author:**Lee, Jay Wook **Year:**2010 View Abstract Link to paper	0.0004	(Disorders of fluid, electrolyte, and acid-base balance, hypertension, secondary hypertension)
11	**Title:**The Effects of Celecoxib or Naproxen on Blood Pressure in Pediatric Patients with Juvenile Idiopathic Arthritis **Author:**Falkner, B; Berger, M; Bhadra Brown, P; Iorga, D; Nickeson, RW; Zemel, L **Year:**2015 View Abstract Link to paper	0.0001	(hypertension, salicylates, secondary hypertension)

Figure 5.3: Screenshot of PheKnow–Cloud search result.

5.1.2.2 PheKnow–Cloud: Back End Process

Using PheKnow–Cloud, a user can sift through curated evidence that can either support or detract from the validity of a candidate phenotype.

We first discuss the motivation for the evidence curation process and then detail the evidence curation and lift calculation process, which is depicted in Figure 5.2.

Having generated phenotypes through machine learning techniques, PheKnow–Cloud uses co-occurrence analysis of PubMed as a way to study and assess the clinical significance of candidate phenotypes. Although the idea of using co-occurrence of terms to examine the relationship of those terms is conceptually simple, there are several challenges that our automated framework must address before the co-occurrence analysis can take place. For one, each phenotype consists of a set of phenotypic items, and the representation of each element of the phenotype is important as it can drastically impact the number of articles returned during the PubMed query. Thus, the co-occurrence search needs to take into account encoding, form/tense, incorrect spellings, capitalization, and regularization as well as be flexible enough that at least a subset of synonyms and concepts related to the phenotypic item will be captured in a query. For example, if "myocardial failure" is a phenotypic item, our method should also know to count "heart failure" when it occurs in an article in PubMed. To this end, the first step of the co-occurrence analysis process is gathering sets of potential synonyms and related terms for each item in a phenotype using several medical ontologies.[3]

We then filter the set of potential synonyms based on the amount of

[3] We primarily used NCBI MeSH terms, SNOMED-CT, and ICD-10 [Wasserman and Wang, 2003].

overlap between PubMed searches on the synonym and the phenotypic item. Based on experimentation (refer to the discussion of Figure 5.4 in the *Results* section), we found that using the six synonyms, or n-grams, with the highest overlap in a search with the phenotypic items resulted in a good trade-off between computational complexity (i.e., the more terms used to represent the phenotypic item, the longer it will take to perform the co-occurrence counts) and representing the phenotypic item well enough to be captured in the co-occurrence analysis. The phenotypic items are thus represented by themselves as well as a list of synonyms and related concepts, which we refer to as the "phenotypic item synonym set."

With the phenotypic synonym sets in hand, we now outline the co–occurrence calculation. For computational reasons, we used a randomly selected subset of 25% of the articles available in PubMed for this analysis. Given the phenotypic item synonym sets, all possible power sets between these synonym sets are computed. The co–occurrences for each power set is then tallied. Thus any appearance of an item from the phenotypic item synonym set counts as an appearance of the phenotypic item. We minimally process the PubMed text but do regularize capitalization and encoding (utf-8), remove words included in NLTK's English stopword list, use a conservative regular expression to remove references (e.g. Smith, et al.), and remove special characters like quotes and parenthesis.

Having counted all co-occurrences, the next challenge is to choose a phenotype significance metric that reflects the strength of association of the phenotypic items overall. We use lift as a measure of significance. Given items a set of items, $I_1, I_2, ..., I_N$, lift is defined as

$$\text{lift}(I_1, I_2, \ldots, I_N) = \frac{P(I_1 \cap I_2 \cap \cdots \cap I_{N-1} \cap I_N)}{P(I_1)P(I_2)\cdots P(I_{N-1})P(I_N)} \tag{5.1}$$

Probabilities are calculated as the number of sentences where the item occurs divided by the total number of sentences. Lift is a widely used metric to measure the statistical independence of object. A lift of greater than 1 suggests a nonrandom relationship [Brin et al., 1997]. Although there are many metrics (e.g., support, gain, certainty, confidence, and coverage) that can help assess the plausibility of relationships between objects, lift has the benefit of being symmetric (i.e., $\text{lift}[A, B] = \text{lift}[B, A]$), and therefore, the order of the objects does not matter [Ventura et al., 2016]. Another metric called leverage also has this symmetric property. However, unlike leverage, lift is not impaired by the "rare item problem," which refers to the property of a metric excluding objects that appear infrequently [Sheikh et al., 2004]. In the OA corpus, phenotypic items appear infrequently, so it is especially important to use a metric that does not suffer from the rare item problem.

Having calculated the lift for each co–occurring set of terms within a phenotype, the next task is to combine the lifts in such a way that will give a measure of the clinical "significance" of a phenotype. Experimentation showed that the size of the co-occurrence tuple is positively correlated to the size of the lift. This suggests that aggregating all the lifts within a phenotype will drown out the lifts of the smaller sets, and based on this observation, the goal of the measure of the overall significance of a phenotype should somehow take into account the size of the phenotypic items subsets (the size we refer to as the “phenotype cardinality”).

To address this problem we calculate measures of significance with respect to the size of co–occurring phenotypic item sets. We first combine the lifts of all the phenotypic item subsets across all phenotypes and then partition the lifts into sets based on the phenotype item cardinality. We then calculate the median and standard deviation of the lifts within these partitioned sets. As a final step, we repartition the subsets of phenotypic items back into the phenotypes to which they belong and calculate the average of the standard deviations above the median. The average standard deviation above the median across all the possible subsets of phenotypic items within a phenotype is used as the measurement of phenotype clinical significance.

After this analysis has been run, it is summarized and presented to the user of PheKnow–Cloud (like in Figure 5.3). In the “Lift” column, we present the standard deviations above the median that the co-occurring phenotypic item tuple has in the analysis. In the *Results* section, we discuss how this analysis can be used to determine whether or not a given phenotype is clinically meaningful.

5.1.2.3 Data: Test Phenotypes

We use two sets of phenotypes to explore and test the potential of PheKnow–Cloud and the phenotype validation framework. The first set consists of annotated results of candidate phenotypes generated by two different unsupervised, high-throughput phenotype generation processes. The first automatic method, Rubik [Wang et al., 2015], generated phenotypes from a de-identified EHR dataset from Vanderbilt University Medical Center with 7,744 patients over a five year observation period. For more details about the pre-processing of the data and phenotype generation, please refer to their paper [Wang et al., 2015]. The authors graciously shared the file with 30 computational phenotypes as well as the annotations of a panel of three domain experts. For each phenotype, each expert assigned one of the following three choices: 1) yes - the phenotype is clinically meaningful, 2) possible - the phenotype is possibly meaningful, and 3) not – the phenotype is not clinically meaningful. The second set of candidate phenotypes was generated by Marble [Ho et al., 2014b] using the VUMC EHR data. The 50 candidate phenotypes that Marble generated were then annotated by two domain experts in a manner identical to above.

We combined the 30 Rubik-generated candidate phenotypes with the 50 Marble-generated candidate phenotypes and used the resulting set of 80 candidate phenotypes in the co-occurrence experiment. Of these 80 phenotypes, the annotators found that approximately 14% are clinically meaningful, 78% are possibly significant and 8% are not clinically meaningful.

The second set of phenotypes consists of randomly generated phenotypes and phenotypes curated to represent known significant clinical narratives. The random phenotypes are generated by randomly selecting phenotypic items from a set of 1000+ phenotypic items generated by Marble/Rubik phenotypes not used in this work. The curated phenotypes were constructed by representing clinical narratives described in Epocrates references[4] and the AHRQ national guidelines[5] using phenotypic items. We randomly generated phenotypes and created phenotypes based on known medical concepts to demonstrate the efficacy of our method.

5.1.3 Experiments and Results

First we used the Marble and Rubik phenotypes that were annotated as either "clinically significant" or "not clinically significant" to determine the optimal size of the phenotypic item synonym set. We performed a grid search over phenotypic item synonym set sizes, calculated the co-occurrence counts for each phenotype, and then used this information to classify annotated phenotypes as clinically meaningful or not (summarized in Figure 5.4). Specifically, Figure 5.4 shows the precision, recall and F1 score for classifying the annotated phenotypes when characterized by different sizes of phenotypic item synonym sets (n-grams). Using six n-grams per phenotypic item to do the co-occurrence analysis resulted in the classification with the best balance

[4] http://www.epocrates.com/

[5] http://www.ahrq.gov/professionals/clinicians-providers/guidelines-recommendations/index.html

between precision and recall (F1 score of 0.87) We note that while two n-grams scored 0.88, the lower precision delivered by this scenario was not desirable.

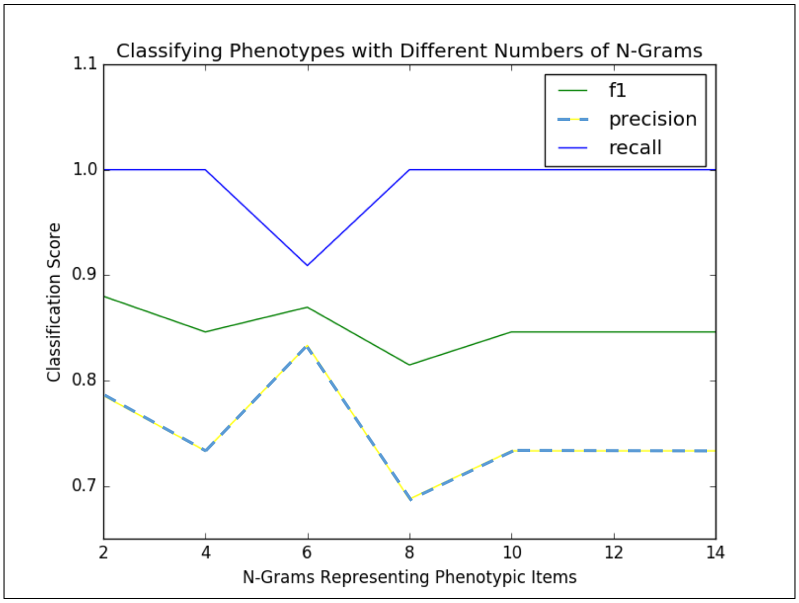


Figure 5.4: Classification Scores for Marble/Rubik Phenotypes versus size of Synonym Set

Using six synonyms or related concepts for each phenotypic item, we examine the lift averages of the randomly generated and curated phenotypes to examine if there is a difference between random and curated phenotypes. Figure 5.5 shows the boxplot of the average standard deviations above the median for the two groups of phenotypes. In nearly all cases lift average of the curated phenotypes is above that of the randomly generated phenotypes, which gives support to the claim that constructing lift in this manner is an effective way of determining the clinical significance of a candidate phenotype.

We then applied this analysis to the candidate phenotypes generated

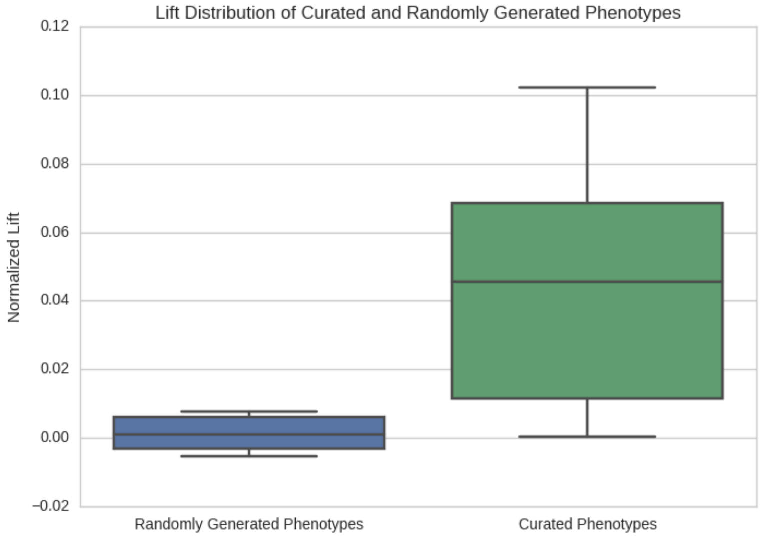


Figure 5.5: Normalized Average Lift of Curated Phenotypes

by Marble and Rubik. Figure 5.6 shows the normalized lift average of the phenotypes generated by Marble and Rubik [Ho et al., 2014a,b; Wang et al., 2015]. If we consider only the candidate phenotypes labeled "significant" and "not significant" by the annotators and draw a boundary at 0.028, we are able to classify candidate phenotypes with an F1 score of 0.87. At this point, we focus on this binary classification task because 1) we consider the annotations to be a "silver" standard ground truth and 2) this binary classification task helps us study the separation between the clinically significant and not clinically significant phenotypes.

This analysis gives support to using lift as a measure of clinical significance of a candidate phenotype.

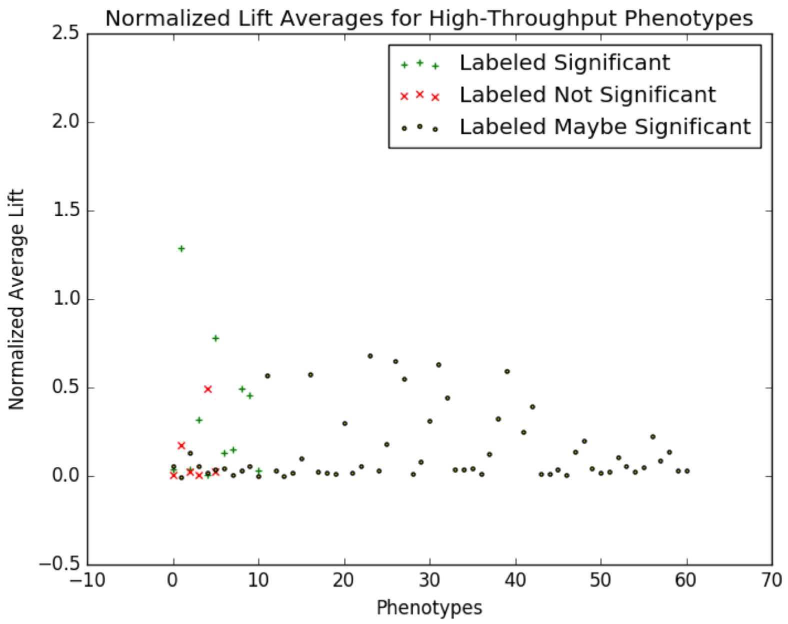


Figure 5.6: Normalized Average Lift of Marble/Rubik Phenotypes

5.1.4 Discussion

PheKnow–Cloud allows users to analyze the evidence behind the lift calculation and assess its validity. For example, in the phenotype depicted in Figure 5.3, a user can examine the evidence given by the co-occurrence tuple "(Disorders of fluid, electrolyte, and acid-base balance, hypertension, secondary hypertension)" by clicking on the associated paper. In that paper, the user would find the sentence, "If urinary K+ excretion is high, transtubular potassium gradient (TTKG), acid-base status, and the presence or absence of hypertension are helpful in differential diagnosis of hypokalemia due to renal potassium loss," which may give support to the candidate phenotype [Lee, 2010]. In the future, we plan to enhance PheKnow–Cloud to highlight the sentences where the terms co-occur.

However, from the PheKnow–Cloud screenshot, we see that the tuple that has the highest standard deviation from the median, is "(calcium channel blocking agents, selective immunosuppressants)," and the paper in which they occur the most is about lupus. The lift captures that they are correlated with one another but maybe not with the phenotype on the whole. This co-occurrence detracts from the body of evidence supporting this phenotype. In the future, we may introduce a semi-supervised aspect to our validation tool where the user can weight tuples they think are the most important.

On the back-end side of things, we note that while lift thresholding classifies phenotypes with relative success in both high-throughput and curated phenotypes, the method does not provide a universal threshold guaranteed for all phenotypes. In addition, the majority of phenotypes are very close to the optimal threshold. In Section 5.2, we build a classification model to overcome this issue.

As noted in Section 5.1.3, the classification task is whether or not an annotated phenotype is clinically significant or not. However, this task leaves out 78% of the compiled phenotypes that were labeled as "possibly significant." One potential use of PheKnow–Cloud is to examine these possibly significant phenotypes as well as incorporate them into the classification task. In Section 4.2, we use PheKnow–Cloud to examine new phenotypes that were annotated as "possibly significant" and identify some as candidates for further study.

5.1.5 Conclusion

When rapidly generating candidate phenotypes in an unsupervised manner, it is necessary to have some measure of their clinical validity and relevance. PheKnow–Cloud is an interactive tool that generates a measure of significance for any proposed phenotype and points to supporting material in the medical literature. PheKnow–Cloud has several potential uses including improving the phenotype verification process and facilitating knowledge discovery by tying evidence across multiple publications. Displaying the results in terms of lift makes one able to quickly analyze which tuples are contributing the most to a phenotype and the associated strength of evidence based on co-occurrence. While PheKnow-Cloud demonstrated that the medical expertise contained in PubMed articles can be harnessed to build evidence sets for the clinical validity of candidate phenotypes, it has a few limitations that we seek to improve. The first limitation is that PheKnow–Cloud functions in a batch setting, which can be useful in some settings (e.g., high throughput phenotyping methods that derive sets of phenotypes) but less useful in other situations (e.g., a clinician is interested in generating an evidence set for a particular phenotype he or she has encountered). The second limitation is the brute-force analysis that PheKnow–Cloud uses is time-consuming to execute. In the next section, we discuss the tool we developed to address these limitations.

