Marble: High-throughput Phenotyping from Electronic Health Records via Sparse Nonnegative Tensor Factorization

Joyce Ho
University of Texas at Austin
joyceho@utexas.edu

Joydeep Ghosh
University of Texas at Austin
ghosh@ece.utexas.edu

Jimeng Sun
Georgia Institute of Technology
jsun@cc.gatech.edu

ABSTRACT

The rapidly increasing availability of electronic health records (EHRs) from multiple heterogeneous sources has spearheaded the adoption of data-driven approaches for improved clinical research, decision making, prognosis, and patient management. Unfortunately, EHR data do not always directly and reliably map to phenotypes, or medical concepts, that clinical researchers need or use. Existing phenotyping approaches typically require labor intensive supervision from medical experts.

We propose Marble, a novel sparse non-negative tensor factorization method to derive phenotype candidates with virtually no human supervision. Marble decomposes the observed tensor into two terms, a bias tensor and an interaction tensor. The bias tensor represents the baseline characteristics common amongst the overall population and the interaction tensor defines the phenotypes. We demonstrate the capability of our proposed model on both simulated and patient data from a publicly available clinical database. Our results show that Marble derived phenotypes provide at least a 42.8% reduction in the number of non-zero element and also retains predictive power for classification purposes. Furthermore, the resulting phenotypes and baseline characteristics from real EHR data are consistent with known characteristics of the patient population. Thus it can potentially be used to rapidly characterize, predict, and manage a large number of diseases, thereby promising a novel, data-driven solution that can benefit very large segments of the population.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Tensor; Dimensionality reduction; EHR phenotyping; Application

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623658 .

1. INTRODUCTION

Electronic health records (EHRs) are becoming an increasingly important source of detailed patient information. Effective integration and efficient analysis of EHRs can aid in solving many of the healthcare problems: making informed clinical decisions, improving patient safety, and facilitating investigations and knowledge discovery. However, several formidable challenges arise from the application of EHR data to clinical research, including diverse populations, heterogeneous and noisy information, and interpretability constraints. Medical professionals are accustomed to reasoning based on concise and meaningful medical concepts, or phenotypes. EHR-based phenotyping is a process to map raw EHR data into meaningful medical concepts, learning medically relevant characteristics of the data [15], and is important for supporting genome-wide association studies [10]. An example is the severe early childhood obesity phenotype[1], which identifies children with increased risk of adult obesity and a potential lifetime of complications.

State of the art phenotype development, such as the eMerge Network[2], relies primarily on approaches that are heuristic, rule, and iterative based, and is a collaborative team effort between clinicians and IT experts [15, 22]. Recent work has focused on high-throughput phenotyping, efficient and automated phenotype extractions to reduce manual development. Although data mining tools have been utilized to automate the phenotype process, current high-throughput methodologies cannot generate large amounts of candidate phenotypes and achieve good performance without human annotated samples [5]. Therefore, a major limitation of existing phenotype efforts is the need for human annotation of case and control samples, which require substantial time, effort, and expert knowledge to develop.

The "ideal" phenotype (i) represents complex interactions between several sources, (ii) is concise and easily understood by a medical professional, and (iii) maps to domain knowledge. Thus, phenotyping can be viewed as a form of dimensionality reduction, where each phenotype forms a latent space [15]. Matrix factorization, a common dimensionality reduction approach, is insufficient as it cannot concisely capture structured EHR source interactions, such as multiple medications prescribed to treat a single disease. A more natural transformation is tensor factorization, which utilizes

[1] The phenotype definition can be found in Phenotype KnowledgeBase.

[2] The eMerge network explores the use of EHR to obtain phenotypic information at multiple medical institutions.

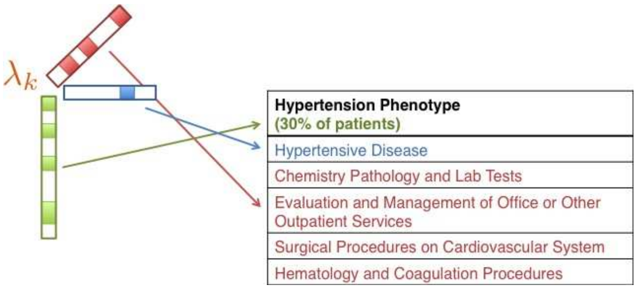

λ_k
Hypertension Phenotype
(30% of patients)
Hypertensive Disease
Chemistry Pathology and Lab Tests
Evaluation and Management of Office or Other Outpatient Services
Surgical Procedures on Cardiovascular System
Hematology and Coagulation Procedures

Figure 1: Example of a Marble-derived phenotype from CMS data.

the multiway structure to produce concise and potentially more interpretable results. We conducted a pilot study to evaluate tensor-derived phenotypes using EHR data from the Geisinger Health System [14]. A medical expert evaluated 50 tensor-derived phenotypes and found that 82% generally mapped to a medical concept. However, the domain expert's comments primarily revolved around the lack of conciseness and the presence of "unnecessary" diagnoses and medications. Therefore, a sparse non-negative tensor factorization of count data is desired to simultaneously generate multiple concise phenotypes. An efficient and automated high-throughput phenotyping process can help identify existing, as well as novel, medical concepts from large-scale EHR data, and increase our ability to create personalized applications to improve the health and well-being of the general population.

This paper presents *Marble*, a novel sparse non-negative tensor factorization model to fit count data. Analogous to the geology domain, where marble rock is used to produce monuments, buildings, and sculptures, our algorithm can serve as the basis of automated high-throughput phenotyping tools. Our model extends the non-negative CANDECOMP/PARAFAC (CP) Poisson tensor decomposition [6] from two aspects: (i) constraints on the factor matrices to minimize the number of non-zero elements, and (ii) augmentation of the tensor approximation. Marble decomposes an observed tensor into two terms, a bias (or offset) tensor and an interaction (or signal) tensor. The bias tensor represents the baseline characteristics common amongst the overall population and also provides computational stability. The interaction term is compromised of concise, intuitive, and interpretable phenotypes in the data, illustrated in Figure 1. This paper details the tensor factorization model and presents the algorithm to solve the problem formulation using both an alternating minimization and sequential unconstrained minimization approach. We corroborate our model on simulation data as well as real EHR data. Our results demonstrate that Marble achieves at least a 42.8% reduction in the number of non-zero elements compared to CP-APR without sacrificing the quality of the tensor decomposition. Furthermore, the phenotypes and the baseline characteristics derived from the real EHR data are consistent with existing studies on the population.