5.2 PIVET

5.2.1 Introduction

In this section, we discuss Phenotype Instance Verification and Evaluation Tool (PIVET) [Henderson et al., 2018b]. PIVET is the next iteration of PheKnow–Cloud [Henderson et al., 2017a] introduced in Section 5.1. PIVET is built on the same conceptual framework as PheKnow–Cloud, but in PIVET, we have optimized each piece of PheKnow–Cloud's pipeline to deliver vast improvements in speed and interpretability without sacrificing the integrity of PheKnow–Cloud's phenotype evaluation.

The PheKnow–Cloud pipeline consists of three major steps: (1) representing each phenotype so occurrences of it and related terms in the corpus will be recognized (phenotypic representation), (2) analyzing the corpus using the phenotype representation (corpus analysis), and (3) calculating a clinical relevance score and designation (clinical validity determination). In the phenotype representation step, PIVET uses succinct and possibly more interpretable representations of terms contained within each phenotype. In the corpus analysis step, PIVET migrates from a brute force approach of analyzing the corpus to an approach that uses a NoSQL database to store and index the articles efficiently. PIVET then utilizes a variation of the Aho-Corasick algorithm to count appearances of the terms within each phenotype. Finally, in the clinical validity calculation step, PIVET streamlines the clinical relevance score analysis and uses a model, trained on domain expert-verified phenotypes, to classify the clinical relevance of supplied phenotypes. Through a combination of these improvements, PIVET runs an order of magnitude faster (Table 5.1 shows the speed improvements) than PheKnow–Cloud without sacrificing the discriminative power of the original tool.

PheKnow–Cloud was developed to function in high-throughput phenotyping situations where a researcher has a large set of potential phenotypes to validate. Consequently, PheKnow–Cloud was built to run only in a batch setting. However, in clinical settings and some research settings, a user may only have a few new phenotypes to analyze, so we developed PIVET to run in either an online or batch environment. This improvement will allow clinicians to query PIVET even with single phenotypes, which could possibly help in decision-making processes. Additionally, it could help researchers tune their phenotype extraction algorithms. Thus, while the prototype tool demonstrated the analysis of medical articles could be used to evaluate candidate phenotypes, the improvements in speed and automation realized by PIVET make it useful in both research and clinical settings.

This section is organized as follows. First, we describe the PIVET framework, noting the important differences between PheKnow–Cloud and the new system. We then report the performance of PIVET on automatically generated phenotypes as well as domain expert-curated phenotypes and demonstrate how the framework can be used in an online setting. We conclude the section with a discussion of the limitations of this work and thoughts on future directions.

Table 5.1: The time in seconds and (hours: minutes: seconds) each method used to complete task in phenotype generation process. All experiments were run on a machine with 3 AMD A6-5200 APU with Radeon(TM) HD Graphics processors, 8 GB of memory, 1 TB hard drive, running Ubuntu 14.04.5 LTS.

Task	**PheKnow–Cloud**	**PIVET**
Synonym generation	7,809 (02:10:09)	5,948 (01:39:08)
Cooccurrence analysis	50,822 (14:07:02)	289 (00:04:59)
Lift analysis	2,092 (00:34:52)	2 (00:00:02)
Total	60,723 (16:52:03)	6,239 (01:43:59)

5.2.2 Methods

In this section, we describe how PIVET performs cooccurrence analysis on an online corpus of publicly available journal articles to build evidence sets for phenotypes. This involves five components: (1) a database of phenotypes to analyze, (2) a database of the PubMed article corpus indexed by medical terms the articles contain, (3) an algorithm to generate and rank synonyms for the phenotypic items (phenotypic item representation), (4) a co-occurrence analysis module (corpus analysis), and (5) a clinical relevance scoring system (clinical validity determination). Figure 5.7 captures the PIVET workflow and the different components of the system. Both MongoDB (an open-source, document-based NoSQL database system) and MySQL (an open-source, relational database management system) are used to ensure consistency, durability, and efficiency.

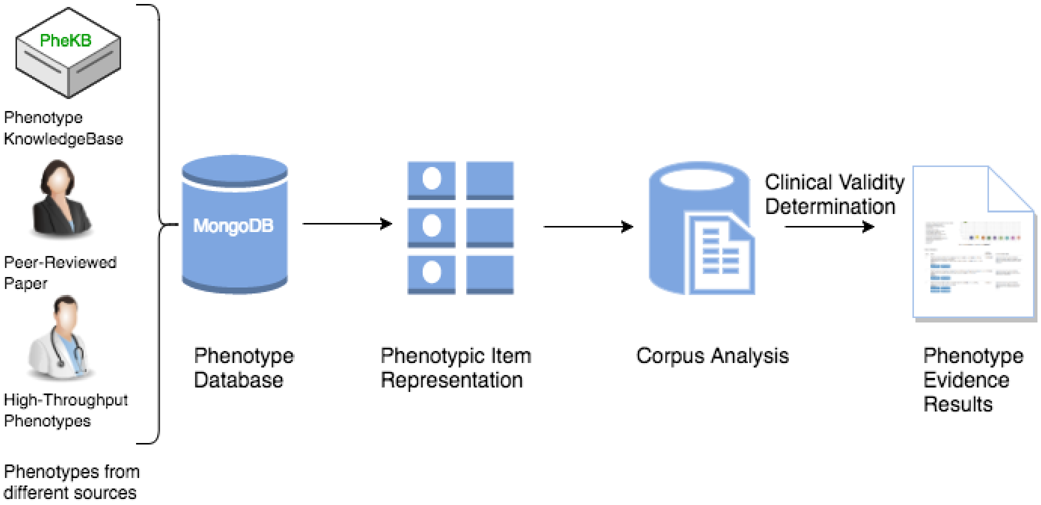

PheKB
Phenotype KnowledgeBase
Peer-Reviewed Paper
High-Throughput Phenotypes
Phenotypes from different sources
MongoDB
Phenotype Database
Phenotypic Item Representation
Corpus Analysis
Clinical Validity Determination
Phenotype Evidence Results

Figure 5.7: Phenotype Instance Verification and Evaluation Tool (PIVET) analysis process. Phenotypes are collected in a standardized format in a MongoDB (i.e., "phenotype database"). For a single phenotype, synonyms for each phenotypic item in a phenotype are generated using the National Library of Medicine (NLM) Medical Subject Headings (MeSH) database and ranked based on their similarity to the phenotypic item (i.e., "phenotypic item representation"). Co-occurrence analysis is performed on PubMed using the synonyms generated in the previous step (i.e., "corpus analysis"). Lift analysis is performed, clinical relevance scores are calculated, and a classifier classifies the phenotype as clinically relevant or not (i.e., "clinical validity determination"). The results of the analysis of the phenotype are presented to the viewer (ie, "phenotype evidence results").

5.2.2.1 Phenotype Extraction and Storage

PIVET can be used to analyze phenotypes generated from a variety of methods. Every phenotype analyzed by PIVET is stored in a MongoDB using a standardized representation to ensure consistency. We also created a simple parser to ingest new phenotypes that are stored in JavaScript Object Notation (JSON). The choice of JSON will also facilitate the eventual integration with a Web platform where users can provide new phenotypes. We populate the phenotype database with phenotypes from different sources (Figure 5.8).

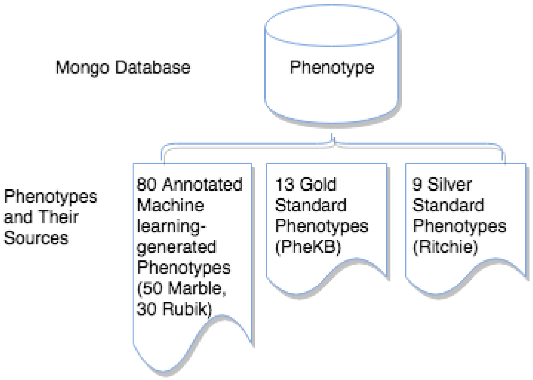

Mongo Database
Phenotype
Phenotypes and Their Sources
80 Annotated Machine learning-generated Phenotypes (50 Marble, 30 Rubik)
13 Gold Standard Phenotypes (PheKB)
9 Silver Standard Phenotypes (Ritchie)

Figure 5.8: Database for storing phenotype information. The large cylinder at the top represents the phenotype database. The phenotype database consists of phenotypes (documents) extracted from three different sources (bottom). The first set of phenotypes, 80 in total, were generated by machine learning algorithms called Marble and Rubik and annotated for clinical relevance by 3 medical doctors. The second set of phenotypes, 13 in total, we refer to as gold standard phenotypes and come from Phenotype KnowledgeBase, an online repository of domain expert-developed phenotypes. The third set of phenotypes, 9 in total, we refer to as silver standard phenotypes and were derived by domain experts and extracted from a peer-reviewed journal article.

For our purposes, we collected a total of 102 phenotypes from the following sources: (1) two high-throughput phenotyping algorithms, (2) a catalog of algorithms from a collaborative database, and (3) a peer-reviewed paper. Each phenotype we extracted was either derived by domain experts or validated as clinically relevant by domain experts.

The phenotype database includes 80 domain expert-verified phenotypes generated by Marble and Rubik used in the preliminary work done in PheKnow–Cloud. Of the 80 combined Marble and Rubik phenotypes, the domain experts labeled 11 (14%) as clinically meaningful, 62 (78%) as possibly significant, and 7 (8%) as not clinically meaningful. For the handful of phenotypes where the domain experts disagreed on the clinical relevance, the label that awarded the least amount of clinical significance was assigned.

Additionally, the phenotype database includes two groups of domain expert-derived phenotypes. The first set, which we will refer to as the "gold standard" phenotypes, are from the Phenotype KnowledgeBase, an online phenotype knowledgebase that stores researchers' collaborations of electronic algorithms of phenotypes [Kirby et al., 2016]. Gold standard phenotypes are developed by panels of domain experts across multiple sites. We manually extracted 13 phenotypes that have been reviewed and finalized by the Electronic Medical Records and Genomics phenotype working group. The second set of domain expert-derived phenotypes, which we will refer to as "silver standard" phenotypes, are the group of validated phenotype algorithms published by Ritchie et al. [2010]. Silver standard phenotypes are developed by a panel of domain experts at a single site. Nine phenotypes were manually extracted from the article. This peer-reviewed paper is not part of the article corpus. In summary, the full set of 102 phenotypes collected over the three different sources consists of 80 machine learning-extracted phenotypes validated by domain experts, 13 gold standard phenotypes, and 9 silver standard phenotypes.

5.2.2.2 PubMed Open Access Corpus

Like PheKnow–Cloud, PIVET works by analyzing cooccurrences of phenotypic items within the PMC OA subset, an openly available online repository of medical articles, which constitutes roughly one-third of the total collection of articles in the PMC (over 1 million articles). The articles within the OA subset are copyright protected but have a flexible license concerning reuse. Trimmed down versions of the articles are stored in a MongoDB. We use the NoSQL database MongoDB because it is a document-based database without restrictive schema, ideal for storing articles that vary in content. Furthermore, MongoDB has been shown to outperform SQL-based databases in terms of read, write, and delete operations and scaling to larger datasets [Li and Manoharan, 2013; Boicea et al., 2012; Indrawan-Santiago, 2012].

We limit the corpus in the database to those articles with attached MeSH terms; this amounts to 379,766 articles. MeSH is a hierarchical vocabulary curated by the NLM to index and catalog biomedical information [Lipscomb, 2000]. There are 26,000 biomedical concepts or headings and over 200,000 supplementary concepts that form qualifiers for the headings. MeSH has two major benefits over the other existing ontologies. First, a large portion of the PubMed corpus has been manually annotated with MeSH labels. Expert indexers at the NLM assign MeSH terms to each article that best summarize the text. These terms are periodically reviewed and updated. We index the PMC database with the MeSH terms each article contains, and we represent each item in a phenotype with a set of MeSH terms, which is discussed in the next section. The index and phenotypic item representation combined with search optimization techniques described in the subsequent section speed up the co-occurrence analysis process considerably.

5.2.2.3 Phenotypic Item Representation: Constructing Medical Subject Headings Synonym Sets

Once the phenotypes are stored in the database, the next step is to build representations for the phenotypic items within each phenotype. PheKnowCloud built representations for each phenotypic item from related terms and concepts found in the following medical ontologies: MeSH, Systemized Nomenclature of Medicine-Clinical Terms, and International Classification of Diseases-9 or -10. Further experiments indicated this approach can introduce noise into the representation. Instead, PIVET uses only MeSH terms to generate a phenotypic item representation for each phenotypic item with the following two-step process: (1) assign the most relevant MeSH term and (2) generate a ranked list of closely related MeSH terms.

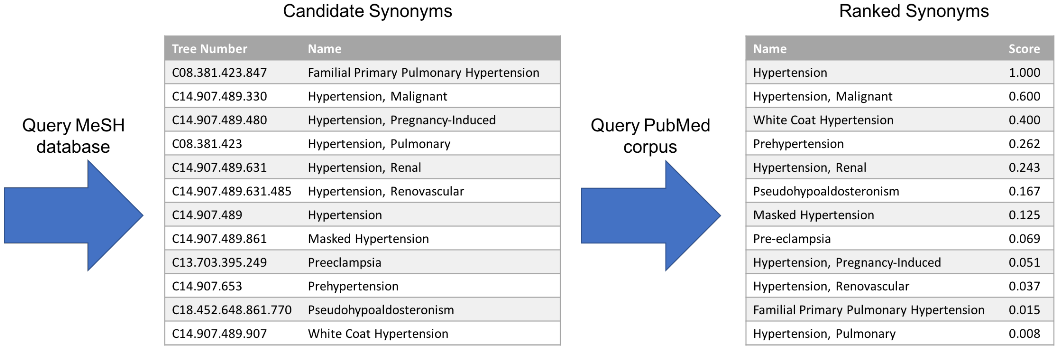
To generate a candidate set of representations for a phenotypic item, PIVET first queries the NLM MeSH database using Biopython [Cock et al., 2009] with a cleaned version of the phenotypic item. The search returns a set of MeSH tree numbers. MeSH terms are formed into a hierarchical tree, where each MeSH term is assigned a node in the tree and labeled by a number. This number designates the MeSH term's place in the hierarchy. For example, the tree number of "hypertension" is C14.907.489, which indicates that it is a child of the node C14.907 ("vascular diseases'). Vascular diseases is in turn a child of node C14 ("cardiovascular diseases"). Gathering nodes with the prefix C14.907.489 gives a set of possible synonyms for the original phenotypic item "hypertension." Generally, this hierarchy gives a relatively straightforward method for finding synonyms and relevant concepts.

As the query does not rank the results (i.e., it does not designate which tree number is most relevant to the search), it is necessary to identify the MeSH term that most closely matches the phenotypic item. For example, querying the phenotypic item "hypertension" returns the tree numbers that map to the natural language headings: "hypertension, malignant"; "hypertension, portal"; "hypertension, pulmonary"; "hypertension, renal"; "hypertension"; "masked hypertension"; "prehypertension"; etc. (shown in Figure 5.9). PIVET designates the "most relevant synonym" for the original phenotypic item by finding the natural language heading associated with each of the tree numbers that most closely matches the original phenotypic item. Specifically, for each natural language heading or synonym, PIVET forms a set where each element is a word of the synonym and then finds the size of the intersection between the set and the original cleaned item, which has also been turned into a set. It also records the size difference between the two sets. For example, the phenotypic item "hypertension" and candidate synonym "hypertension, malignant" have an intersection of length one (i.e., "hypertension") and a size difference of 1. However, PIVET would assign "hypertension" as the most relevant synonym because it has an intersection of size 1 and a set size difference of 0 with the original phenotypic item. In the event of a tie, the algorithm designates the tied candidate synonyms as the most relevant synonyms and builds the synonym sets for each.

The remaining synonyms are then ranked based on the percentage overlap between each candidate synonym and the most relevant synonym in our PubMed OA corpus. The percentage overlap, calculated as the number of times the candidate synonym appears with the most relevant synonym divided by the number of times the candidate synonym appears overall, serves as the relevance score to rank each synonym. The ranked list is then used to adjust the number of synonyms. An example of a ranked synonym set can be seen in Figure 5.9.

5.2.2.4 Corpus Analysis

The aim of the corpus analysis step is to gauge the strength of the relationship between items in a phenotype. However, it is unlikely all items in a phenotype will appear together, so instead, PIVET searches the corpus for occurrences of subsets of the phenotypic items (represented by their phenotypic item MeSH synonym sets as described in the last section). Through experimentation, we found only a small fraction of subsets of any phenotype occur in the article corpus. This means it is inefficient as well as computationally infeasible for even moderately sized phenotypes to look for all possible subsets (i.e., the power set in this case has $2^{|S|}$ elements, where $|S|$ is the cardinality of the phenotype and is the synonym set size for phenotypic item).



Candidate Synonyms

Tree Number	Name
C08.381.423.847	Familial Primary Pulmonary Hypertension
C14.907.489.330	Hypertension, Malignant
C14.907.489.480	Hypertension, Pregnancy-Induced
C08.381.423	Hypertension, Pulmonary
C14.907.489.631	Hypertension, Renal
C14.907.489.631.485	Hypertension, Renovascular
C14.907.489	Hypertension
C14.907.489.861	Masked Hypertension
C13.703.395.249	Preeclampsia
C14.907.653	Prehypertension
C18.452.648.861.770	Pseudohypoaldosteronism
C14.907.489.907	White Coat Hypertension

Ranked Synonyms

Name	Score
Hypertension	1.000
Hypertension, Malignant	0.600
White Coat Hypertension	0.400
Prehypertension	0.262
Hypertension, Renal	0.243
Pseudohypoaldosteronism	0.167
Masked Hypertension	0.125
Pre-eclampsia	0.069
Hypertension, Pregnancy-Induced	0.051
Hypertension, Renovascular	0.037
Familial Primary Pulmonary Hypertension	0.015
Hypertension, Pulmonary	0.008

Figure 5.9: Synonym generation process for the term “hypertension.” First the National Library of Medicine (NLM) Medical Subject Headings (MeSH) database is queried with the term “hypertension,” which returns a list of candidate MeSH terms. From this query result, the “most relevant synonym” is determined through a process of string matching between the original queried term and the candidate synonyms. In this case, the most relevant synonym is “hypertension.” The candidate synonyms are then ranked based on the percentage overlap between PubMed articles that contain the MeSH term associated with the candidate synonym and the MeSH term of the most relevant synonym.

Moreover, as the size of the subset increases, the likelihood of all the terms appearing in any given article diminishes. Therefore, it is not necessary to enumerate all the possible subsets. Using this observation, we implement an algorithm inspired by the string-matching Aho-Corasick algorithm to search the space effectively [Aho and Corasick, 1975], an approach also made popular by the Apriori algorithm for finding association rules in data mining. We sketch the algorithm with a set comprised terms A, B, C, and D that we assume all occur individually in the corpus. We observe that if terms A and B, comprising a tuple (A,B), do not co-occur in any article together, then any larger subset also containing these two terms will necessarily have zero counts (eg, [A,B,C], [A,B,D], and [A,B,C,D]). As a result, only nonzero (feasible) cooccurrence subsets need to be expanded. A key insight for efficient expansion of an existing cooccurrence subset with nonzero counts is to join it with the associated tuple pairs with one overlapping term that have nonzero counts. For example, if the only non-zero tuple pairs are (A,C), (A,D), (B,C), (B,D), and (C,D), then the possible tuples with cardinality 3 are (A,C,D) and (B,C,D). As increasing the cardinality size of the tuple is equivalent to a join operation in a SQL database, PIVET uses MySQL to implement this portion of the analysis. After constructing the query tuples of MeSH terms in MySQL, PIVET then counts the number of articles where each tuple appears.

Additionally, we set a few more restrictions on the subset queries to make them even more efficient. For one, each subset is constructed using "different" phenotypic items to avoid arbitrary inflation of counts. If two or more phenotypic items contain identical MeSH synonym sets, a "super" phenotypic item is formed (e.g., "tuberculosis of adrenal glands" and "tuberculosis of adrenal glands, bacteriological or histological examination not done" are merged together). In addition, terms for the same phenotypic item (e.g., all MeSH terms associated with "myocardial infarction") are never paired with one other.

Given these tuple cooccurrence counts, the next step is to map the cooccurring subsets of phenotypic synonyms back to their phenotypic items. For example, if the synonym set for the phenotypic item "attention deficit disorder" contains two synonym terms "attention deficit and disruptive behavior disorders" and "attention deficit disorder with hyperactivity," then any tuple of cardinality 1 with either of these terms is collected, and the sum of the cooccurrences is then designated as the number of times the phenotypic item "attention deficient disorder" occurred. The aggregated cooccurrence counts for all the nonzero subsets of the phenotypic items are then used to calculate the clinical relevance scores for the phenotype.

5.2.2.5 Clinical Validity Determination

PIVET uses a two-step process to calculate the clinical relevance score: (1) obtain the lift (see below) for each co-occurring subset of phenotypic items and (2) classify the relevance of the phenotype based on features derived from the previous step. As in PheKnow–Cloud, PIVET uses lift to evaluate the strength of the relationship between the items in a phenotype (see Section 5.1.2.2 for definition). In PIVET, the lift calculation (Equation 5.1) entails dividing the percentage of times items appear together in the corpus by the product of percentages of times each item appears individually in the corpus, which can be rewritten as Equation 5.2, where count(A) is the number of articles in the corpus that contain the set A, and D is the number of documents in the corpus.

$$\text{lift}(I_1, I_2, \ldots, I_N) = \frac{\text{count}(I_1, I_2, \ldots, I_N)}{\text{count}(I_1)\text{count}(I_2)\cdots\text{count}(I_N)} D^{N-1} \tag{5.2}$$

It was observed in PheKnow–Cloud that the lift increases exponentially with the size of the cooccurrence set [Bridges et al., 2016; Henderson et al., 2017a]. This is consistent with Equation 5.2. For example, if a set of six items appears together then the fraction of counts will be multiplied by the size of the corpus to the fifth power. These lifts of larger cooccurring subsets drown out the lifts of smaller-sized subsets, which is not necessarily desirable. Thus, we must "normalize" the cardinality of cooccurrence sets. To this end, PheKnow–Cloud calculated the lift for any subset that occurred in the corpus without regard to whether the subset occurred in a phenotype, separated the lifts by the cardinality of the subsets, computed the SDs above the median within that cardinality, aggregated all the SDs above the median values back into the respective phenotypes, and averaged the SD values for each phenotype. This average served as the "clinical relevance score" for that phenotype. This implies that the relevance score will vary depending on the phenotype corpus, as phenotype scores are relative to other candidate phenotypes.

PIVET mitigates this issue inherent to PheKnow–Cloud normalization by including the number of tuples with zero cooccurrences. The number of subsets that had zero occurrences in the corpus is calculated using a simple combinatorial formula as shown in Equation 5.3, where S^j is the number of phenotypic items in phenotype j.

$$\text{Size(zeros for phenotype } j) = \sum_{i=1}^{S^j} \binom{S^j}{i} - \text{size(cooccurrences of cardinality } i) \tag{5.3}$$

Including the zero occurrence counts for each cardinality pulls down the overall lift of the larger items (as it is improbable that large subsets of the phenotype will occur) and thus mitigates the impact of larger cooccurring subsets. Consequently, PIVET avoids the need to pool the phenotypic items across all the phenotypes and avoids unnecessary cooccurrence queries for tuples that do not occur in a phenotype. Perhaps more importantly, this implies that the relevance score is decoupled from the phenotype corpus and can be computed independently for a given phenotype.

The final step in the process is to classify the relevance of the phenotype. We compared four separate classification models: logistic regression, logistic regression with least absolute shrinkage and selection operator (lasso), ridge logistic regression, and k-nearest neighbors (k-NN) on the entire phenotype corpus to predict clinically significant vs not clinically significant. Gold and silver standard phenotypes are denoted as clinically significant because of their relatively small numbers. The features we use are lift mean, lift median, and lift SD for each individual cardinality from 1, 2, 3, and 4 (12 features). We also include the overall lift mean, median, and SD (3 features) and the average cardinality of subsets of the phenotype with nonzero cooccurrences (16 features in total). Model-specific parameters (ie, K for k-NN and the regularization parameter for ridge and lasso) are chosen based on the best area under the receiver operating characteristic via five-fold cross-validation.

In summary, the PIVET lift analysis differs from that performed by PheKnow-Cloud in two key ways. First, we eliminate the need to pool the lifts across the entire phenotype corpus, which means that phenotypes can be analyzed on an individual basis. Second, we introduce classification models to determine relevance based on lift-based features, removing the need to perform an exhaustive search to determine the clinical relevancy threshold.

5.2.3 Results

PIVET is evaluated using two different methods. The first compares the new framework with its predecessor, PheKnow–Cloud, on the set of phenotypes PheKnow–Cloud examined. Differences in computation time, synonym generation, and clinical relevance scores are quantitatively and qualitatively examined. This comparison shows that PIVET delivers clinical relevance determination performance comparable with that of PheKnow–Cloud in a fraction of the time. Furthermore, PIVET's performance justifies shifting from the old to the new framework.

In the second set of experiments, we demonstrate the full PIVET framework on the combined set of machine learning-generated phenotypes, gold standard phenotypes, and silver standard phenotypes. This experiment and discussion show how PIVET's classification method can be used to identify clinically relevant phenotypes from the pool of possibly clinically relevant phenotypes.

Table 5.2: Counts of the 80 machine learning-generated phenotypes by clinical relevance annotation category.

Domain expert annotation category	Count, n (%)
Clinically significant	11 (14)
Possibly clinically significant	62 (78)
Not clinically significant	7 (8)

5.2.3.1 PheKnow–Cloud and PIVET Comparison

5.2.3.1.1 Phenotypic Item Representation

A subset comprising one-quarter of the PMC OA corpus is used to compare our framework's use of MeSH terms for the phenotypic item synonym sets with PheKnow–Cloud's phenotypic item synonym sets. This subset is identical to the one used in the original evaluation of PheKnow–Cloud (see [1] for more details regarding the construction of the dataset). We limit this subset to articles with MeSH terms, which results in a corpus of articles that comprises 7.85% of the PMC OA subset (94,673/1,206,506). We restrict the phenotypes in question to the 80 domain expert-verified, machine learning-generated phenotypes used in the original validation framework (See [Bridges et al., 2016; Henderson et al., 2017a] or Section 5.1). Table 5.2 shows the clinical validity annotations of these 80 phenotypes.

PIVET takes less than 2 hours to evaluate 80 phenotypes on the 8% PMC OA subset; PheKnow–Cloud required 17 hours for the same phenotypes. Note that this is the time it took for PheKnow–Cloud to analyze articles that had MeSH terms associated with them, which is a subset of the corpus analyzed in Bridges et al. [2016]; Henderson et al. [2017a]. The breakdown of the computation time for the major components of the two frameworks is shown in Table 5.1. The phenotypic item representation process time is roughly the same for both PIVET and PheKnow–Cloud, and querying the NLM MeSH database remains the bottleneck. However, PIVET is 170 and 35 times faster for the corpus analysis and clinical relevance determination steps, respectively. Not only does PIVET provide an overall speedup of 10 times on the same article corpus, but the entire process does not need to be repeated to analyze new phenotypes.

As discussed in an earlier section, the phenotypic item representation is different between the two frameworks. PIVET uses sets of MeSH terms to represent each phenotypic item, whereas PheKnow–Cloud's representative synonym sets are built from several ontologies that include the MeSH terms. Overall, PIVET finds more descriptive, discriminative, and possibly more interpretable representations of phenotypic items, whereas PheKnow–Cloud's synonym sets produced a sizeable number of less descriptive words in comparison. Figure 5.10 shows the top 50 PheKnow–Cloud-generated synonyms that were found in the corpus. Although PheKnow–Cloud excludes the first 30 most common terms from its cooccurrence analysis, the remaining 20 words are not discriminative. For example, the word "diseases" is associated with many of the phenotypic items but is too generic to be a meaningful representation of the items.

Further qualitative evidence of the nonspecific nature of the synonym sets produced by PheKnow–Cloud can be found by consideration of examples. Table 5.3 shows the synonyms for the phenotypic item “unspecified chest pain.” Under the PheKnow–Cloud framework, although discriminative terms such as “unspecified chest pain” and “chest pain” are present in the synonym set, the terms “pain,” “chest,” and “unspecified” are words that will be present in many articles that do not actually refer to “unspecified chest pain.” In contrast, under the PIVET framework, the MeSH term for “unspecified chest pain” is “chest pain,” which while less specific than the original term, has the advantage that it will only be found in articles that mention chest pain.

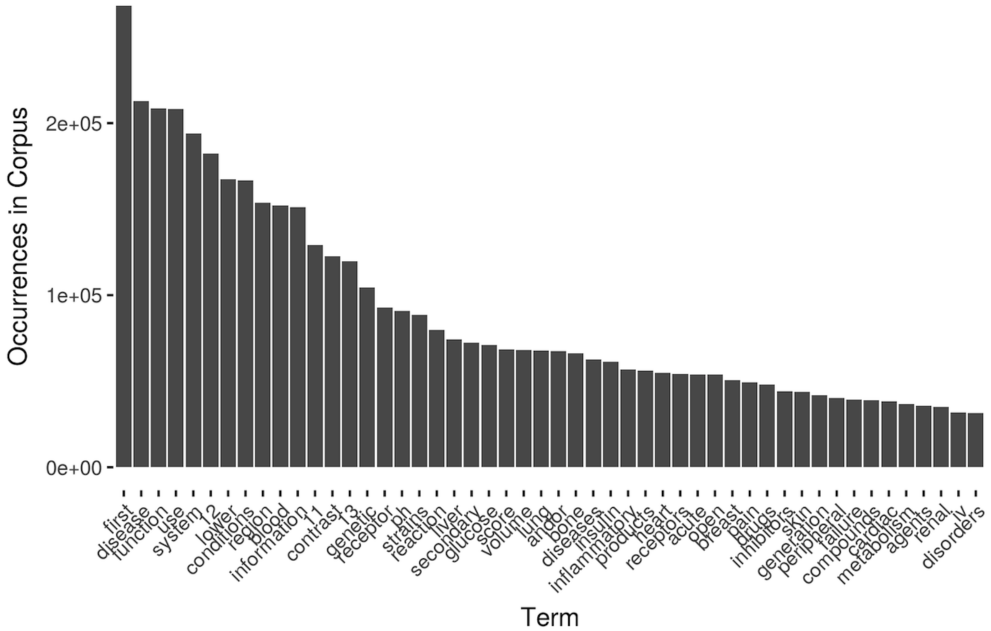


Figure 5.10: Most common synonyms found in corpus using PheKnow–Cloud synonym generation process.

In some cases, the synonym sets are reasonable representations of the

Table 5.3: Comparison of representation of the phenotypic item "unspecified chest pain" generated by PheKnow–Cloud (left column) and Phenotype Instance Verification and Evaluation Tool (PIVET; right column).

PheKnow–Cloud (synonyms)	PIVET (MeSH terms)
Unspecified chest pain	Chest pain
Chest pain	
Uspecified chest	
Pain	
Chest	
Unspecified	

Table 5.4: Comparison of representation of the phenotypic item "laxatives" generated by PheKnow–Cloud (left column) and Phenotype Instance Verification and Evaluation Tool (PIVET; right column).

PheKnow–Cloud (synonyms)	**PIVET (MeSH terms)**
Laxatives	Laxatives
Laxatives pharmacological action	Senna extract
Psyllium	
Senna	
Senna extract	

item and similar for both frameworks. For example, PIVET and PheKnow–Cloud can capture the meaning of the phenotypic item "laxatives" (shown in Table 5.4). PheKnow–Cloud extracts synonyms that are close literal matches to the phenotypic item or specific kinds of laxatives. Similarly, PIVET finds a MeSH term that is an exact match to the phenotypic item and a specific example of the phenotypic item. When looking through the corpus for occurrences of the original term "laxatives," both frameworks should recover mentions of the original term.

Table 5.5: Number of articles that each framework's synonym generation process found.

Synonym type	**Number of articles**
PIVET	28,068
PheKnow–Cloud	79,786
PIVET and PheKnow–Cloud	23,901

5.2.3.1.2 Clinical Validity Determination

Next, we examine how PIVET's phenotype representation compares with that of PheKnow–Cloud in terms of identifying clinically relevant phenotypes. To do this, we instrumented PIVET to record cooccurrences in the same manner as PheKnow–Cloud. Table 5.5 summarizes the number of articles that are found under each framework. Although the PIVET MeSH representation identifies significantly fewer articles from the corpus, the articles have an 85% overlap with PheKnow–Cloud articles. In conjunction with Figure 5.10 and Table 5.2, the results suggest that not all of the PheKnow–Cloud articles are relevant or directly related to the phenotypic item. Thus, PIVET synonym sets may result in higher precision.