The remainder of the paper is structured as follows. Section 2 presents preliminaries of matrix and tensor factorization and related work. Next, we detail our model in Section 3. Section 4 demonstrates and evaluates our model on simulation data and real EHR data. Finally, we summarize our work in Section 5.

Table 1: List of notations used in this paper

Symbol	Definition
α, γ	scalar
$\boldsymbol{\lambda}, \mathbf{u}, \mathbf{a}$	vector
$\mathbf{A}, \mathbf{B}, \mathbf{Z}, \boldsymbol{\Lambda}, \boldsymbol{\Pi}, \boldsymbol{\Phi}, \boldsymbol{\Psi}$	matrix
$\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{V}}$	tensor
$\vec{i}$	tensor element index $(i_1, i_2, \cdots, i_N)$
$x_{\vec{i}}$	tensor element at index $\vec{i}$
$\mathbf{X}_{(n)}, \mathbf{M}_{(n)}, \mathbf{C}_{(n)}$	mode-n matricization of tensor
$*$	element-wise multiplication
$\oslash$	element-wise division
$\circ$	outer product
$\odot$	Khatri-Rao product
$< \mathbf{x}, \mathbf{y} >$	inner product of $\mathbf{x}, \mathbf{y}$

2. PRELIMINARIES AND RELATED WORK

This section describes the preliminaries of matrix and tensor decomposition and related tensor factorization work. Table 1 provides a key for the symbols used in the paper. For indexing of matrix $\mathbf{A}$, we denote the (i, j)th element as a_{ij}, the jth column as $\mathbf{a}_{:j}$, and the ith column as $\mathbf{a}_{i:}$.

Matrix decomposition. Matrix factorization (MF) is a common dimensionality reduction approach, which represents the original data using a lower dimensional latent space. Standard MF approaches find two lower dimensional matrices that when multiplied together approximately produce the original matrix, $\mathbf{X} \approx \mathbf{WH}$. Although many matrix decomposition techniques exist, singular value decomposition and nonnegative matrix factorization (NMF) are two common algorithms used to reduce the feature dimension.

Notation Details. Definitions for algebraic operations used in the paper are provided below.

Definition 1. *The outer product of N vectors, $\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$, produces a Nth order tensor $\boldsymbol{\mathcal{X}}$ where each element $x_{\vec{i}} = x_{i_1, i_2, \cdots, i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$.*

Definition 2. *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 3. *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. The Kronecker product of two vectors* $\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \mathbf{b} \\ a_2 \mathbf{b} \\ \vdots \\ a_{I_A} \mathbf{b} \end{bmatrix}$.

Tensor Decomposition. A tensor is a generalization of matrices to higher dimensions. Thus, tensor factorization (decomposition) is a natural extension of matrix factorization and utilizes information from the multiway structure that is lost when modes are collapsed to use matrix factorization algorithms [21, 23]. The CANDECOMP / PARAFAC

(CP) [3, 13] model is a common tensor decomposition and can be viewed as a higher-order generalization of singular value decomposition [17]. The CP model approximates the original tensor $\boldsymbol{\mathcal{X}}$ as a sum of R rank-one tensors and can be expressed as

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

Note that $[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \ldots; \mathbf{A}^{(N)}]\!]$ is shorthand notation to describe the CP decomposition, where $\boldsymbol{\lambda}$ is a vector of the weights λ_r and $\mathbf{a}_r^{(n)}$ is the rth column of $\mathbf{A}^{(n)}$. The CP tensor decomposition has been used for concept discovery [16], network analysis of fMRI data [9], and community discovery [20]. The details of computing the CP decomposition and other tensor decomposition models can be found in [17].

Some domain applications may desire non-negative components, a higher-order generalization of NMF. Non-negative tensor factorization (NTF) requires the elements of the factor matrices and the weights to be non-negative. A broad survey of practical and useful NMF and NTF algorithms can be found in [8]. Our paper will focus on the nonnegative CP alternating Poisson regression (CP-APR) model to fit sparse count data [6]. Details of the algorithm and model are presented in the paper by Chi and Kolda [6].

NTF and NMF algorithms generally produce sparse representations. However, additional sparsity may be desired for the factor matrices. Traditional sparsity-inducing penalties such as ℓ_1 and ℓ_2 regularization [23] only deal with the standard least-squares minimization. Non-parametric Bayesian approaches to sparse Tucker decomposition have been recently proposed [24]. Nonetheless, there is a paucity of existing work regarding sparse factor representations using KL divergence as an objective function. A multi-layer NTF has been proposed to achieve sparse representations for various cost functions including KL divergence using a nonlinearly transformed gradient descent approach [7]. Yet, the approach is computationally expensive because multiple tensor factorization stages are required and sparsity constraints are achieved via an exponential update (i.e. $s \leftarrow s^{1+\gamma}$, for small γ). Sparse factor representation using KL divergence is difficult because (1) an incorrect zero in the factor representation causes the objective function to be ill-defined as $\lim_{i\to 0} \log i = -\infty$, and (2) data centering, a technique commonly used to remove the bias in continuous data prior to matrix/tensor factorization, is not feasible as the objective function (KL divergence) is defined on the non-negative orthant. Both challenges will be addressed in Section 3.3.

3. MARBLE

Marble is a sparse non-negative tensor factorization for count data. Marble decomposes an observed tensor into two terms, a bias (or offset) tensor and an interaction (or signal) tensor. The bias tensor represents the baseline characteristics common amongst the overall population and also provides computational stability. The interaction term is compromised of the R most prevalent phenotypes in the data (or medical concepts that are observed). Our model imposes sparsity constraints by reducing the "probabilistically unlikely" mode elements. Figure 2 illustrates the factorization of the patient by diagnosis by procedure tensor into R phenotypes and the bias vectors. We will first formulate the problem and provide a general overview of the algorithm. Next, we introduce the sparse factor representation and the augmented bias tensor. Then, we present the algorithm to solve the problem formulation. Finally, we illustrate how Marble can be used to perform high-throughput phenotyping in EHR data.