Finally, we compared the two frameworks' ability to discriminate between clinically significant and not significant phenotypes using the process PheKnow–Cloud used. To do this, we first calculated the normalized lift for all the phenotypes using the synonyms sets generated by PheKnow–Cloud and PIVET. Figure 5.11 plots the pooled normalized lift values for the 80 phenotypes based on the annotated significance level. As we saw in the PheKnow–

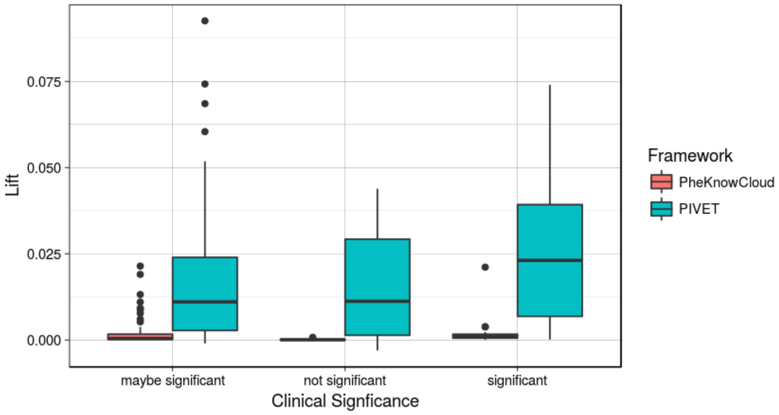


Figure 5.11: Normalized lift comparison between Phenotype Instance Verification and Evaluation Tool (PIVET) and PheKnow–Cloud. Normalized lift is calculated as follows: the lift for any subset of phenotypic items that occurred in the corpus without regard to whether the subset occurred in a phenotype is calculated. Then the lifts are separated by the cardinality of the subsets, and the standard deviations above the median within that cardinality is computed (i.e., this is the normalized lift). The boxplot depicts the normalized lift for the subsets that appeared in each type (ie, "maybe significant," "not significant," and "significant") of phenotype.

Cloud framework, under the PIVET representation, the distributions of normalized lift between significant and not significant phenotypes are not identical, which indicates that lift scores can be used to discriminate between significant and not significant phenotypes.

In the final step, we calculated clinical validity scores for each phenotype by taking the average of the normalized lift scores in each phenotype. An exhaustive search was performed on the clinical validity scores to determine the boundaries for PIVET and PheKnow–Cloud, which was the method used in PheKnow–Cloud that maximized the F1 score. F1 is computed as

Equation 5.4:

$$\mathrm{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.4}$$

We obtained an F1 score of 0.85 and 0.89 for PIVET and PheKnow–Cloud, respectively. Although the predictive performance of PIVET is slightly lower than that of PheKnow–Cloud, the performance loss is negligible when compared with the total run time of each framework (Table 5.1) on 8% of the PMC OA subset. Moreover, by mapping directly to MeSH terms, PIVET can leverage the "automatic" assignment of MeSH terms for all articles and can have a higher probability of capturing appearances of the original phenotypic item in the corpus.

5.2.3.2 Phenotype Instance Verification and Evaluation Tool

In the first set of experiments, we demonstrated PIVET's synonym generation process results in discriminative performance comparable with that of PheKnow–Cloud in a fraction of the time. In the second set of experiments, we use PIVET's full framework (Figure 5.7) to predict which phenotypes are clinically valid and show how PIVET can be used to examine phenotypes that are possibly clinically valid.

5.2.3.2.1 Corpus Analysis: Classification Score Evaluation

We evaluated the ability of the PIVET classification system to identify clinically significant phenotypes. The entire phenotype corpus, including the gold and silver standard phenotypes, were analyzed using the entire PMC OA corpus. There is ambiguity regarding the “possibly significant” Marble and Rubik phenotypes, and they were therefore excluded from the training set. Thus, a total of 45 phenotypes were used to build the classifier, with 7 annotated as not significant.

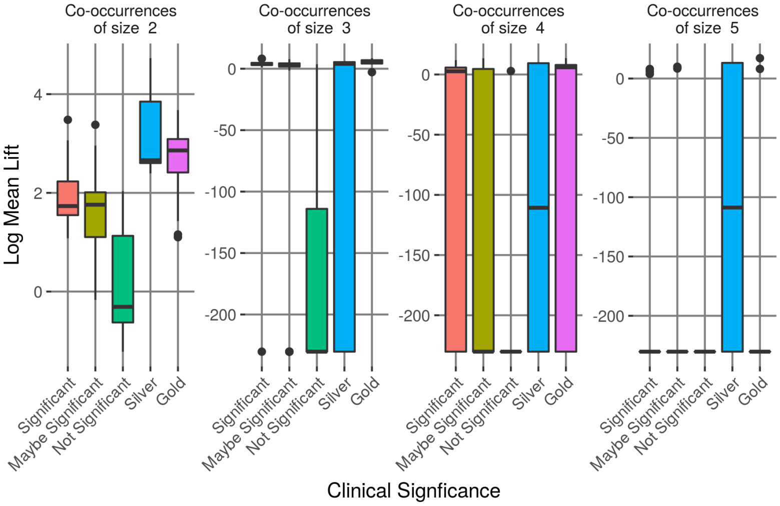


Figure 5.12: Log mean lift for co-occurrences of sizes 2, 3, 4, and 5 for each type of phenotype.

The diversity of the phenotypes in our corpus yielded phenotypes that contained anywhere from 3 to 63 phenotypic items. The size of the phenotype sets impacted the cardinality of the nonzero cooccurrence tuples; thus, we limited the lift summary features to only include tuples up to 4 (the average across the phenotype corpus). Figure 5.12 illustrates the differences in the mean lift values between the various categories, with the gold and silver standard separated from the clinically significant group. The results show that the phenotypes that are clinically significant exhibited a higher (more positive) distribution in lift mean compared with the nonsignificant phenotypes. Moreover, for cooccurrence cardinality less than 5, gold standard phenotypes generally had a higher lift. The figure suggests it is suitable to use the mean lift of tuples of cardinalities 2, 3, and 4 as individual features to distinguish the clinical significance of a phenotype.

Next, we used logistic regression to analyze the effect of the size of the synonym. For each synonym set size ranging from 2 to 10, we used five-fold cross-validation to examine how the size of the synonym set generalizes to an unseen dataset for different metrics. Figure 5.13 plots the average precision, recall, and F1 score as a function of the synonym set size. The figure shows significant increases for all three metrics at synonym size 6, at which point an F1 score of 0.89, recall rate of 0.89, and a precision score of 0.88 are achieved. On the basis of these results, we used six synonyms for each phenotypic item for the remaining analysis.

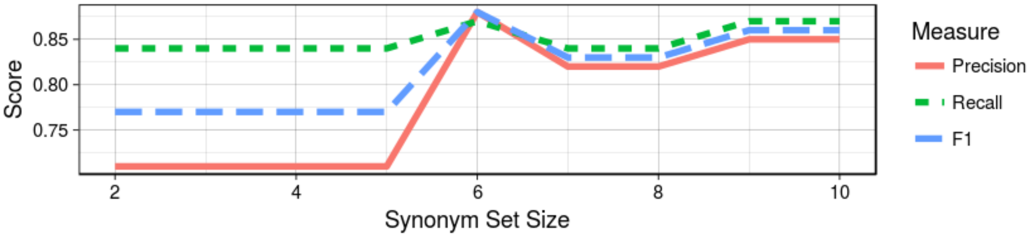


Figure 5.13: Classification scores for different sizes of synonyms using the Phenotype Instance Verification and Evaluation Tool (PIVET) framework.

We repeated the classification process using four models (logistic re-

Table 5.6: Performance metrics for classification task to identify clinically relevant phenotypes using synonym sets of size 6.

Metric	Logistic regression	K-nearest neighbors	Lasso	Ridge regression
Area Under the Receiver Operating Curve	0.79	0.72	0.33	0.6
F1	0.87	0.9	0.77	0.91

gression, k-NN, logistic regression with lasso, and ridge-regularized logistic regression) with six MeSH term synonyms for each phenotypic item. Of the four classification models, ridge regression achieved the highest F1 score of 0.91 and an Area Under the Receiver Operating Curve score of 0.60. On the basis of these results, we use ridge regression as our classification model for the remaining results (Table 5.6). Incorporating a classification model into the framework is an improvement over PheKnow–Cloud, which depended on an exhaustive search to obtain a boundary between clinically relevant and not clinically relevant phenotypes.

5.2.3.2.2 Clinical Validity Determination: Phenotype Instance Verification and Evaluation Tool Analysis of Possibly Clinically Significant Phenotypes

We demonstrate the potential of using PIVET to annotate phenotypes by examining the 62 “possibly clinically significant” phenotypes in our phenotype dataset. Using the PIVET classification ridge model, we predicted the clinical relevance scores of these ambiguous phenotypes. Table 5.7 shows the two extremes based on the averaged prediction score: phenotypes with the highest probability of being “clinically significant” (top two rows) and phenotypes

Table 5.7: Diagnoses and medications for candidate phenotypes along with domain expert annotations, classification score, and lift for two possibly significant phenotypes with high (top two rows) and low (bottom two rows) classification scores.

Diagnoses	Medications	Comment	Score	Lift
Hypotension, heart failure, cardiac dysrhythmias, unspecified chest pain, ischemic heart disease, hypertension, cardiomyopathy	Statins, proton pump inhibitors, gabapentin, noncardioselective beta blockers, sodium, group v antiarrhythmics, potassium-sparing diuretics	The arrhythmic heart patient	1	317.38
Disorders of fluid, electrolyte, and acid-base balance; other and unspecified anemias; hypertensive chronic kidney disease; hypertension; diabetes mellitus; type 2; other disorders of kidney and ureter; chronic kidney disease	Antiadrenergic agents, centrally acting, angiotensin receptor blockers, angiotensin converting enzyme inhibitors, selective immunosuppressants, loop diuretics, gabapentin	Heading toward dialysis	0.999	24683.383
Volume depletion; dehydration, nausea, or vomiting; hypopotassemia; abdominal pain	Heparins, antihistamines, 5HT3 receptor antagonists, minerals and electrolytes, narcotic analgesic combinations, proton pump inhibitors	Gastroenteritis	0.418	0.27
Disorders of fluid, electrolyte, and acid-base balance; other diseases of lung; hypotension; pleurisy, atelectasis, and pulmonary collapse; unspecified chest pain; o ther disorders of the kidney and ureter	Anticholinergic bronchodilators, loop diuretics	Lung diseases?	0.417	0.509

with the lowest probability of being "clinically significant" (bottom two rows), as well as the annotator's comment on the phenotype and the average lift calculated by PIVET. The prediction scores seem to reflect the annotator's certainty, as the lowest prediction score is associated with a question mark, whereas the top two scoring phenotypes seem to capture a relevant concept. The results underscore the potential of PIVET system to help resolve uncertainties.

5.2.4 Conclusion

5.2.4.1 Principal Findings

The potential for computational phenotyping to help physicians reason about patient populations will only be realized if the phenotypes generated are clinically meaningful. To increase the utility of such data-driven phenotype discovery, some measure of inferred clinical meaningfulness should be reported to help clinicians sort the signals from the noise. We developed PIVET to meet this need. PIVET generates evidence sets and clinical relevance scores for data-driven candidate phenotypes using the literature available in PubMed, a large online repository of biomedical articles.

We compared our framework with PheKnow-Cloud, its predecessor, and showed that PIVET improves the run time dramatically. In addition to scaling up to the entire PMC OA corpus, PIVET can analyze phenotypes individually and automatically assign clinical relevance scores that are independent of the other phenotypes in the corpus. Furthermore, there was anecdotal evidence that the PIVET synonym generation process was more discriminative and meaningful than its PheKnow-Cloud counterpart. In the future, one goal is to make PIVET available to researchers and clinicians. To this end, we plan to deploy a live version of the phenotype parser that users can interact with via a REST API and receive phenotype JSON files in return. We are currently investigating the best way to release PIVET for general use.

5.2.4.2 Possible Use Cases

For researchers developing models and algorithms to automatically extract phenotypes from EHRs without supervision, all phenotypes are possibly clinically significant before they have been validated. We envision PIVET being used by researchers to gain understanding into the phenotypes they have extracted. Outside a machine learning setting, there are several potential uses for PIVET. For example, a pharmaceutical company may uncover a potentially interesting pathway analysis or phenotype, and they can use PIVET to identify all the articles that have been previously published on the subject, as well as PIVET's clinical validity determination to decide if the pathway is worth pursuing and how much it can be trusted. Similarly, in a healthcare setting, a clinician could encounter an interesting group of patients and use PIVET to explore what pathways have been discovered with relation the set of patient characteristics. As in the pharmaceutical setting, PIVET's ability to deliver a clinical validity determination, as well as generate a body of evidence in the form relevant articles, can help clinicians reason about the patterns they encounter on a daily basis.

In Section 4.2 we use PIVET to analyze phenotypes resulting from tensor factorization with constraints on the patient mode. In Chapter 6 we incorporate lift information from PIVET into the tensor factorization process to guide the decomposition to potentially more meaningful phenotypes.

5.2.4.3 Limitations

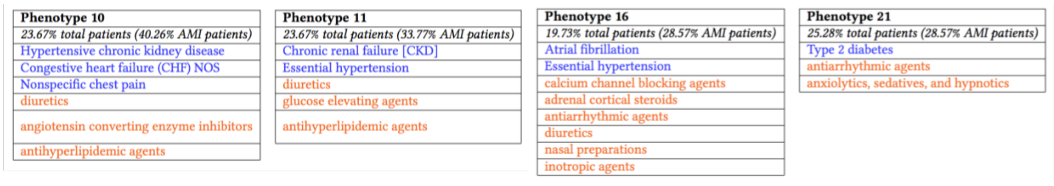
One possible way to improve PIVET is to include more phenotypes when training the classifier. We continue to gather additional domain expert annotated phenotypes to include in the framework. One limitation of the current analysis was that all the gold and silver standard phenotypes were combined with the domain expert-labeled examples for classification purposes.

As we continue to gather more gold and silver phenotypes, we plan to refine the classification process by incorporating this annotation quality information. We also plan to test new sets of features that incorporate interaction between the lift statistics and to examine different metrics for evaluating the clinical significance of candidate phenotypes.

Chapter 6

Guiding the Phenotyping Process

One possible weakness of using CP decomposition is that there can be noise between and across the modes (i.e., elements appear together that do not belong together). In computational phenotyping, we have found that this noise can manifest as a medication and diagnosis co-occurring in a component when they do not actually have a clinical relationship. This weakness can degrade the interpretability of the fits. Incorporating domain expertise into the tensor decomposition may help overcome this weakness. However, few tensor decomposition methods have used supervision or domain expertise to increase the number of interpretable components (see [Wang et al., 2015] for an example), and like many problems in machine learning, incorporating supervision can be challenging and costly in terms of the time and domain expertise necessary for gathering labels or domain-specific constraints. However, there are many sources of publicly available information (e.g., census data, online journals, and forums) that can serve as weak proxies for domain expertise to inform the problem at hand. In Section 6.1, we show how to incorporate insights learned from PIVET in the tensor factorization process to guide components to potentially more meaningful phenotypes. Then, in Section 6.2, we show how to learn candidate cannot-link constraints during the fitting process and



Phenotype 10
23.67% total patients (40.26% AMI patients)
Hypertensive chronic kidney disease
Congestive heart failure (CHF) NOS
Nonspecific chest pain
diuretics
angiotensin converting enzyme inhibitors
antihyperlipidemic agents

Phenotype 11
23.67% total patients (33.77% AMI patients)
Chronic renal failure [CKD]
Essential hypertension
diuretics
glucose elevating agents
antihyperlipidemic agents

Phenotype 16
19.73% total patients (28.57% AMI patients)
Atrial fibrillation
Essential hypertension
calcium channel blocking agents
adrenal cortical steroids
antiarrhythmic agents
diuretics
nasal preparations
inotropic agents

Phenotype 21
25.28% total patients (28.57% AMI patients)
Type 2 diabetes
antiarrhythmic agents
anxiolytics, sedatives, and hypnotics

Figure 6.1: PIVETed-Granite phenotypes derived from a tensor constructed from VUMC patient-level data. These phenotypes have high membership of patients who had at least one myocardial infarction.

then accept or reject these constraints based on evidence found in proxies for domain expertise.

6.1 PIVETed-Granite

6.1.1 Introduction

In this section, we show how to incorporate PIVET, which is described in detail in Chapter 5.2, into the phenotype derivation process. The goal is to increase the number of meaningful components in the CP decomposition process without human input. Our method, which we refer to as PIVETed-Granite [Henderson et al., 2018c], involves a novel application of PIVET to provide side information in the form of a cannot-link matrix between the diagnosis and medication modes of a tensor constructed from a set of EHRs. By automatically leveraging available biomedical literature, PIVETed-Granite enables more concise and diverse phenotypes that are discriminative and interpretable.