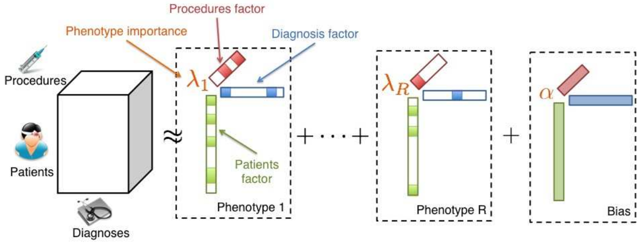


Figure 2: Deriving candidate phenotypes using Marble.

3.1 Problem Formulation and Overview

Let $\boldsymbol{\mathcal{X}}$ denote an observed tensor constructed from count data with size $I_1 \times I_2 \times \cdots \times I_N$ and $\boldsymbol{\mathcal{M}}$ represent a same-sized tensor of Poisson parameters for $\boldsymbol{\mathcal{X}}$. $\boldsymbol{\mathcal{M}}$ is split into two terms, a rank one bias tensor $\boldsymbol{\mathcal{C}}$ and a rank R interaction tensor $\boldsymbol{\mathcal{V}}$. The bias term, or baseline characteristics of the population, is composed of N positive vectors $\mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(N)}$ and a positive scalar α. The interaction term is similar to the Poisson decomposition tensor, where each rank one tensor is comprised of N stochastic vectors (elements sum to 1 and non-negative) with a non-negative weight λ_r. However, Marble constrains the feasible space of the vectors to either be zero or above some threshold value γ_n. The optimization problem is defined as

$$\min f(\boldsymbol{\mathcal{M}}) \equiv \sum_{\vec{i}} (m_{\vec{i}} - x_i \log m_{\vec{i}}) \tag{1}$$

$$\begin{aligned}\text{s.t } \boldsymbol{\mathcal{M}} &= \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{V}} \\ \boldsymbol{\mathcal{C}} &= [\![\alpha; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!] \in \Omega_C && (2) \\ \boldsymbol{\mathcal{V}} &= [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!] \in \Omega_V \\ \Omega_C &= \Omega_\alpha \times \Omega_{u1} \times \cdots \times \Omega_{uN} \\ \Omega_\alpha &= (0, +\infty) && (3) \\ \Omega_{un} &= \{\mathbf{u} \in (0,1]^{I_n \times 1} \mid ||\mathbf{u}||_1 = 1\} && (4) \\ \Omega_V &= \Omega_\lambda \times \Omega_{A1} \times \cdots \times \Omega_{AN} \\ \Omega_\lambda &= [0, +\infty)^R \\ \Omega_{An} &= \{\mathbf{A} \in \{0, [\gamma_n, 1]\}^{I_n \times R} \mid ||\mathbf{a}_{:r}||_1 = 1 \;\; \forall r\}. && (5)\end{aligned}$$

We solve the problem using an alternating minimization approach, cycling through each mode while fixing all other modes. For each mode, the algorithm first calculates the factor matrix associated with the interaction tensor. The matrix is then gradually projected onto the feasible space, described in Section 3.2, using a penalty method approach detailed in Section 3.4.2. Once the interaction factor matrix is computed, the bias vector is computed. After an iteration (where the algorithm has cycled through all the modes), the projection penalty is updated and the whole process is repeated until convergence occurs. A high-level view of the Marble algorithm is illustrated in Algorithm 1, with details described in Section 3.4.

Algorithm 1: Overview of Marble algorithm

```
while not converged do
    foreach mode n do
        Solve the nth interaction factor matrix (Section 3.2);
        Project onto sparse factors (Section 3.4.2);
        Solve nth bias vector (Section 3.3);
    end
    Calculate gradual projection penalty (Section 3.4.2);
end
```

3.2 Sparse Factor Representation

Sparse factor representations are desired to improve interpretability and address the problem where the CP-APR tensor factorization model produced "probabilistically unlikely" diagnoses and medications. The CP-APR model imposes a stochastic constraint on the columns of the factor matrices. For the rth rank one tensor, each non-zero element along the nth vector $\mathbf{a}_r^{(n)}$ represents a probabilistic estimate of the element's membership (e.g. probability the diagnosis diabetes belongs to this phenotype). Therefore, our model removes small non-zero elements to achieve sparse factor matrices, while also guaranteeing convergence to a local minimum. Marble modifies the stochastic constraints of CP-APR such that each nth factor matrix, $\mathbf{A}^{(n)}$, will have non-zero components that range from γ_n to 1 and the elements of each column sum to 1. Equation (5) captures the sparse factor representation constraint. Each mode can have a different threshold, γ_n, as the desired level of sparsity may depend both on domain constraints and the mode size.

3.3 Bias Tensor

Marble introduces a rank one bias tensor to capture baseline characteristics common amongst the overall population. Traditional tensor (or matrix) factorization approaches, which use a least squares loss, center the data by subtracting the feature mean from all the observations. Thus, the decomposition is only performed on the "signal" (or "interaction") aspects of the data. Unfortunately, the data centering technique is not feasible for Marble as the KL divergence is only defined for non-negative $x_{\vec{i}}$.

The bias tensor also provides computational stability for the "inadmissible zeros" resulting from the sparse factor representation. Under the assumption that γ_n in Equation (5) is set to be non-zero, the sparse factor representation will produce a limited number of non-zero elements. Given that R is not over-specified, the probability an element $m_{\vec{i}}$ is zero is high. If the corresponding observed tensor element is zero ($x_{\vec{i}} = 0$), then $m_{\vec{i}} - x_i \log m_{\vec{i}} = 0$. However, for a non-zero $x_{\vec{i}}$, the objective function is ill-defined as $\lim_{i \to 0} \log i = -\infty$, leading to an "inadmissible zero". Several works have dealt with this problem by either avoiding zeros [11], altering the updates [19], or shifting elements into the interior [6]. However, all these methods adversely effect the sparsity of the resulting factors.

We *augment* the Poisson tensor decomposition with an additional rank-one tensor, shown in Equation (2). The scalar constant α and the factor vectors $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \cdots, \mathbf{u}^{(N)}$ must be *strictly positive*, which are captured in Equations (3) and (4). Although similar in nature to an additional clinical phenotype, the bias tensor is not equivalent to increasing the rank of the original decomposition $\boldsymbol{\mathcal{V}}$ to $R+1$. Each factor vector ($\mathbf{u}^{(1)}$ where $n > 1$) represents the common baseline characteristics amongst the *entire* population (e.g., the overall likelihood of developing diabetes, getting a chest x-ray, etc) and captures the data bias, or offset, of the observed tensor. Furthermore, the positive augmented tensor stabilizes the optimization problem by avoiding inadmissible zeros in $\boldsymbol{\mathcal{M}}$ and allows sparse factor matrices in the interaction tensor, $\mathbf{A}^{(n)}$.

3.4 Algorithm

3.4.1 Alternating Minimization Updates

The optimization problem presented in Section 3.1 is solved via an alternating minimization approach, where all factor vectors/matrices are fixed except for the one being updated. For each mode, we compute the subproblem solution using an approach similar to CP-APR [6]. For the nth mode, we express the mode-n matricization $\mathbf{V}_{(n)} = \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)}$, where

$$\mathbf{B}^{(n)} = \mathbf{A}^{(n)} \mathbf{\Lambda}, \text{ where } \mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda}) \tag{6}$$

$$\mathbf{\Pi}^{(n)} = \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)} \right)^{\intercal}. \tag{7}$$

The weights ($\boldsymbol{\lambda}$) are absorbed into the nth factor matrix $\mathbf{B}^{(n)}$ in Equation (6) and we use the matrix $\mathbf{\Pi}^{(n)}$ to denote the fixed parts in Equation (7). Note that the size of matrix $\mathbf{B}^{(n)}$ is $I_n \times R$ while the size of matrix $\mathbf{\Pi}^{(n)}$ is $R \times \prod_{j=1, j \neq n}^{N} I_j$. Using the notation above, we can rewrite the objective function, Equation (1) as follows:

$$\mathbf{e}^{\intercal} \left[\mathbf{C}_{(n)} + \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)} - \mathbf{X}_{(n)} * \log \left(\mathbf{C}_{(n)} + \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)} \right) \right] \mathbf{e},$$

where $\mathbf{e}$ is a vector of all ones, $\mathbf{C}_{(n)}$ is the mode-n matricization of the bias tensor and $\mathbf{X}_{(n)}$ is the mode-n matricization of the observed tensor.

The multiplicative update derivation for the weighted factor matrix $\mathbf{B}^{(n)}$ can be computed by taking the partial derivative of the objective function respect to a single element b_{gh}.

$$\frac{\partial f}{\partial b_{gh}} = \sum_{\ell} \pi_{h\ell} - \sum_{\ell} x_{g\ell} \frac{\pi_{h\ell}}{c_{g\ell} + \sum_r b_{gr} \pi_{r\ell}}.$$

Setting the gradient descent step size set to $\frac{b_{gh}}{\sum_{\ell} \pi_{h\ell}}$ yields the multiplicative update:

$$b_{gh} = b_{gh} \left[\sum_{\ell} x_{g\ell} \frac{\pi_{h\ell}}{c_{g\ell} + \sum_r b_{gr} \pi_{r\ell}} \right].$$

The non-negative factor matrix constraints are satisfied using the multiplicative update. Generalizing for the entire factor matrix, the update equation is as follows:

$$\mathbf{B}^{(n)} = \mathbf{B}^{(n)} * \left[\mathbf{X} \oslash (\mathbf{C}_{(n)} + \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)}) \right] \mathbf{\Pi}^{(n)\intercal}. \tag{8}$$

We adopt the same alternating update strategy for the bias tensor $\boldsymbol{\mathcal{C}}$. In particular, we update the mode n bias vector $\mathbf{u}^{(n)}$ holding other modes constant, with the fixed parts denoted as $\mathbf{\Psi}^{(n)}$.

$$\mathbf{C}_{(n)} = \alpha \mathbf{u}^{(n)} \mathbf{\Psi}^{(n)}$$

$$\mathbf{\Psi}^{(n)} = \left(\mathbf{u}^{(N)} \odot \cdots \odot \mathbf{u}^{(n+1)} \odot \mathbf{u}^{(n-1)} \odot \cdots \odot \mathbf{u}^{(1)} \right)^{\intercal} \tag{9}$$

The update for the augmented bias vector follows a similar derivation as $\mathbf{B}^{(n)}$.

$$\mathbf{u}^{(n)} = \mathbf{u}^{(n)} * \left[\mathbf{X}_{(n)} \oslash (\alpha \mathbf{u}^{(n)} \mathbf{\Psi}^{(n)} + \mathbf{V}_{(n)})\right] \mathbf{\Psi}^{(n)\intercal}. \tag{10}$$

3.4.2 Gradual Projection

The alternating minimization updates satisfy the non-negative constraints of the weights ($\boldsymbol{\lambda}$), factor matrices $\mathbf{A}^{(n)}$, and basis vectors $\mathbf{u}^{(n)}$. However, the factor matrices will not satisfy the modified stochastic constraints in Equation (5). The projected gradient descent method can be used to threshold the factor matrices after each alternating minimization update. Experimental results (shown in Section 4) show that zeroing out components too early in the iterative process negatively impacts the quality of the factor representation. Instead, Marble uses a penalty method approach [2] to gradually adjust the projection threshold at each iteration. During the early iterations, when the factor representations are changing drastically, the “feasible set” is closer to the $[0, 1]$ range. As the factor representations start to stabilize (difference in objective function starts to approach zero), the projection occurs on the feasible set described in Equation (5). A new scalar ξ is introduced to calculate the gradual projection, where each iteration uses the threshold $\xi\gamma_n$. The range of ξ is [0,1], where zero represents no projection and 1 represents the full projection. After each iteration k where all the modes have been cycled through, we update ξ as follows:

$$\begin{aligned}
\kappa^{(k)} &= 1 - \frac{|f(\boldsymbol{\mathcal{M}}^{(k-1)}) - f(\boldsymbol{\mathcal{M}}^{(k)})|}{|f(\boldsymbol{\mathcal{M}}^{(k-1)})|} \\
\xi^{(k+1)} &= \max(\xi^{(k)}, \frac{1}{2}\xi^{(k)} + \frac{1}{2}\kappa^{(k)}). \qquad (11)
\end{aligned}$$

Marble uses a moving average of the ξ to minimize drastic changes and also ensures that the penalty is non-decreasing at each iteration.