6.1.2 Problem Formulation

PIVETed-Granite combines Granite (Chapter 3) with PIVET (Section 5.2). We focus on a 3-mode tensor where the three dimensions are (1) patients, (2) diagnoses, and (3) medications and each element is a count of the number of times a patient received a diagnosis and medication prescription in a given period of time. An observed tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is approximated as the sum of R 3-way rank-one tensors $\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{Z}} = [\![\boldsymbol{\lambda}; \mathbf{A}; \mathbf{B}; \mathbf{C}]\!]$, which are the patient, diagnosis, and medication factor matrices, respectively. To discourage specified diagnosis and medication pairs from appearing together in the same phenotype, PIVETed-Granite introduces a cannot-link matrix between the diagnosis ($\mathbf{B}$) and the medication ($\mathbf{C}$) factor matrices. The optimization problem for the observed tensor $\boldsymbol{\mathcal{X}}$ is:

$$
\begin{aligned}
f(\boldsymbol{\mathcal{X}}) = \min(&\sum_{\vec{i}} (z_{\vec{i}} - x_{\vec{i}} \log z_{\vec{i}}) && (6.1)\\
&+ \beta_1 \mathrm{trace}(\mathbf{B}^\intercal \mathbf{M} \mathbf{C})) && (6.2)\\
&+ \frac{\beta_2}{2} \sum_{r=1}^{R} \sum_{p=1}^{r} \left((\max\{0, \frac{(\mathbf{d}_p)^\intercal \mathbf{d}_r}{||\mathbf{d}_p||_2 ||\mathbf{d}_r||_2} - \theta_{\mathbf{d}}\})^2 \right) && (6.3)\\
&+ \frac{\beta_3}{2} \sum_{r=1}^{R} (||\mathbf{a}_r||_2^2 + ||\mathbf{b}_r||_2^2 + ||\mathbf{c}_r||_2^2) && (6.4)\\
\text{s.t } \boldsymbol{\mathcal{Z}} = &[\![\sigma; \mathbf{u}_a; \mathbf{u}_b; \mathbf{u}_c]\!] + [\![\boldsymbol{\lambda}; \mathbf{A}; \mathbf{B}; \mathbf{C}]\!] && (6.5)\\
&\mathbf{d} \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}\} && (6.6)\\
&||\mathbf{a}_r||_1 = ||\mathbf{b}_r||_1 = ||\mathbf{c}_r||_1 = 1, \mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r \geq 0 && (6.7)\\
&||\mathbf{u}_a||_1 = ||\mathbf{u}_b||_1 = ||\mathbf{u}_c||_1 = 1, \mathbf{u}_a, \mathbf{u}_b, \mathbf{u}_c > 0. && (6.8)
\end{aligned}
$$

For count data, the loss function is KL-divergence (6.1). The cannot-link penalty in Equation 6.2 is discussed in detail in Section 6.1.2.1. An angular penalty term (6.3) discourages any factors from being too similar, where similarity is defined as the cosine angle between two factor vectors, and an ℓ_2 penalty term controls the growth of the size of the factors (6.4) (See Chapter 3 for details).

6.1.2.1 Incorporating PIVET

In Equation 6.2, $\mathbf{M} \in \mathbf{1}^{I_2 \times I_3}$ is a binary cannot-link matrix defined as follows:

$$\mathbf{M}_{jk} = \begin{cases} 1, & \text{if lift}(b_j, c_k) < \alpha) \\ 0, & \text{otherwise} \end{cases}$$

We construct $\mathbf{M}$ with PIVET. PIVET calculates the lift for each (diagnosis, medication) pair (i.e., b_j, c_k) based on analysis of biomedical journal articles. A lift of much greater than 1 indicates diagnosis j and medication k co-occur often and therefore may have a clinical relationship with one another, and a value of 1 or less means diagnosis j and medication k do not co-occur often in the corpus and may not have a clinical relationship. In this work, we use $\alpha = 1$. The terms in Equation 6.2 are of the form $b_{jr}\mathbf{M}_{jk}c_{kr}$, and only contribute to the objective function if the j^{th} diagnosis and the k^{th} medication appear in the r^{th} component. Since Equation 6.2 is a soft constraint, if there is actually a relationship between (b_j, c_k) in the data, they can still appear together in components. However, if the relationship is weak in the data, these elements will be discouraged from appearing together in a phenotype.

6.1.2.2 Minimizing the Objective Function

We solve for $\mathfrak{Z}$ using Stochastic Gradient Descent (SGD) with Adam, which was introduced by Kingma and Ba [2015]. We follow the work on Generalized CP Decomposition presented by Hong et al. [2017] for the implementation. Using SGD to minimize a CP gradient is equivalent to a sparse implementation of CP decomposition where a subset of data points are taken to be the nonzero entries. We use the work of Acar et al. [2011b] to implement operations on sparse tensors.

Section 3.3.2 for derivation of the gradients for Equations 6.1 and 6.3. For Equation 6.2, the derivatives with respect to the factor matrices $\mathbf{B}$ and $\mathbf{C}$ are:

$$\frac{\partial \mathrm{Tr}(\mathbf{B}^\intercal \mathbf{M} \mathbf{C})}{\partial \mathbf{B}} = \mathbf{M}\mathbf{C} \tag{6.9}$$

$$\frac{\partial \mathrm{Tr}(\mathbf{B}^\intercal \mathbf{M} \mathbf{C})}{\partial \mathbf{C}} = \mathbf{M}^\intercal \mathbf{B} \tag{6.10}$$

$$\tag{6.11}$$

6.1.3 Experiments

Dataset Description. To explore the feasibility of using guidance from PIVET, we constructed a tensor from the diagnosis and medication counts of 1622 patients from the Synthetic Derivative (SD), a de-identified EHR database gathered at the VUMC [Roden et al., 2008]. A panel of domain experts developed sets of characteristics (i.e., billing and medical codes) to identify patients as case and control for a set of diseases [Ritchie et al., 2010].

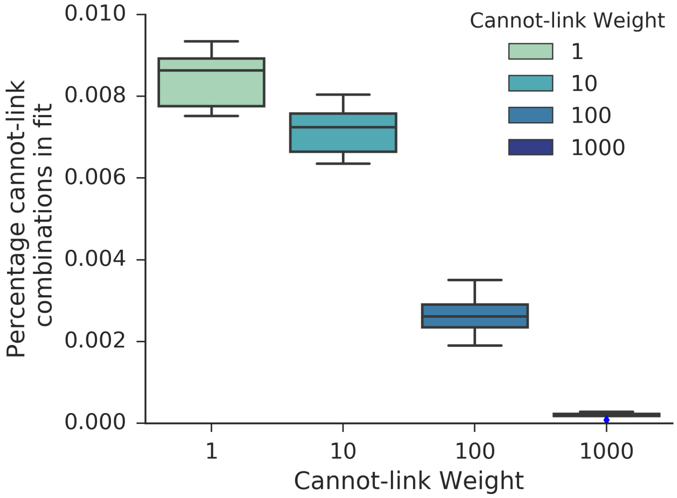
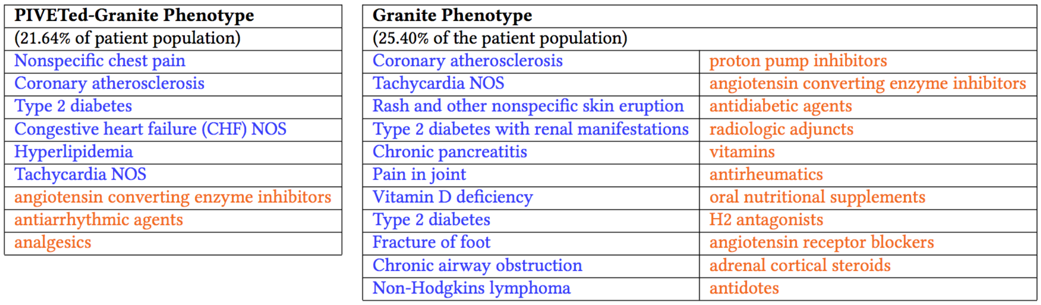


Figure 6.2: Percentage of (diagnosis, medication) cannot-link constraints appearing in the final fit.



PIVETed-Granite Phenotype
(21.64% of patient population)
Nonspecific chest pain
Coronary atherosclerosis
Type 2 diabetes
Congestive heart failure (CHF) NOS
Hyperlipidemia
Tachycardia NOS
angiotensin converting enzyme inhibitors
antiarrhythmic agents
analgesics

Granite Phenotype	
(25.40% of the patient population)	
Coronary atherosclerosis	proton pump inhibitors
Tachycardia NOS	angiotensin converting enzyme inhibitors
Rash and other nonspecific skin eruption	antidiabetic agents
Type 2 diabetes with renal manifestations	radiologic adjuncts
Chronic pancreatitis	vitamins
Pain in joint	antirheumatics
Vitamin D deficiency	oral nutritional supplements
Type 2 diabetes	H2 antagonists
Fracture of foot	angiotensin receptor blockers
Chronic airway obstruction	adrenal cortical steroids
Non-Hodgkins lymphoma	antidotes

Figure 6.3: Two phenotypes, one derived using PIVETed-Granite (left) and one using Granite (right) where both methods were initialized with the same factor vetors.

Using these specifications, we included 304 resistant hypertension case patients and 399 resistant hypertension control patients in the tensor. For each case patient, we counted the medication and diagnosis interactions that occurred two years before they received the hypertension diagnosis. For each non-case patient, we counted the interactions that occurred two years before their last

Table 6.1: Fit information for phenotypes derived using Marble, Granite, and PIVETed-Granite.

		Average Number of Non-Zeros		
Method	**KL-divergence**	**Patient**	**Diagnosis**	
Marble	2803253.42 (194914.35)	26.72 (1.3)	7.01 (0.37)	8.44 (0.22)
Granite	2311866.35 (27826.92)	70.45 (1.69)	18.21 (1.06)	12.31 (0.31)
PIVETed-Granite	2224824.04 (19758.83)	57.21 (2.65)	5.78 (0.2)	5.89 (0.56)

Table 6.2: Cosine similarity of factor matrices derived using Marble, Granite, and PIVETed-Granite.

	Cosine Similarity		
Method	**Patient**	**Diagnosis**	**Medication**
Marble	0.07 (0.01)	0.01 (0.01)	0.24 (0.01)
Granite	0.18 (0.01)	0.02 (0.01)	0.12 (0.01)
PIVETed-Granite	0.20 (0.02)	0.03 (0.02)	0.05 (0.02)

interaction with the VUMC. In their raw form, the diagnosis codes (International Classification of Diseases (ICD-9) system) capture a very detailed level of information. We use PheWAS coding to aggregate the diagnosis codes into broader categories [Denny et al., 2013] and Medical Subject Headings (MeSH) pharmacological terms from the RxClass RESTful API to group the medications into more general categories[1]. These coarser hierarchies produced a tensor with 1622 patients by 1325 diagnoses by 148 medications.

Quantitative Evaluation. We compared PIVETed-Granite to two baseline models, Granite and Marble [Ho et al., 2014b]. Table 6.1 shows fit quality (KL-divergence) and sparsity, and Table 6.2 shows diversity measures for the three models ($R = 30$). PIVETed-Granite was the best fit to the data with the lowest KL-divergence, and also resulted in the smallest number of non-zero

[1] https://rxnav.nlm.nih.gov/RxClassAPIs.html

elements in the diagnosis and medication modes. Additionally, PIVETed-Granite resulted in diagnosis factors that were comparably diverse to those of Granite and medication factors that were more diverse than Granite. We also evaluated the effect of the cannot-link weight β_1 on the percentage of (diagnosis, medication) cannot-link pairs present in the factor matrices. In Figure 6.2, as β_1 increases, the percentage of cannot-link pairs decreases.

Additionally, we evaluated the discriminative capabilities of PIVETed-Granite in a prediction task where the patient factor matrix $\mathbf{A}$ served as the feature matrix. We compared the performance of PIVETed-Granite, Granite, and Marble using logistic regression to predict which patients were hypertension case and control. The model ran with five 80-20 train-test splits, and the optimal LASSO parameter for the model was learned using 10-fold cross-validation. Table 6.3 shows the AUC for PIVETed-Granite, Granite, and Marble. The patient factor matrix derived using PIVETed-Granite resulted in the most discriminative model in this task.

Qualitative Exploration. To evaluate the effect of the cannot-link matrix $\mathbf{M}$ on the decomposition process we initialized PIVETed-Granite and Granite fits with the same factors and then examined the differences between the fitted factors. Figure 6.3 shows one phenotype from each method initialized with the same factors. While the phenotypes are similar to one another, PIVETed-Granite's characteristics form a more succinct, focused characterization of heart disease complicated with type 2 diabetes. Additionally, the Granite phenotype contains many cannot-link combinations (e.g., ("Fracture

Table 6.3: AUC for predicting resistant hypertension case patients.

Method	**AUC (st. dev.)**
Marble	.6656 (.09)
Granite	.7083 (.04)
PIVETed-Granite	**.7172 (.01)**

of foot", "antidotes")) whereas the PIVETed-Granite phenotype does not. The cannot-link constraints seem to result in phenotypes that are descriptive and cohesive.

As a way to qualitatively explore the clinical meaningfulness of the discovered phenotypes, we identified patients who experienced acute myocardial infarctions (AMI), which resulted in a cohort of 77 unique patients within the tensor. In Figure 6.1, we show the phenotypes with the highest proportions of AMI patients. These automatically generated phenotypes seem to give nuanced descriptions of patients who have AMIs. For example, in Phenotype 10 one of the diagnoses is congestive heart failure, which is primarily caused by acute myocardial infarctions [Cahill and Kharbanda, 2017]. Type-2 diabetes patients (Phenotype 21 and 28) are also more likely to experience heart attacks and have more negative outcomes from them [Lago and Nesto, 2009].

6.1.4 Conclusion

Adding guidance in the form of constraints to computational phenotyping models can help improve the quality of the fit and shows promise in increasing the clinical meaningfulness of derived phenotypes. However, obtaining informative constraints can be difficult and expensive in regard to time and effort required by domain experts. We showed how to leverage publicly available information in the form of medical journals to guide the decomposition process to discriminative and interpretable phenotypes. PIVETed-Granite derived phenotypes that were more discriminative, more diverse, and sparser than two competing baseline models. Incorporating cannot-link constraints between modes is a general method that can be applied to many domains. In this application, the quality of the auxiliary information provided by PIVET seems to be high, but in other applications, it may not be. In the next section, we show how to incorporate auxiliary information when that side information is noisy.

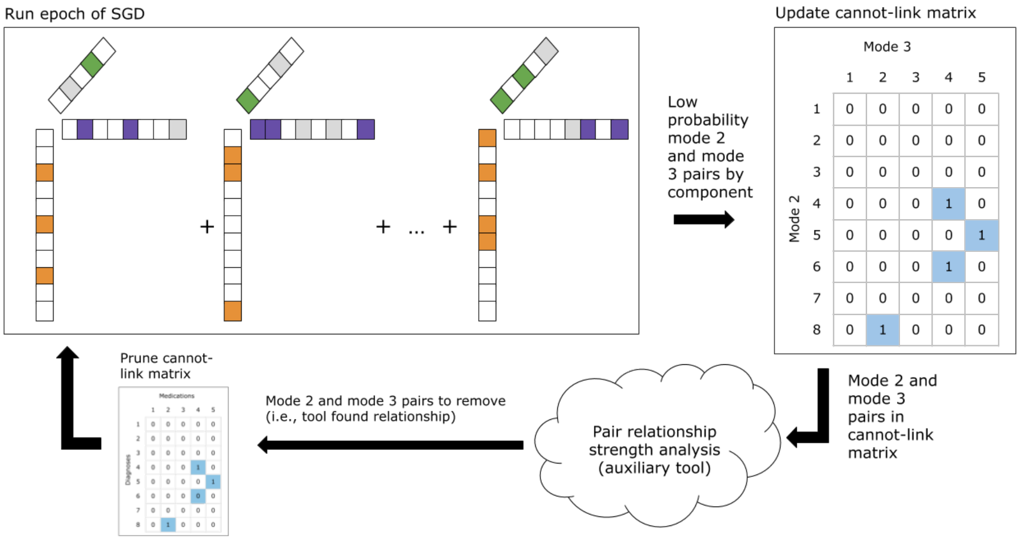
6.2 CP decomposition with Cannot-Link Inter-mode Constraints (CP-CLIC)

6.2.1 Introduction

In Section 6.1, we showed how to build a constraint matrix that could be used in the tensor factorization process to guide components to concise and focused phenotypes that show potential in being clinically interesting. The cannot-link constraint matrix is applicable to many other domains and domain expertise can be extracted from a variety of sources (e.g., census data, online journals, and forums). However, the integrity of the auxiliary information may not be as strong as that that was generated by PIVET. In this section, we show how to learn a cannot-link constraint matrix during the decomposition process and to evaluate the proposed constraints using side information. This model, called CP decomposition with Cannot-Link Inter-mode Constraints (CP-CLIC) [Henderson et al., 2018d], gradually builds cannot-link constraints between different modes during the decomposition process and refines these constraints using domain expertise via proxy information gathered from openly available sources. Using computational phenotyping as an example, CP-CLIC is faster and achieves sparser components compared to the baseline models. CP-CLIC is generalized for many data types and can incorporate both guidance information and a variety of constraints including non-negativity, simplex, and angular, to uncover sparse and diverse factors on a large-sized tensor. Using data simulated from multiple distributions, we demonstrate CP-CLIC can recover components accurately, improving the fits in most cases. We performed a case study of CP-CLIC on computational phenotyping. We show that the CP-CLIC-discovered phenotypes are sparse, diverse, and clinically interesting. Additionally, the meaningfulness of the discovered components increased by 66% over the baseline.

6.2.2 Problem Formulation

Unlike existing constrained tensor decomposition models, CP-CLIC gradually learns constraints about inter-mode relationships within tensors and refines these constraints using auxiliary information. Automatically discovering the constraints reduces the cost of obtaining guidance from domain experts and allows the decomposition to discover the multiway relationships within the data. Implemented using SGD, CP-CLIC scales to large tensors and is formulated to accommodate a large family of objective functions that work in concert with sparsity- and diversity-encouraging constraints to derive meaningful components. In this section, we outline the CP-CLIC formulation and learning process.