3.4.3 Algorithm Details

Subproblem Convergence. Marble performs several subproblem iterations (user-defined maximum L) at each mode n for both the factor matrix $\mathbf{A}^{(n)}$ and the bias vector $\mathbf{u}^{(n)}$. Empirical evidence suggests extra inner iterations can accelerate the convergence [6]. The benefit of subproblem iterations on simulation data is presented in Section 4. The exit criterion (Karush-Kuhn-Tucker (KKT) conditions) for the subproblem iterates is similar to the CP-APR algorithm [6]. For convenience, we introduce the following to matrices:

$$\mathbf{\Phi}^{(n)} = \left[\mathbf{X}_{(n)} \oslash (\alpha \mathbf{u}^{(n)} \mathbf{\Psi}^{(n)} + \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)})\right] \mathbf{\Pi}^{(n)\intercal} \tag{12}$$

$$\mathbf{Z}^{(n)} = \left[\mathbf{X}_{(n)} \oslash (\alpha \mathbf{u}^{(n)} \mathbf{\Psi}^{(n)} + \mathbf{B}^{(n)} \mathbf{\Pi}^{(n)})\right] \mathbf{\Psi}^{(n)\intercal}, \tag{13}$$

which capture the multiplicative terms in Equations (8) and (10) respectively. These matrices are used to check convergence of the subproblem, with the algorithm exiting under the following conditions:

$$\begin{aligned}
\min(\mathbf{A}^{(n)}, \mathbf{E} - \mathbf{\Phi}^{(n)}) &= \mathbf{0} \\
\min(\mathbf{u}^{(n)}, \mathbf{E} - \mathbf{Z}^{(n)}) &= \mathbf{0},
\end{aligned}$$

where $\mathbf{E}$ represents a matrix of all ones. Note that min is the element-wise minimum of the two matrices, and the result should be a matrix of all zeros. We relax the zero equality for practical purposes and check for convergence within a certain tolerance, e.g. $\min(\mathbf{u}^{(n)}, \mathbf{E} - \mathbf{Z}^{(n)}) \leq$ kktTol. The detailed Marble algorithm is presented in Algorithm 2.

Algorithm 2: Detailed Marble algorithm

```
Data: $\boldsymbol{\mathcal{X}}, R, \alpha, \boldsymbol{\gamma}$
Result: $\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{C}}$
for $k = 1, 2, \cdots, K$ do
    lastObj $\leftarrow 0$
    $\xi \leftarrow 0$
    //For each mode $n$
    for $n = 1, \cdots, N$ do
        $\hat{\Omega}_n \leftarrow \{0, [\xi\gamma_n, 1]\}^{I_n \times R}$
        Set $\mathbf{\Psi}^{(n)}$ using Equation (9)
        Set $\mathbf{B}^{(n)}$ using Equation (6)
        Set $\mathbf{\Pi}^{(n)}$ using Equation (7)
        //Solve $n$th interaction factor matrix
        for $\ell = 1, \cdots, L$ do
            Calculate $\mathbf{\Phi}^{(n)}$ using Equation (12)
            if $\min(\mathbf{B}^{(n)}, \mathbf{E} - \mathbf{\Phi}^{(n)}) \leq$ kktTol then break
            $\mathbf{B}^{(n)} \leftarrow \mathbf{B}^{(n)} * \mathbf{\Phi}^{(n)}$
        end
        //Project onto sparse factors
        $\mathbf{B}^{(n)} \leftarrow P_{\hat{\Omega}_n}(\mathbf{B}^{(n)})$
        $\boldsymbol{\lambda} \leftarrow \mathbf{e}^\intercal \mathbf{B}^{(n)}$
        $\mathbf{A}^{(n)} \leftarrow \mathbf{B}^{(n)} \mathbf{\Lambda}^{-1}$
        //Solve $n$th bias vector
        for $\ell = 1, \cdots, L$ do
            Calculate $\mathbf{Z}^{(n)}$ using Equation (13)
            if $\min\left(\mathbf{u}^{(n)}, \mathbf{E} - \mathbf{Z}^{(n)}\right) \leq$ kktTol then break
            $\mathbf{u}^{(n)} \leftarrow \mathbf{u}^{(n)} * \mathbf{Z}^{(n)}$
        end
        $\mathbf{u}^{(n)} \leftarrow \frac{\mathbf{u}^{(n)}}{||\mathbf{u}^{(n)}||_1}$
    end
    obj $\leftarrow f(\boldsymbol{\mathcal{M}})$
    if |obj $-$ lastObj| $<$ convergenceTol then break
    //Calculate projection penalty
    Update $\xi$ using Equation (11)
    lastObj $\leftarrow$ obj
end
```

Sparse Implementation. When $\boldsymbol{\mathcal{X}}$ is a sparse tensor, we construct $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{C}}$ using the non-zero elements of the observed tensor and avoid constructing $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{C}}$ explicitly. Our algorithm adopts the sparse tensor implementation approach presented in [1, 6]. The only calculations necessary are the ones that correspond to the non-zero elements in $\boldsymbol{\mathcal{X}}$. We can store $\boldsymbol{\mathcal{X}}$ as a set of values and indices (v^q, c), where Q are the non-zero elements of the tensor. We then form Q rows of $\mathbf{\Pi}$ and $\mathbf{Z}$ that correspond to each non-zero element of $\boldsymbol{\mathcal{X}}$. The qth vector of $\mathbf{\Pi}$ and $\mathbf{Z}$ are:

$$\begin{aligned}
w^{(q)} &= \mathbf{a}^{(1)}_{(\vec{i}^{(q)}_1 :)} * \cdots * \mathbf{a}^{(n+1)}_{(\vec{i}^{(q)}_{n+1} :)} * \mathbf{a}^{(n-1)}_{(\vec{i}^{(q)}_{n-1} :)} * \cdots * \mathbf{a}^{(N)}_{(\vec{i}^{(q)}_N :)} \\
y^{(q)} &= \mathbf{u}^{(1)}_{(\vec{i}^{(q)}_1 :)} * \cdots * \mathbf{u}^{(n+1)}_{(\vec{i}^{(q)}_{n+1} :)} * \mathbf{u}^{(n-1)}_{(\vec{i}^{(q)}_{n-1} :)} * \cdots * \mathbf{u}^{(N)}_{(\vec{i}^{(q)}_N :)},
\end{aligned}$$

where $\mathbf{a}^{(1)}_{(\vec{i}_1^{(q)}:)}$ is the $\vec{i}_1^{(q)}$ row of $\mathbf{A}^{(1)}$. Thus, we can calculate element (i, r) of $\mathbf{\Pi}$ and $\mathbf{Z}$ as:

$$v^{(q)} = x^{(q)}/(<\mathbf{y}^{(q)}, \alpha\mathbf{u}^{(n)}_{(i_n^{(q)}:)}> + <\mathbf{w}^{(q)}, \mathbf{B}^{(n)}_{(i_n^{(q)}:)}>)$$

$$\Phi_{jr} = \sum_{q:i_n^{(q)}=j} v^{(q)}\mathbf{w}^{(q)}$$

$$z_{jr} = \sum_{q:i_n^{(q)}=j} v^{(q)}\mathbf{y}^{(q)}$$

Note that we require storage for the augmented tensor, which entails storing $\mathbf{y}^{(q)}$. Thus, we require Q additional storage compared to CP-APR.

Computational Complexity. Note that the subproblem iterates in the CP-APR model is the computational bottleneck of the algorithm. In particular, calculating $\mathbf{\Phi}$ requires on the order of the decomposition rank R times the product of the dimension of each tensor mode I_n. If we denote the size of the largest mode as D, then CP-APR has the computational complexity of $\mathcal{O}\left(D^N\right)$. Although our algorithm has to compute the additional augmented vector, the computational complexity remains the same as CP-APR.

3.5 Application to EHR-Phenotyping

While Marble is a general non-negative sparse tensor factorization model to fit count data, we motivated the problem to simultaneously derive multiple EHR phenotypes with minimal human intervention. We briefly describe the construction of an EHR count tensor and the resulting candidate phenotypes using our algorithm. Figure 3 provides a conceptual illustration of the high-throughput phenotyping process. Each patient is anchored using an index date (i.e. hospital admit date) and the observation window can be defined as a fixed time window either before or after the index date depending on the application. Any data that occurs during the observation window is used during the construction process. The EHR tensor is then constructed using the count of the co-occurrences between the various modes. In Figure 2, each tensor element represents the number of times either a normal or abnormal clinical measurement occurred.

Once the tensor is constructed, we can use Marble to fit a non-negative Poisson tensor decomposition to the data. The resultant tensor $\boldsymbol{\mathcal{V}}$ is then used to define R candidate phenotypes, similar to Figure 1. Thus, the rth candidate phenotype is defined using the non-zero elements of the rth column from all N factor matrices.

New patients can be projected onto the tensor-derived phenotypes to obtain a *phenotype membership* vector. We define the phenotype membership vector as the convex combination of the tensor-derived phenotypes, where the rth element denotes the probability the patient exhibits characteristics consist with the rth phenotype. For notation purpose, we will assume the patient mode is the first mode of the tensor. Thus given a new patient's tensor $\hat{\boldsymbol{\mathcal{X}}}$, we want to find $\hat{\boldsymbol{\lambda}}$ and $\hat{\mathbf{a}}^{(1)}$ that provides that best approximates the new patient's tensor. Our projection also needs to determine $\hat{\alpha}$, the strength of the bias in $\hat{\boldsymbol{\mathcal{X}}}$. We observe that this is equivalent to the optimization subproblem for the first mode with several noticeable differences: (1) the phenotype membership vector is obtained by normalizing the entries of $\hat{\mathbf{b}}^{(1)}$ across all R phenotypes instead of the standard column normalization for each phenotype, (2) setting the augmented vector $(\hat{u}^{(1)})$ to 1 and absorbing the weight into $\hat{\alpha}$, and (3) ignoring the projection onto the feasible set defined in the optimization problem.

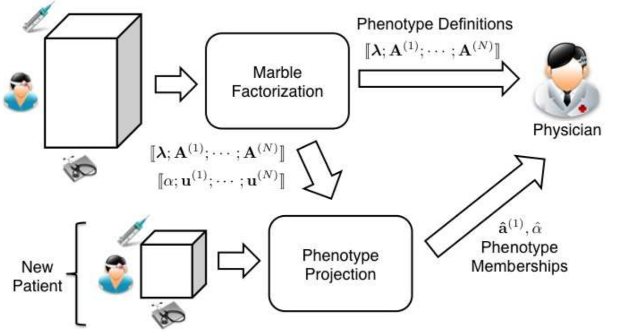

Phenotype Definitions
$[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]$
Marble Factorization
Physician
$[\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]$
$[\![\alpha; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!]$
New Patient
Phenotype Projection
$\hat{\mathbf{a}}^{(1)}, \hat{\alpha}$
Phenotype Memberships

Figure 3: A high-level depiction of using Marble to generate high-throughput phenotypes.

4. EXPERIMENTS

In this section, we will first evaluate the algorithmic performance using a simulated dataset where the actual tensor factors are known. Then, we evaluate the phenotyping performance using a realistic EHR dataset.

Evaluation Metric Details. Definitions for the evaluation metrics are provided below.

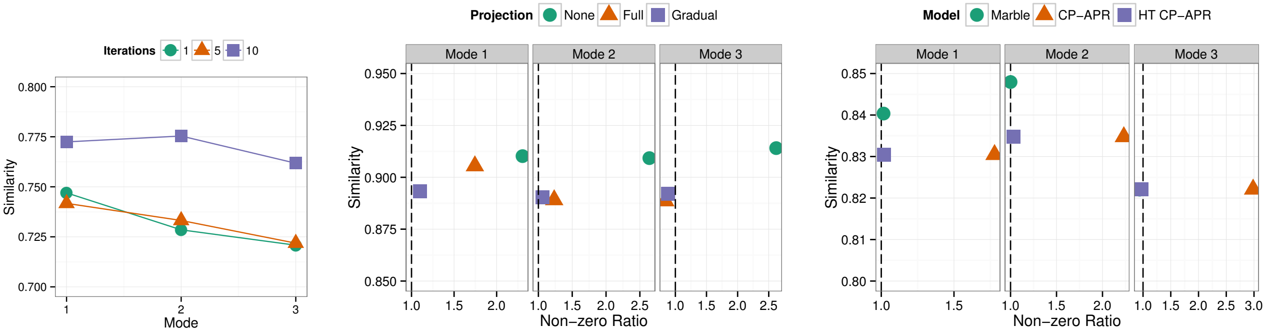
$$\text{Non-zero Ratio} = \frac{\text{\# Non-zeros in Computed Solution}}{\text{\# Non-zeros in Actual Solution}}$$

$$\text{Similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^{\intercal}\mathbf{b}}{||\mathbf{a}||||\mathbf{b}||}$$

Similarity is calculated using the cosine similarity, a component of the factor match score (FMS), to quantify the similarity between the computed solution and the actual factor representation. FMS is commonly used to quantify the closeness of the computed solution and provides a single number between $[0, 1]$ [6]. However, FMS is an aggregate measure and can mask the mode-specific similarity results. For our analysis, we pair the computed rank-one tensors with the true rank-one tensors using a greedy algorithm, providing a lower bound on the similarity scores.