Run epoch of SGD
+
+ ... +
Low probability mode 2 and mode 3 pairs by component
Update cannot-link matrix
Mode 3
Mode 2
Mode 2 and mode 3 pairs in cannot-link matrix
Pair relationship strength analysis (auxiliary tool)
Mode 2 and mode 3 pairs to remove (i.e., tool found relationship)
Prune cannot-link matrix
Medications
Diagnoses

Mode 2 \ Mode 3	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	0	0	0	1	0
7	0	0	0	0	0
8	0	1	0	0	0

Diagnoses \ Medications	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	0	0	0	0	0
7	0	0	0	0	0
8	0	1	0	0	0

Figure 6.4: Cartoon illustration of the CP-CLIC process. Outlined items represent an action being taken, while text above arrows represent data moving through the constraint matrix-building process. Starting in the upper lefthand corner, after an epoch of the CP-CLIC SGD fitting process is complete, CP-CLIC finds the elements in modes 2 and 3 of each component that has probabilities below a predetermined threshold (light grey boxes). These (mode 2, mode 3) pairs are given a 1 in the cannot-link matrix. The pairs are evaluated using auxiliary information. If the auxiliary information finds there is a relationship, these pairs are removed from the cannot-link matrix.

More specifically, let $\mathbfcal{X}$ denote an $I_1 \times I_2 \times \cdots \times I_N$ tensor of binary, nonnegative count, or continuous data and $\mathbfcal{Z}$ represent a same-sized tensor where each element $z_{\vec{i}}$ contains the optimal parameters of the observed tensor

$x_{\vec{i}}$. The full objective function of CP-CLIC is as follows:

$$
\begin{aligned}
f(\boldsymbol{\mathcal{X}}) &= \min(\mathcal{L}(\boldsymbol{\mathcal{Z}}|\boldsymbol{\mathcal{X}}) \qquad\qquad (6.12)\\
&+ \beta_1 \sum_{n=1}^{N} \sum_{m=1}^{n-1} \mathrm{Tr}(\mathbf{A}^{(m)\intercal}\mathbf{M}^{(m,n)}\mathbf{A}^{(n)})) \qquad\qquad (6.13)\\
&+ \frac{\beta_2}{2} \sum_{n=1}^{N} \sum_{r=1}^{R} \sum_{p=1}^{r} max(0, \frac{(\mathbf{a}_p^{(n)})^\intercal \mathbf{a}_r^{(n)}}{||\mathbf{a}_p^{(n)}||_2 ||\mathbf{a}_r^{(n)}||_2} - \theta_n)^2 \qquad\qquad (6.14)\\
&+ \frac{\beta_3}{2} \sum_{n=1}^{N} \sum_{r=1}^{R} \|\mathbf{a}_r^{(n)}\|_2^2 \qquad\qquad (6.15)\\
\text{s.t } \boldsymbol{\mathcal{Z}} &= [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots ; \mathbf{A}^{(N)}]\!] \qquad\qquad (6.16)\\
&\lambda_r \geq 0, \ \forall r; \mathbf{A}^{(n)} \in [0,1]^{I_n \times R}, \ \forall n\\
&||\mathbf{a}_r^{(n)}||_1 = 1, \ \forall n \qquad\qquad (6.17)
\end{aligned}
$$

The parameters $z_{\vec{i}}$ can be determined by minimizing the negative log-likelihood of the observed $x_{\vec{i}}$ and model the parameters $z_{\vec{i}}$ (see Equation 6.12). Here $\mathcal{L}(\boldsymbol{\mathcal{Z}}|\boldsymbol{\mathcal{X}})$ stands for a Bregman divergence. Bregman divergences encompass a broad range of useful loss functions including least squares, KL divergence, and logistic loss. These loss functions map to different data types (continuous, count, or 0/1, respectively for the functions in the preceding sentence). Commonly used Bregman divergences and their gradients are listed in Table 2.1.

Additionally, we augment the negative log-likelihood with constraints on the objective function to encourage rank-one components with the following characteristics: sparsity, diversity, and sparsity in terms of a specified set of between-mode combinations.

6.2.2.1 Constraints

6.2.2.1.1 Stochastic Constraints

The column stochastic constraints (Equation 6.17) allow each nonzero element to be interpreted as a conditional probability given the component (e.g., phenotype and mode) A high (close to 1) element indicates a strong relationship for this element in the component. Alternatively, a low probability (close to 0) represents a weak relationship.

6.2.2.1.2 Cannot-link Constraints

The cannot-link constraints, expressed in Equation 6.13, are motivated by the probabilistic interpretation of the components. CP-CLIC identifies the elements with low probabilities in each mode in each component (i.e., probabilities less than α) and discourages them from appearing together in the component through the penalty imposed by Equation 6.13. In Equation 6.13, $\mathbf{M}^{(m,n)} \in \mathbf{1}^{I_m \times I_n}$ is a binary cannot-link matrix defined as follows:

$$\mathbf{M}_{jk}^{(m,n)} = \begin{cases} 1, & \text{if } \mathbf{a}_{jr}^{(m)} < \alpha \text{ and } \mathbf{a}_{kr}^{(n)} < \alpha \\ 0, & \text{otherwise} \end{cases}$$

The terms in Equation 6.13 are of the form, $a_{jr}^{(m)} \mathbf{M}_{jk}^{(m,n)} a_{kr}^{(n)}$, and only contribute to the objective function if the j^{th} object in mode m and the k^{th} object in mode n appear in at least one of the R components. This constraint may also encourage sparsity in the number of elements per component since it is penalizing the smaller elements of the factors. We choose α to be an exponential loss function of k, the number of non-zeros per factor, and the epoch l. If all elements have equal probability (i.e., they are equally uninformative), they will have probability $1/k$. However, we exponentially increase α to $1/k$ over the epochs in order to not be as aggressive in earlier iterations. We describe how to refine $\mathbf{M}^{(m,n)}$ in Section 6.2.2.3.

6.2.2.1.3 Sparsity and Diversity Constraints

As in Granite (Chapter 3), Equations 6.14 and 6.15 are used to encourage diversity of the components through an angular penalty on the vectors within each factor matrix and to control the size of the $\boldsymbol{\lambda}$ weights that are fit, respectively. Additionally, we use the projection of the largest k terms in each factor vector onto an ℓ_1 ball to encourage sparse solutions.

6.2.2.2 Minimizing the objective function and building the cannot-link matrix

As in Section 6.1, we minimize the objective function using Stochastic Gradient Descent (SGD) with Adam [Kingma and Ba, 2015]. After each epoch, CP-CLIC finds the low probability elements in each component and updates the cannot-link matrix, $\mathbf{M}^{(m,n)}$ (outlined in Algorithm 5 and Figure 6.4).

For the gradient of Equation 6.12, we give several examples of widely used loss functions in Table 2.1. Section 3.3.2 details for the gradients for Equations 6.14 and 6.15. For Equation 6.13, the derivatives with respect to

Algorithm 5: CP-CLIC fitting process

```
Input: randomly initialized $[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \mathbf{A}^{(2)} \cdots ; \mathbf{A}^{(N)}]\!]$
Output: fit $[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \mathbf{A}^{(2)} \cdots ; \mathbf{A}^{(N)}]\!]$
$burn\_in \in \mathbb{I}$
$\mathbf{M}^{(m,n)} = \text{zeros}(I_m, I_n)$
$S_{\text{all}} = \emptyset$
for $l$ = $1{:}L$ do
    Run epoch of SGD with Adam
    if $l$ ¿ $burn\_in$ then
        $S = \emptyset$
        for $r{=}1{:}R$ do
            # Find low probability elements in factor
            vectors in same component
            $S_m = \{\mathbf{a}_{jr}^{(m)} < \alpha, 0 \leq j \leq I_m\}$
            $S_n = \{\mathbf{a}_{kr}^{(n)} < \alpha, 0 \leq k \leq I_n\}$
            # Obtain all combinations of $S_m$ and $S_n$
            $S = S \cup \{S_m \times S_n\}$
        end
        # Send S to auxiliary tool
        $S_{\text{lift}}$ ={ pairs with ¿ 1 lift}
        $S = S - S_{\text{lift}}$
        $S_{\text{all}} = S_{\text{all}} \cup S$
        # Set elements with indices in $S$ equal to 1
        $\mathbf{M}_{jk}^{(m,n)} = \{\mathbb{1} : j, k \in S_{\text{all}}\}$
    end
    Check convergence
end
```

the factor matrices $\mathbf{A}^{(m)}$ and $\mathbf{A}^{(n)}$ are:

$$\frac{\partial \text{Tr}(\mathbf{A}^{(m)\intercal}\mathbf{M}^{(m,n)}\mathbf{A}^{(n)}))}{\partial \mathbf{A}^{(m)}} = \mathbf{M}^{(m,n)}\mathbf{A}^{(n)} \tag{6.18}$$

$$\frac{\partial \text{Tr}(\mathbf{A}^{(m)\intercal}\mathbf{M}^{(m,n)}\mathbf{A}^{(n)}))}{\partial \mathbf{A}^{(m)}} = \mathbf{M}^{(m,n)\intercal}\mathbf{A}^{(m)} \tag{6.19}$$

6.2.2.3 Incorporating insights from auxiliary information

One possible drawback of building the cannot-link matrix in an unsupervised manner is that it is possible for two elements to have a low probability in a component but actually have a relationship in the domain in question. To mitigate the chance of this occurring, CP-CLIC uses auxiliary information to accept or reject the cannot-link constraints. Figure 6.4 gives a stylized view of how CP-CLIC incorporates auxiliary information. Algorithm 5 specifies how the cannot-link penalty matrix is built through the fitting process. For a set of specified epochs, the fit progresses without the cannot-link matrix. This is similar in spirit to the burn-in iterations in Markov Chain Monte Carlo. Once the burn-in epochs have passed, after each epoch, CP-CLIC extracts the inter-mode pairs that have a probability below a threshold. Then, for each pair, if there is not *strong enough* evidence that the relationship exists according to auxiliary information, CP-CLIC puts a 1 in the cannot-link matrix for that pair. The updated cannot-link penalty matrix is then incorporated into the next epoch of the fitting process.

In practice, auxiliary information could come in many forms (e.g., scraped from web forums or generated from past sales data). It may be possible to use the auxiliary information to build a cannot-link matrix and hard-code the constraints into $\mathbf{M}^{(m,n)}$ from the beginning of the fit instead of gradually building the cannot-link matrix as the fit progresses. This approach, which we refer to as CP-CLIC-1-Shot, may be appropriate in situations where the user has confidence in the veracity of the auxiliary information (note: PIVETed-

Granite is an example of CP-CLIC-1-SHOT introduced in Section 5.2). In other applications, however, the user might not have as much confidence in the auxiliary information. Using CP-CLIC-1-Shot in these applications may introduce noise into the decomposition and degrade the quality of the fit. Thus, gradually building the constraints in CP-CLIC may be more robust to introducing noise in $\mathbf{M}^{(m,n)}$ matrix. There is also a chance that elements in two different modes both have a low probability of occurring in one component but high probability of occurring together in another component. Using auxiliary information would hopefully allow CP-CLIC to discover the high-probability relationships and keep them from being added as constraints.

6.2.3 Experiments

6.2.3.1 Simulated Data

First, we demonstrate the CP-CLIC framework is general enough to be used for different loss functions. We evaluate CP-CLIC's performance against three synthetic tensors, where elements are drawn from a Poisson, Normal, or Exponential distribution. Specifically, we simulate third-order tensors of size $80 \times 40 \times 40$ with rank of 5 ($R = 5$). For each vector in the factor matrix $\mathbf{A}^{(n)}$, we sample non-zero element indices according to a chosen sparsity pattern and then randomly sample along the simplex for the non-zero indices, rejecting vectors that are too similar to those already generated (i.e., their normalized cosine angle is greater than θ_n). We draw the model parameters z_{ijk} from a uniform distribution. Finally, each tensor element x_{ijk} is sampled

Table 6.4: Factor match scores between fitted factor vectors and known factor vectors generated using Poisson, Normal, and Exponential distributions.

		Factor Match Score (st. dev.)		
$\mathcal{L}(\mathfrak{Z}\|\mathfrak{X})$	β_1	**Mode 1**	**Mode 2**	**Mode 3**
Poisson	0	0.934 (0.16)	0.946 (0.14)	0.946 (0.14)
	0.01	0.977 (0.04)	0.958 (0.08)	0.967 (0.06)
Normal	0	0.988 (0.01)	0.994 (0.01)	0.995 (0.00)
	0.01	0.991 (0.0)	0.997 (0.00)	0.997 (0.00)
Exponential	0	0.883 (0.12)	0.894 (0.17)	0.902 (0.17)
	0.1	0.945 (0.02)	0.967 (0.04)	0.963 (0.05)

from a Poisson, Normal, or Exponential distribution with the parameter set to z_{ijk}. For each tensor type, we simulated 40 tensors and then calculated the factor match score between the fitted vectors and the known vectors, which are matched using the Hungarian algorithm [Chi and Kolda, 2012]. A value of 1 is a perfect match.

Table 6.4 shows the factor match scores for fits with and without β_1. In all cases, CP-CLIC improves the quality of the fit and makes the biggest impact in the Exponential case. Thus, for common data types, CP-CLIC can recover the original factors.

6.2.3.2 CP-CLIC in Computational Phenotyping

6.2.3.2.1 Dataset Description

We use the same tensor constructed for experiments in Section 6.1. As in all the other models in the dissertation, we use KL-divergence for $\mathcal{L}(\mathfrak{Z}|\mathfrak{X})$. Using CP-CLIC, we report results for $R = 30$.

6.2.3.2.2 Incorporating auxiliary information in practice

We use PIVET as the source of auxiliary information (described in Section 5.2). We use lift as calculated by PIVET to prune lists of possible cannot-link pairs of diagnoses and medications.

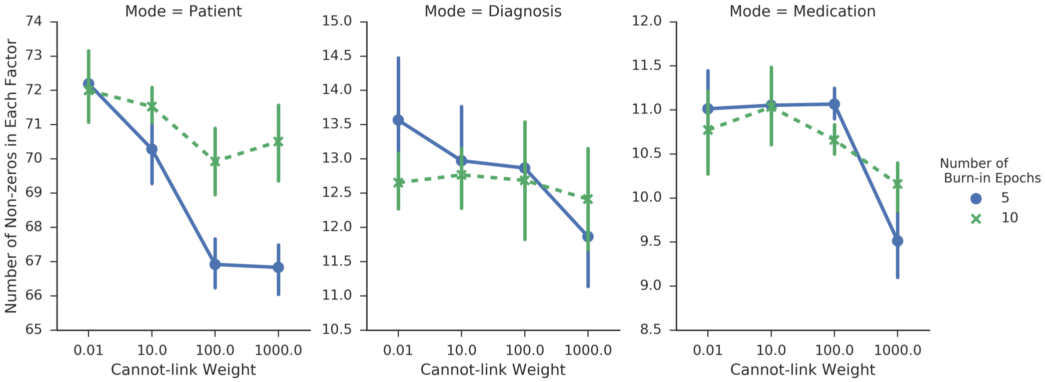


Figure 6.5: Number of non-zeros per mode for different values of β_1, the weight on the cannot-link matrix M.

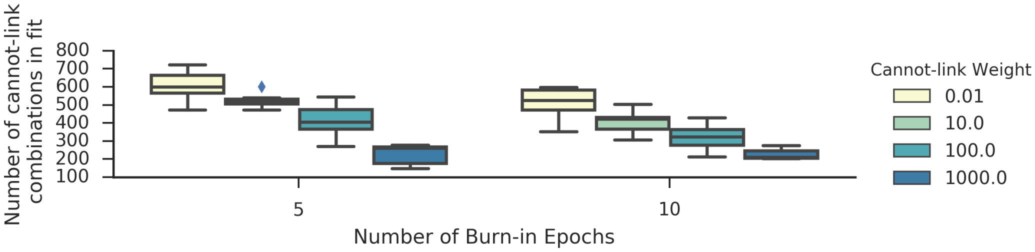


Figure 6.6: Percentage of cannot-link constraints present in after the fitting process by number of burn-in epochs and β_1, the weight on the cannot-link matrix M.

6.2.3.2.3 Computational Phenotyping Results

We evaluate CP-CLIC quantitatively and qualitatively in three ways. First, we compare features of decompositions of three variations of CP-CLIC (i.e., CP-CLIC, CP-CLIC-1-Shot, and CP-CLIC without PIVET) with those from two baselines: Granite (fit using SGD) and CP-APR. CP-APR fits a non-negative tensor factorization using KL-divergence as the objective function without sparsity constraints. Second, we perform a parameter analysis on CP-CLIC in terms of burn-in epochs and the effect of β_1, the weight associated with the cannot-link matrix. Third, based on the annotations of a domain expert, we evaluate CP-CLIC's ability to derive components that map to useful concepts.