4.1 Simulation

First, we analyze simulated data where the underlying factor representation is known. Specifically, we consider a third-order tensor of size 100 × 80 × 60 with rank of 10 ($R = 10$). We generate the model $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{V}}$, where $\boldsymbol{\mathcal{C}} = [\![\alpha; \mathbf{u}^{(1)}; \cdots; \mathbf{u}^{(N)}]\!]$ and $\boldsymbol{\mathcal{V}} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}; \cdots; \mathbf{A}^{(N)}]\!]$. Each factor matrix $\mathbf{A}^{(n)}$ is generated as follows: (1) Sample the non-zero element indices for each column according to the sparsity pattern for the mode; (2) From the sampled indices, randomly select 10% of the entries (or minimum of one) to sample uniformly from the interval $[0, 10]$ to mirror real EHR characteristics (e.g. one or two diagnosis contribute heavily to a phenotype); (3) For the remaining indices, sample uniformly from $[0, 1]$; and (4) Normalize the column so the elements sum to 1, and absorb the weight into $\boldsymbol{\lambda}$. Each augmented vector is chosen in a similar fashion except the weight is set to $\alpha = 2$. The full tensor $\boldsymbol{\mathcal{M}}$ is calculated from the factor matrices and the augmented vectors. Then each tensor element x_{ijk} is sampled from the Poisson distribution



(a) Inner Iterations

(b) Projection Type

(c) Comparing Marble and CP-APR.

Figure 4: Simulation results on a third-order tensor

with the parameter set to m_{ijk}. Ten observation tensors are generated from the tensor $\boldsymbol{\mathcal{M}}$.

4.1.1 Inner iteration benefits

The number of maximum subproblem iterations (L) represents the “closeness” to the subproblem solution. Thus, a single subproblem iteration (equivalent to the Lee-Seung multiplicative update [18]) only takes a step towards the subproblem solution as observed by [6]. For the simulated data, the average computation time (in seconds) for 1, 5, and 10 subproblem iterations was 37.6, 89.4, and 92.56 respectively. Additional inner iterations do not accelerate convergence, as the computational time increase with the number of iterations. However, Figure 4a illustrates the benefit of extra subproblem iterations on the similarity scores from the true solution. The similarity score at 10 inner iterations is the highest across all the modes. Therefore, more subproblem iterations improves the quality of the resulting tensor decomposition.

4.1.2 Gradual projection benefits

Empirical evidence for the gradual projection approach is presented in Figure 4b. We compare the gradual projection described in Section 3.4.2 to (i) no projection where $\xi = 0$ for all iterations, and (ii) full projection where $\xi = 1$ for all iterations. No projection yields the best similarity score across all three modes. However, the gradual projection approach results in tensor factors that are near the ideal non-zero ratio of 1. It also has higher similarity scores on two of the modes compared to full projection. Even though the observation matrix was generated from sparse factor representations with a relatively small bias effect ($\alpha = 2$), without any projection yields factor representations that on average contain at least 2× more non-zero elements in each column. The results demonstrate that the gradual projection approach can reproduce the same sparsity ratio (number of non-zero elements / size of mode) as the true solution but sacrifices in terms of similarity to the true solution.

4.1.3 CP-APR comparison

Next, we perform a comparison with the CP-APR model. The multi-layer sparse NTF model [7] is omitted due to the added computational complexity of their model. Using a sparser underlying factor representation further highlights the differences between Marble and CP-APR. Marble is slower than CP-APR, primarily due to the computation of the augmented tensor decomposition. On the simulated data, Marble takes about 58 seconds to converge to a local optimal, while CP-APR obtains a solution in 45 seconds. However, in comparison to current phenotyping approaches, the 15 second difference is negligible given that a single disease phenotype can take months to develop.

Figure 4c displays a plot of the two algorithms based on the non-zero ratio used in Figure 4b. Marble’s sparse factor representation results in a higher similarity score for the first and second mode. All the Marble modes are close to the ideal non-zero ratio of 1. Furthermore, Marble achieves a 42.8%, 55.1%, and 68.4% reduction on the non-zero ratio. Thus, the non-zero pattern across all three modes is better captured by our model, whereas CP-APR results in a higher number of non-zero elements.

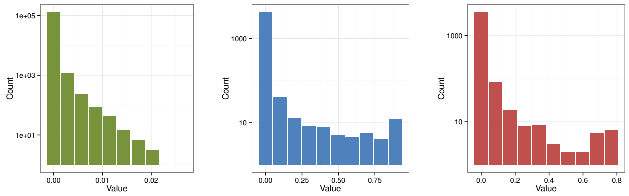
4.2 CMS Claim Records

The Centers for Medicare and Medicaid Services (CMS) provides the *CMS Linkable 2008-2010 Medicare Data Entrepreneurs’ Synthetic Public Use File (DE-SynPUF)*, a publicly available dataset that spans 3 years and contains inpatient, outpatient, carrier, and prescription drug event claims in addition to the beneficiary summary files[3]. The claims records have been synthesized from 5% of the 2008 Medicare population to protect the privacy of the beneficiaries. Although the relationships between some of the variables have been altered to minimize re-identification risk, the sheer volume of patients can still provide interesting and insightful phenotypes.

Our experiments focus on a random subset of 10,000 patients from Sample 1 (CMS released the data in 20 separate samples). We construct the tensor from the carrier claims records using the diagnosis and procedure codes. Individual International Classification of Diseases (ICD-9) diagnosis codes and Healthcare Common Procedure Coding System (HCPCS) procedure codes capture information at a fine-grained level. Thus, similar diagnosis codes (procedure codes) were grouped using the Unified Medical Language System[4] to aggregate the individual ICD-9 codes and HCPCS codes to higher level medical hierarchies. Therefore,

[3] A detailed description can be found on their website.