We compare variations of CP-CLIC's performance to CP-APR and GraniteSGD in terms of computation time, the negative log-likelihood values (i.e., $\mathcal{L}(\boldsymbol{\mathcal{Z}}|\boldsymbol{\mathcal{X}})$), sparsity, and diversity. Table 6.5 shows the time in seconds each method took to complete the decomposition process. Granite implemented in the way originally presented in Henderson et al. [2017c] ran out of memory and failed to complete the decomposition process. GraniteSGD without a diversity penalty was the fastest followed by CP-CLIC without a diversity penalty.

Table 6.8 shows the negative log-likelihood and the average number of non-zeros per mode for each decomposition method. It can be seen that CP-APR has the lowest negative log-likelihood. This is not surprising given that CP-APR is an unconstrained method whereas the other methods have

Table 6.5: Time to complete decomposition by method. Standard deviation is listed in parentheses. A ✓ means $\beta_2 > 0, 0 \leq \theta_n \leq 1$.

Method	**Diversity Penalty**	**Time in seconds (st. dev.)**
CP-APR		4399.41 (850.34)
Granite		Out of Memory
GraniteSGD		**1197.66 (62.72)**
GraniteSGD	✓	9656.35 (37.92)
CP-CLIC-1-Shot		5698.64 (8.73)
CP-CLIC-1-Shot	✓	11498.15 (23.22)
CP-CLIC no PIVET		2371.94 (48.33)
CP-CLIC no PIVET	✓	10832.77 (13.92)
CP-CLIC		3886.46 (17.83)
CP-CLIC	✓	11011.08 (26.29)

sparsity constraints for interpretability purposes Kolda and Bader [2009]. Of the constrained methods, CP-CLIC without a diversity penalty has the lowest negative log-likelihood with CP-CLIC with a diversity penalty after that. This may indicate that the cannot-link constraints in CP-CLIC are improving the fit while still resulting in sparse and diverse components.

In terms of sparsity (Table 6.8), the unconstrained method CP-APR has the most number of non-zero elements in modes 2 and 3, which correspond to the diagnosis and medication modes respectively. CP-APR results in phenotypes that are not succinct. The constrained methods find larger groups of patients described by a more succinct set of attributes. Overall, in the diagnosis and medication modes, CP-CLIC-1-Shot model has sparsest factors,

Table 6.6: Mean cosine similarity of the factor vectors in each mode. A ✓ means $\beta_2 > 0, 0 \leq \theta_n \leq 1$ to encourage diversity.

Method	**Diversity Penalty**	**Mode 1**	**Mode 2**	**Mode 3**
CP-APR		0.0 (0.00)	0.22 (0.00)	0.18 (0.00)
GraniteSGD	✓	0.18 (0.01)	0.02 (0.01)	0.12 (0.01)
CP-CLIC-1-Shot	✓	0.17 (0.02)	0.01 (0.02)	0.09 (0.02)
CP-CLIC, no PIVET	✓	0.24 (0.02)	0.03 (0.02)	0.14 (0.02)
CP-CLIC	✓	0.28 (0.02)	0.03 (0.02)	0.14 (0.02)

followed by CP-CLIC. Figure 6.5 shows the average number of non-zeros per factor vector in each mode as a function of β_1, the weight on the cannot-link matrix. As β_1 increases, the sparsity also increases, which could be because the cannot-link matrix penalizes the smaller elements in the factors.

Additionally, CP-CLIC finds diverse factors with respect to the tensor. Table 6.6 shows the average cosine similarity between the factor vectors in each mode. For this particular tensor, the diversity penalty was strict for the diagnosis mode because there were many diagnoses ($\theta_2 = .45$) and laxer for the medication mode because there were relatively few medications ($\theta_3 = .75$). All CP-CLIC variations produce diagnosis and medication modes that are comparably diverse to those derived through GraniteSGD. Table 6.6 show the diagnosis mode is quite diverse and there is more overlap in the medication mode. We do not put a diversity penalty on the patient mode in this application, which is motivated by the idea that patients should be allowed to belong to any phenotype that fits their observations. Interestingly, CP-APR had very little overlap in the patient mode, which might be because there were

Table 6.7: Fit summary by decomposition method

Method	**Diversity Penalty** ($\beta_2 > 0, 0 \leq \theta_n \leq 1$)	**Negative Log-Likelihood**
CP-APR		227537.66 (14465.72)
GraniteSGD		2206587.70 (22156.60)
GraniteSGD	✓	2311866.35 (27826.92)
CP-CLIC-1-Shot		2366706.67 (8155.52)
CP-CLIC-1-Shot	✓	2309622.12 (27354.06)
CP-CLIC, no PIVET		2219006.91 (30497.52)
CP-CLIC, no PIVET	✓	2266370.60 (14724.98)
CP-CLIC		2182904.73 (29091.72)
CP-CLIC	✓	2207447.99 (28879.85)

Table 6.8: Average number of non-zeros per mode by decomposition method

	Diversity Penalty	**Number of Non-zeros Per Mode**		
Method	($\beta_2 > 0, 0 \leq \theta_n \leq 1$)	**Mode 1**	**Mode 2**	**Mode 3**
CP-APR		22.62 (1.00)	42.21 (1.14)	21.86 (0.23)
GraniteSGD		73.26 (1.03)	13.07 (0.65)	10.85 (0.54)
GraniteSGD	✓	70.45 (1.69)	18.21 (1.06)	12.31 (0.31)
CP-CLIC-1-Shot		57.87 (3.67)	10.14 (0.28)	8.91 (0.12)
CP-CLIC-1-Shot	✓	64.75 (5.73)	9.55 (0.21)	8.61 (0.21)
CP-CLIC, no PIVET		71.63 (1.68)	13.53 (0.70)	11.28 (0.39)
CP-CLIC, no PIVET	✓	72.29 (1.85)	16.65 (1.04)	11.79 (0.31)
CP-CLIC		66.83 (0.91)	11.87 (0.93)	9.51 (0.55)
CP-CLIC	✓	69.09 (0.99)	16.73 (0.49)	10.37 (0.37)

so few patients in each component. CP-CLIC's patient vectors have more in common with each other (i.e., higher cosine similarity scores) indicating that similar groups of patients belong to the same phenotypes overall.

Next, we evaluated the effect of CP-CLIC's parameters on the presence of the cannot-link constraints within the final fit. Figure 6.6 shows the number of the cannot-link (diagnosis, medication) combinations by burn-in epochs that were present in the components after the fit had finished. As the

Table 6.9: Mean (standard deviation) of cannot-link constraint statistics.

	CP-CLIC ($\beta_2 = 0$)	**CP-CLIC** ($\beta_2 > 0$)
Avg. # cannot-link pairs removed by PIVET	518.36 (81.0)	468.98 (147.0)
$\mathbf{M}^{(m,n)}$ density (%)	0.1054 (0.0027)	0.1021 (0.0044)

cannot-link weight, β_1, increases, the number of pairs in the final fit decreases. Additionally, Figure 6.6 suggests CP-CLIC is not sensitive to the number of burn-in epochs that occur before the constraint matrix building process begins, but a smaller burn-in may result in sparser factors (see Figure 6.5). We were also interested in the number of constraints PIVET removed at each iteration. Table 6.9 shows the average number of (diagnoses, medication) pairs that PIVET removed from the cannot-link matrix after each epoch. Interestingly, CP-CLIC with a diversity penalty had fewer constraints to prune at the end of each epoch on average but exhibited a larger standard deviation.

Finally, we evaluated how well the computational phenotypes mapped to clinical concepts. A domain expert analyzed CP-CLIC-1-Shot, CP-CLIC without PIVET, CP-CLIC, and GraniteSGD phenotypes and annotated each one with one of the following labels: 1) yes, clinically relevant, 2) maybe, possibly clinically relevant, and 3) no, not clinically relevant. For a phenotyping algorithm to be considered successful it should have components that map mostly to clinically relevant or possibly clinically relevant labels. We compiled

5 phenotypes from each decomposition with the largest component weights λ and then randomized the order as to not bias the annotator. A subset of phenotypes was given to reduce annotation fatigue and ensure annotation quality. Figure 6.7 shows the results of the annotation process. GraniteSGD and CP-CLIC without PIVET performed the worst in that they resulted in the most phenotypes that were labeled not clinically meaningful. The larger numbers of not clinically meaningful phenotypes indicate these models may be less suitable for phenotype derivation. Overall, CP-CLIC-1-Shot model performed the best, resulting in a collection of clinically significant and possibly significant phenotypes. In this application, the quality of the auxiliary information (i.e., PIVET) is high, which may mean CP-CLIC-1-Shot is the most appropriate approach. However, in other domains the quality of the information may not be as high, and the best strategy may be gradually learning the constraints (i.e., CP-CLIC). After CP-CLIC-1-Shot, CP-CLIC extracted the most clinically meaningful or possibly meaningful phenotypes. The possibly clinically meaningful phenotypes could provide a new avenue for clinical studies and aid in knowledge discovery. Since only a percentage of the phenotypes were annotated, the distribution of clinical meaningfulness might change. However, these results suggest using inter-mode constraints can help improve the clinical relevance of the derived phenotypes.

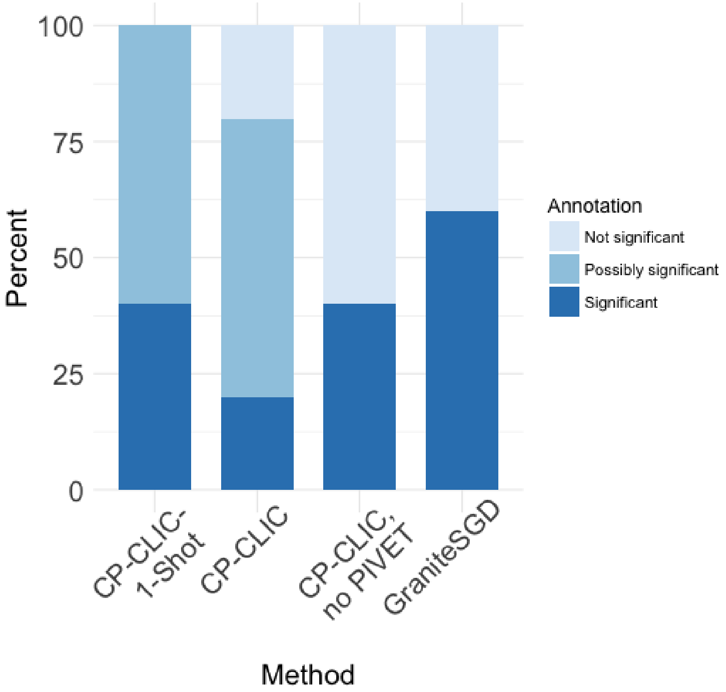


Figure 6.7: Clinical significance of phenotypes by method.

6.2.4 Conclusion

Adding guidance in the form of constraints to tensor decompositions can help improve the quality of the derived components in terms of interpretability, sparsity, and diversity. However, obtaining informative constraints can be expensive in regard to time and effort required by domain experts. This section shows that features of the CP decomposition process can be utilized to discover constraints through the learning method. This framework, CP-CLIC, gradually uncovers between-mode cannot-link constraints and then validates the constraints using domain expertise in the form of auxiliary information. CP-CLIC is a flexible, novel framework in that it 1) works on all or a subset of modes of the tensor, 2) is well-suited for many different types of data, and 3) scales to large tensors. In situations where the quality of the auxiliary information is high, it may be appropriate to forgo the gradual discovery of cannot-link constraints and supply the dense cannot-link matrix at the beginning of the learning process (CP-CLIC-1-Shot). We show that in both the simulated and computational phenotyping experiments, gradually discovering the constraints can improve the quality of the fit. Moreover, in a real-world case study, CP-CLIC yields 66% more interpretable components than the baseline.

Chapter 7

Conclusion

The widespread adoption of EHR systems to track patients' interactions with health care systems offers the promise of improving patient care through computational techniques. The ability to efficiently characterize large volumes of healthcare data is essential to enabling clinicians to use this information effectively to better understand the populations that they serve. In this dissertation, we investigated adapting tensor factorization methods to produce phenotypes that fit the specifications of sparsity and diversity. With **Granite**, we discovered succinct and different phenotypes with minimal human supervision. We also showed how to incorporate side information about patient disease status into the tensor factorization process to discover phenotypes that could be descriptive of those diseases in supervised (**gamAID**) and semisupervised (**PSST**) frameworks.

Additionally, we investigated how to extract the domain expertise contained in a corpus of medical articles to build evidence for the clinical relevance of the discovered phenotypes. PheKnow–Cloud, a prototype tool, showed promise in the clinical validation of phenotypes, but only functioned in a batch setting and used a brute-force analysis method. We improved on PheKnow–

Cloud with **PIVET**, a fast and flexible validation tool that works in both batch and individual settings.

Finally, we formulated a framework (**PIVETed-Granite**) that incorporates domain expertise provided by a phenotype validation tool to guide the factorization process to more focused and discriminative phenotypes. Furthermore, we showed how to use features of the tensor factorization process in conjunction with auxiliary information to guide the tensor factorization process to sparse, diverse, and discriminative phenotypes (**CP-CLIC**). The novel algorithms, together with the validation framework, facilitate the discovery of phenotypes that are interpretable and have the promise of adding trusted and validated insights to precision medicine efforts.

7.1 Future Work

There are many veins of research that would be worth pursuing in the future. One way to extend the tensor factorization algorithms would be to include time, which could be done by adding time as a mode in the tensor. Since patients interact with healthcare practitioners at different rates, a key challenge to including a time mode is aligning the patients in a meaningful way. If this challenge is addressed, including time as a mode would allow clinicians to more clearly see disease progression and could help identify patients for interventions proactively. Additionally, although domain expert input can be difficult to obtain, it is also possible we could make CP-CLIC interactive with the help of volunteer clinicians. After every epoch, CP-CLIC could pause and obtain domain expert input. Domain expert users could then specify constraints as they see fit and direct the phenotype discovery process to phenotypes of interest. It would be interesting to compare an interactive approach with the automated approach.

On the phenotype validation side, it would be beneficial to increase the size of the training set on which PIVET trains, which would involve collecting more domain-expert annotated phenotypes. Furthermore, it may be valuable to shift away from co-occurrence analysis to an analysis based on representation learning. Applying representation learning to phenotypes and articles may uncover semantic similarities that are not present when purely using text matching techniques.

Bibliography

Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of J. Chemom.*, 25(2):67–86, 2011a.

Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mrup. Scalable tensor factorizations for incomplete data. *Chemometr. Intell. Lab. Syst.*, 106(1):41 – 56, 2011b. ISSN 0169-7439.

Evrim Acar, Mathias Nilsson, and Michael Saunders. A flexible modeling framework for coupled matrix and tensor factorizations. *Proc. Eur. Signal Process. Conf. EUSIPCO*, pages 111–115, 2014.

Ardavan Afshar, Joyce C. Ho, Bistra Dilkina, Ioakeim Perros, Elias B. Khalil, Li Xiong, and Vaidy Sunderam. Cp-ortho: An orthogonal tensor factorization framework for spatio-temporal data. In *Proc. ACM SIGSPATIAL Int. Conf. Adv. Inf.*, SIGSPATIAL'17, pages 67:1–67:4, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5490-5.

Alfred V Aho and Margaret J Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

Noha Alnazzawi, Paul Thompson, Riza Batista-Navarro, and Sophia Ananiadou. Using text mining techniques to extract phenotypic information from the phenochf corpus. In *BMC medical informatics and decision making*, volume 15, page S3. BioMed Central, 2015.

Sophia Ananiadou, Douglas B. Kell, and Jun ichi Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12): 571 – 579, 2006. ISSN 0167-7799. doi: https://doi.org/10.1016/j.tibtech.2006.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0167779906002423`.

S Becker, V Cevher, C Koch, and A Kyrillidis. *Sparse projections onto the simplex*. Proc. of Int. Conf. Mach. Learn., 2013.

Alexandru Boicea, Florin Radulescu, and Laura Ioana Agapin. Mongodb vs oracle–database comparison. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on*, pages 330–335. IEEE, 2012.

Mary Regina Boland, Zachary Shahn, David Madigan, George Hripcsak, and Nicholas P Tatonetti. Birth month affects lifetime disease risk: a phenome-wide method. *Journal of the American Medical Informatics Association*, page ocv046, 2015.

Giuseppe Boriani, Irina Savelieva, Gheorghe-Andrei Dan, Jean Claude Deharo, Charles Ferro, Carsten W Israel, Deirdre A Lane, Gaetano La Manna, Joseph Morton, Angel Moya Mitjans, et al. Chronic kidney disease in patients with cardiac rhythm disturbances or implantable electrical devices:

clinical significance and implications for decision making-a position paper of the european heart rhythm association endorsed by the heart rhythm society and the asia pacific heart rhythm society. *Europace*, 17(8):1169–1196, 2015.

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. and Math. Phys.*, 7(3):200–217, 1967.

Ryan Bridges, Jette Henderson, Joyce C. Ho, Byron C. Wallace, and Joydeep Ghosh. Automated verification of phenotypes using pubmed. In *ACM BCB Workshop on Methods and Applications in Healthcare Analytics*. Accepted, ACM, 2016.

Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, SIGMOD '97, pages 255–264, New York, NY, USA, 1997. ACM. ISBN 0-89791-911-4. doi: 10.1145/253260.253325. URL `http://doi.acm.org/10.1145/253260.253325`.

Thomas J Cahill and Rajesh K Kharbanda. Heart failure after myocardial infarction in the era of primary percutaneous coronary intervention: Mechanisms, incidence and identification of patients at risk. *World journal of cardiology*, 9(5):407, 2017.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

J Douglas Carroll, Sandra Pruzansky, and Joseph B Kruskal. Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45(1):3–24, Mar 1980. ISSN 1860-0980.

Robert J Carroll, Anne E Eyler, and Joshua C Denny. Naive electronic health record phenotype identification for rheumatoid arthritis. In *Proc. of Am. Med. Inf. Assoc. Annu. Symp.*, volume 2011, pages 189–96, 2011.

Kerri L. Cavanaugh. Diabetes management issues for patients with chronic kidney disease. *Clinical Diabetes*, 25(3):90–97, 2007. ISSN 0891-8929. doi: 10.2337/diaclin.25.3.90. URL `http://clinical.diabetesjournals.org/content/25/3/90`.

Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 507–516. ACM, 2015.

You Chen, Joydeep Ghosh, Cosmin Adrian Bejan, Carl A Gunter, Siddharth Gupta, Abel Kho, David Liebovitz, Jimeng Sun, Joshua Denny, and Bradley Malin. Building bridges across electronic health record systems through inferred phenotypic topics. *J. Biomed. Inf.*, 55:82–93, 2015.

Yukun Chen, Robert J Carroll, Eugenia R McPeek Hinz, Anushi Shah, Anne E Eyler, Joshua C Denny, and Hua Xu. Applying active learning to high-throughput phenotyping algorithms for electronic health records data. *J. Am. Med. Inform. Assoc.*, 20(e2):e253–e259, 2013.

Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM J. Matrix Anal. Appl.*, 33(4):1272–1299, 2012.

Peter J A Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.

Nigel Collier, Tudor Groza, Damian Smedley, Peter N Robinson, Anika Oellrich, and Dietrich Rebholz-Schuhmann. PhenoMiner: from text to a database of phenotypes associated with OMIM diseases. *Database*, 2015, October 2015.

Ian Davidson, Sean Gilpin, Owen Carmichael, and Peter Walker. Network discovery via constrained tensor analysis of fmri data. In *Proc. of 19th ACM SIGKDD Int. Conf. on Knowl. Discov. Data Min.*, KDD '13, pages 194–202, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7.

AL De Francisco. Gastrointestinal disease and the kidney. *European journal of gastroenterology & hepatology*, 14:S11–5, 2002.

Joshua C Denny, Lisa Bastarache, Marylyn D Ritchie, and Robert J et al Carroll. Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nat. Biotechnol.*, 31(12):1102–1111, November 2013.

K Dickersin. The existence of publication bias and risk factors for its occurrence. *JAMA*, 263(10):1385–1389, March 1990.

John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. *Proc. of Int. Conf. Mach. Learn.*, pages 272–279, 2008.

Kerry Dwan, Douglas G Altman, Juan A Arnaiz, Jill Bloom, An-Wen Chan, Eugenia Cronin, Evelyne Decullier, Philippa J Easterbrook, Erik Von Elm, Carrol Gamble, Davina Ghersi, John P A Ioannidis, John Simes, and Paula R Williamson. Systematic review of the empirical evidence of study publication bias and outcome reporting bias. *PLoS One*, 3(8):e3081, August 2008.

Phillipa J Easterbrook, Ramana Gopalan, JA Berlin, and David R Matthews. Publication bias in clinical research. *The Lancet*, 337(8746):867–872, 1991.

Perihan Elif Ekmekci. An increasing problem in publication ethics: Publication bias and editors' role in avoiding it. *Med. Health Care Philos.*, 20(2):171–178, June 2017.

Matthias Frisch, Bernward Klocke, Manuela Haltmeier, and Kornelie Frech. Litinspector: literature and signal transduction pathway mining in pubmed abstracts. *Nucleic acids research*, 37(suppl 2):W135–W140, 2009.

Samantha Hansen, Todd Plantenga, and Tamara G Kolda. Newton-based optimization for kullback-leibler nonnegative tensor factorizations. *Optim. Methods Softw.*, 30(5):1002–1029, 2015.

Richard A Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

Ron D Hays, Joel D Kallich, Donna L Mapes, Stephen J Coons, and William B Carter. Development of the kidney disease quality of life (kdqol?) instrument. *Quality of Life Research*, pages 329–338, 1994.

Ricardo Henao, James T Lu, Joseph E Lucas, Jeffrey Ferranti, and Lawrence Carin. Electronic health record analysis via deep poisson factor models. *J. Mach. Learn. Res.*, 17(186):1–32, 2015.

Jette Henderson, Ryan Bridges, Joyce C. Ho, Byron C. Wallace, and Joydeep Ghosh. PheKnow-Cloud: A tool for evaluating high-throughput phenotype candidates using online medical literature. In *2017 Joint Summits on Translational Bioinformatics*, volume 2017, pages 149–157. American Medical Informatics Association, 2017a.

Jette Henderson, Joyce C. Ho, and Joydeep Ghosh. gamAID: Greedy CP tensor decomposition for supervised EHR-based disease trajectory differentiation. In *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2017. IEEE, 2017b.

Jette Henderson, Joyce C. Ho, Abel N Kho, Joshua C Denny, Bradley A Malin, Jimeng Sun, and Joydeep Ghosh. Granite: Diversified, sparse tensor factorization for electronic health record-based phenotyping. In *Fifth IEEE International Conference on Healthcare Informatics*, volume 2017, pages 214–223. IEEE, 2017c.

Jette Henderson, Huan He, Bradley A. Malin, Joshua C. Denny, Abel N. Kho, Joydeep Ghosh, and Joyce C. Ho. Phenotyping through semi-supervised tensor factorization (PSST). To appear in Am Med Inf Assoc Ann Symp, 2018a.

Jette Henderson, Junyuan Ke, C. Joyce Ho, Joydeep Ghosh, and C. Byron Wallace. Phenotype instance verification and evaluation tool (PIVET): A scaled phenotype evidence generation framework using web-based medical literature. *J. Med. Internet Res.*, 20(5):e164, May 2018b.

Jette Henderson, Bradley A. Malin, Joyce C. Ho, and Joydeep Ghosh. PIVETed-Granite: Computational phenotypes through constrained tensor factorization. To appear at 2018 KDD Workshop on Machine Learning for Medicine and Healthcare, 2018c.

Jette Henderson, Bradley A. Malin, Abel N. Kho, Joydeep Ghosh, and Joyce C. Ho. CP tensor decomposition with cannot-link intermode constraints (CP-CLIC). submitted, 2018d.

Joyce C Ho, Joydeep Ghosh, Steve R Steinhubl, Walter F Stewart, Joshua C Denny, Bradley A Malin, and Jimeng Sun. Limestone: High-throughput candidate phenotype generation via tensor factorization. *J. Biomed. Inf.*, 52:199–211, 2014a.

Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proc. of ACM Knowledge Discover and Data Mining*, pages 115–124, 2014b.

David Hong, Tamara G. Kolda, and Cliff Anderson-Bergman. Stochastic gradient for generalized cp tensor decomposition (preliminary results). Presentation by Tamara G. Kolda at Autumn School: Optimization in Machine Learning and Data Science, Trier University, Trier, Germany, Aug 2017.

Sally Hopewell, Kirsty Loudon, Mike J Clarke, Andrew D Oxman, and Kay Dickersin. Publication bias in clinical trials due to statistical significance or direction of trial results. *Cochrane Database Syst. Rev.*, 1:MR000006, January 2009.

George Hripcsak and David J Albers. Next-generation phenotyping of electronic health records. *J. Am. Med. Inform. Assoc.*, 20(1):117–121, 2013.

Changwei Hu, Piyush Rai, Changyou Chen, Matthew Harding, and Lawrence Carin. Scalable bayesian non-negative tensor factorization for massive count data. In *Mach. Learn. Knowl. Discov. Databases*, pages 53–70. Springer, 2015.

Maria Indrawan-Santiago. Database research: Are we at a crossroad? reflection on nosql. In *Network-Based Information Systems (NBiS), 2012 15th International Conference on*, pages 45–51. IEEE, 2012.

Lars Juhl Jensen, Jasmin Saric, and Peer Bork. Literature mining for the biologist: from information retrieval to biological discovery. *Nature reviews genetics*, 7(2):119–129, 2006.

Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews: Genetics*, 13(6):395–405, 2012.

Shalmali Joshi, Suriya Gunasekar, David Sontag, and Joydeep Ghosh. Identifiable phenotyping using constrained non-negative matrix factorization. In *1st Conf. on Machine Learning and Health Care*, 2016.

David C Kale, Zhengping Che, Mohammad Taha Bahadori, Wenzhe Li, Yan Liu, and Randall Wetzel. Causal phenotype discovery via deep networks. In *Proc. of Am. Med. Inf. Assoc. Annu. Symp.*, pages 677–686, 2015.

Jeongkyun Kim, Jung-jae Kim, and Hyunju Lee. An analysis of disease-gene relationship from medline abstracts by digsee. *Scientific reports*, 7:40154, 2017a.

Yejin Kim, Robert El-Kareh, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Discriminative and distinct phenotyping by constrained tensor factorization. *Scientific reports*, 7(1):1114, 2017b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd Int. Conf. for Learn. Rep.*, 2015.

Jacqueline C Kirby, Peter Speltz, Luke V Rasmussen, Melissa Basford, Omri Gottesman, Peggy L Peissig, Jennifer A Pacheco, Gerard Tromp, Jyotishman Pathak, David S Carrell, Stephen B Ellis, Todd Lingren, Will K Thompson, Guergana Savova, Jonathan Haines, Dan M Roden, Paul A Harris, and Joshua C Denny. PheKB: a catalog and workflow for creating electronic phenotype algorithms for transportability. *J. Am. Med. Inform. Assoc.*, 23(6):1046–1052, November 2016.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Rodrigo M Lago and Richard W Nesto. Type 2 diabetes and coronary heart disease: focus on myocardial infarction. *Current diabetes reports*, 9(1):73–78, 2009.

Jean-Baptiste Lamy, Alain Venot, and Catherine Duclos. PyMedTermino: an open-source generic API for advanced terminology services. *Stud Health Technol Inform.*, 210:924–928, 2015.

Hany Lashen. Role of metformin in the management of polycystic ovary syndrome. *Therapeutic advances in endocrinology and metabolism*, 1(3):117–128, 2010.

Jay Wook Lee. Fluid and electrolyte disturbances in critically ill patients. *Electrolytes & Blood Pressure*, 8(2):72–81, 2010.

Yishan Li and Sathiamoorthy Manoharan. A performance comparison of sql and nosql databases. In *Communications, computers and signal processing (PACRIM), 2013 IEEE pacific rim conference on*, pages 15–19. IEEE, 2013.

C E Lipscomb. Medical subject headings (MeSH). *Bull. Med. Libr. Assoc.*, 88 (3):265–266, July 2000.

Amanda N Long and Samuel Dagogo-Jack. Comorbidities of diabetes and hypertension: mechanisms and approach to target organ protection. *J. Clin. Hypertens. (Greenwich)*, 13(4):244–251, 2011.

Zhiyong Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011:baq036, January 2011.

Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Min. and Know. Disc.*, 25 (2):298–324, Sep 2012. ISSN 1573-756X.

NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, 46(D1):D8–D13, January 2018.

W. Peng. Constrained nonnegative tensor factorization for clustering. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 954–957, Dec 2010. doi: 10.1109/ICMLA.2010.152.

Ioakeim Perros, Robert Chen, Richard Vuduc, and Jimeng Sun. Sparse hierarchical tucker factorization and its application to healthcare. In *IEEE Int. Conf. on Data Mining*, pages 943–948, 2015.

Rimma Pivovarov, Adler J Perotte, Edouard Grave, John Angiolillo, Chris H Wiggins, and Noémie Elhadad. Learning probabilistic phenotypes from heterogeneous ehr data. *J. Biomed. Inf.*, 58:156–165, 2015.

Conrad Plake, Torsten Schiemann, Marcus Pankalla, Jörg Hakenberg, and Ulf Leser. AliBaba: PubMed as a graph. *Bioinformatics*, 22(19):2444–2445, October 2006.

Sune Pletscher-Frankild, Albert Pallejà, Kalliopi Tsafou, Janos X Binder, and Lars Juhl Jensen. Diseases: Text mining and data integration of disease–gene associations. *Methods*, 74:83–89, 2015.

Deepak K Rajpal, Xiaoyan A Qu, Johannes M Freudenberg, and Vinod D Kumar. Mining emerging biomedical literature for understanding disease associations in drug discovery. *Methods Mol. Biol.*, 1159:171–206, 2014.

Jyoti Rani, A B Rauf Shah, and Srinivasan Ramachandran. pubmed.miner: an R package with text-mining algorithms to analyse PubMed abstracts. *J. Biosci.*, 40(4):671–682, October 2015.

PJ Richardson and Lawford S Hill. Relationship between hypertension and angina pectoris. *Br. J. Clin. Pharmacol.*, 7(S2):249S–253S, 1979.

Rachel L Richesson, Jimeng Sun, Jyotishman Pathak, Abel N Kho, and Joshua C Denny. Clinical phenotyping in selected national networks: demonstrating the need for high-throughput, portable, and computational methods. *AI in Medicine*, 71:57–61, 2016.

Marylyn D Ritchie, Joshua C Denny, Dana C Crawford, Andrea H Ramirez, Justin B Weiner, Jill M Pulley, Melissa A Basford, Kristin Brown-Gentry, Jeffrey R Balser, Daniel R Masys, et al. Robust replication of genotype-phenotype associations across multiple diseases in an electronic medical record. *Am. J. Hum. Genet.*, 86(4):560–572, 2010.

D. M. Roden, J. M. Pulley, M. A. Basford, G. R. Bernard, E. W. Clayton, J. R. Balser, and D. R. Masys. Development of a large-scale de-identified DNA biobank to enable personalized medicine. *Clin. Pharmacol. Ther.*, 84 (3):362–369, 2008.

Claude Sammut and Geoffrey I. Webb, editors. *Semi-Supervised Learning*, pages 892–897. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_749. URL https://doi.org/10.1007/978-0-387-30164-8_749.

L M Sheikh, B Tanveer, and M A Hamdani. Interesting measures for mining association rules. In *8th International Multitopic Conference, 2004. Proceedings of INMIC 2004.*, pages 641–644. ieeexplore.ieee.org, December 2004.

F Song, S Parekh, L Hooper, Y K Loke, J Ryder, A J Sutton, C Hing, C S Kwok, C Pang, and I Harvey. Dissemination and publication of research findings: an updated review of related biases. *Health Technol. Assess.*, 14 (8):iii, ix–xi, 1–193, February 2010.

Fujian Song, Lee Hooper, and Yoon Loke. Publication bias: what is it? how do we measure it? how do we avoid it? *Open Access Journal of Clinical Trials*, 2013(5):71–81, 2013.

Krisztian Stadler, Ira J Goldberg, and Katalin Susztak. The evolving understanding of the contribution of lipid metabolism to diabetic kidney disease. *Current diabetes reports*, 15(7):1–8, 2015.

Alwin Stegeman and Pierre Comon. Subtracting a best rank-1 approximation may increase tensor rank. *Linear Algebra Appl.*, 433(7):1276–1300, 2010.

Jerome M Stern and R John Simes. Publication bias: evidence of delayed publication in a cohort study of clinical research projects. *Bmj*, 315(7109): 640–645, 1997.

William N Venables and Brian D Ripley. *Modern applied statistics with S-PLUS.* Springer Science & Business Media, 2013.

Sebastián Ventura, José María Luna, et al. *Pattern mining with evolutionary algorithms*. Springer, 2016.

Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C Denny, Abel Kho, You Chen, Bradley A Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 1265–1274, 2015.

Henry Wasserman and Jerome Wang. An applied evaluation of snomed ct as a clinical vocabulary for the computerized diagnosis and problem list. In *AMIA Annual Symposium Proceedings*, volume 2003, page 699. American Medical Informatics Association, 2003.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Dong Xu, Meizhuo Zhang, Yanping Xie, Fan Wang, Ming Chen, Kenny Q Zhu, and Jia Wei. DTMiner: identification of potential disease targets through biomedical literature mining. *Bioinformatics*, 32(23):3619–3626, December 2016.

Sheng Yu, Katherine P Liao, Stanley Y Shaw, Vivian S Gainer, Susanne E Churchill, Peter Szolovits, Shawn N Murphy, Isaac S Kohane, and Tianxi Cai. Toward high-throughput phenotyping: unbiased automated feature extraction and selection from knowledge sources. *Journal of the American Medical Informatics Association*, 22(5):993–1000, April 2015.

Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. From micro to macro: data driven phenotyping by densification of longitudinal electronic medical records. In *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2014.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.*, 67(2):301–320, 2005.