[4] The Metathesaurus contains the source vocabularies for 150 sources, including ICD-9-CM and HCPCS.



(a) Patient (b) Diagnosis (c) Procedure

Figure 5: The distribution of factor elements along the three CMS tensor modes. Zeros entries are omitted from the plot.

the constructed tensor is 10,000 patients by 129 diagnoses by 115 procedures.

4.2.1 Threshold selection

The Marble algorithm computes the sparse factor representation using predefined thresholds γ_n. These thresholds provide a tunable knob to adjust the sparsity of the candidate phenotypes. Domain constraints can be used to determine the threshold value (e.g. a phenotype should only contain a maximum of 3 unique diagnoses). However, given the absence of domain knowledge, the Marble algorithm can be used with $\gamma_n = 0$ for all n. A plot of the non-zero elements distribution along each mode can be used to determine the thresholds. Figure 5 shows the mean histogram of the CP-APR non-zero factor values along the three modes using $R = 50$ using the 10 subsamples. For all three plots, there is a noticeable difference in size (the y-axis uses a log scale) between the first two bins, which suggests the threshold occur at the start of the second bin. Thus, for the remainder of the paper, the thresholds used are $\boldsymbol{\gamma} = [0.0001, 0.01, 0.01]$.

4.2.2 Predictive performance

The phenotypes are evaluated on a classification task of predicting high cost (above 75th percentile) beneficiaries. Our algorithm is compared against (i) CP-APR derived phenotypes and (ii) the raw feature matrix with 129 × 115 columns, where each column represents a diagnosis-procedure combination. 10 random subsamples are obtained via stratified sampling with a 50-50 train test split, and the phenotypes are derived from the training dataset only. An ℓ_1 regularized logistic regression model is trained separately on each of the three feature sets (the phenotype membership matrix is the feature matrix for both Marble and CP-APR) and the predictive performance is evaluated on the test set.

Figure 6 displays a plot of the area under the receiver operating characteristic curve (AUC) as a function of the number of phenotypes (R). The predictive performance of both tensor factorization models using 50 phenotypes is similar to baseline, providing a 300× feature reduction from the original raw feature matrix. Beyond 50 phenotypes, the accuracy gradually increases with the number of phenotypes. Marble takes approximately 3.5 hours[5] to factor the data using 50 phenotypes. For the remainder of this paper, we use $R = 50$ to provide detailed analysis of the individual phenotypes.

Marble and CP-APR perform similarly in terms of predictive power. Table 2 illustrates a comparison between the first, or highest λ_r, Marble-derived phenotype and a "similar" CP-APR derived phenotype, where FMS is used to quantify overall closeness of the phenotypes. The CP-APR phenotype contains 10× more diagnosis and procedure elements because there is no sparsity constraint on the factorization. In contrast, a medical professional can easily digest the contents of the phenotype resulting from the Marble algorithm. Marble yields concise phenotypes without losing predictive power.

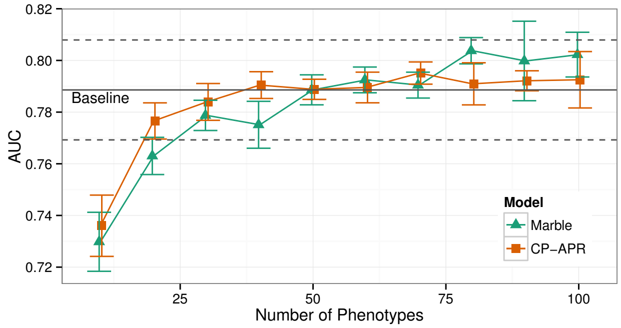


Figure 6: Area under the receiver operating characteristic curve for the three feature sets as a function of the number of phenotypes. The error bars display the 95% confidence interval.

4.2.3 Data Bias

An added benefit of Marble is that it captures the baseline characteristics that exist in the overall population via the rank-one bias tensor. Table 3 shows the 10 highest valued elements from the diagnosis and procedure mode, in decreasing magnitude. The diagnosis bias vector shows that Medicare patients generally visit clinics because of various symptoms and complications. Furthermore, the diagnosis vector contains several chronic diseases common in the elderly population, such as hypertension, arthritis (arthropathies), heart disease, and diabetes (disease of other endocrine glands). Centers for Disease Control and Prevention (CDC) estimate that 80% of older adults suffer from at least one chronic condition and 50% have two or more chronic conditions [4]. The procedure basis vector also contains procedure codes relevant to the treatment of patients with chronic conditions. We verified the results in Table 3 with chronic disease reports provided by CDC[6].

4.2.4 Chronic disease performance

The United States spends more than 2.1 trillion dollars (75% of medical care) on the treatment of chronic diseases [4]. Thus, obtaining phenotypes related to chronic disease such as heart failure, diabetes, and cancer can help medical professionals tailor treatment options based on the patient's phenotypes. The CMS dataset provides chronic disease indicators[7] that we will use to identify phenotypes associated with specific chronic diseases.

Table 4 illustrates two of the heart-failure related phenotypes, which maps to varying degrees of disease severity. Pa-

[5] A computer with an Intel® Xeon® Processor X5660 and 8 GB RAM.

[6] The latest disease reports are located at `http://www.cdc.gov/chronicdisease/resources/publications/aag.htm`

[7] A patient's chronic condition flag cannot be perfectly reproduced due to the synthetic claim process used.

Table 2: Comparison of two tensor-derived phenotypes (blue and red text correspond to non-zero elements in the diagnosis and procedure factors, respectively)

Marble Phenotype
Other metabolic and immunity disorders
Hypertensive disease
Complications of surgical and medical care
Chemistry Pathology and Laboratory Tests
Organ or Disease Oriented Panels
Hematology and Coagulation Procedures
Surgical Procedures on the Cardiovascular System

CP-APR Phenotype
Diseases of the blood and blood-forming organs
Nonspecific abnormal findings
Other diseases of digestive system
· · · 24 total diagnoses
Chemistry Pathology and Laboratory Tests
Hematology and Coagulation Procedures
Organ or Disease Oriented Panels
Surgical Procedures on the Cardiovascular System
· · · 63 total procedures

Table 3: Top 10 elements for the augmented bias tensor

Diagnosis Mode
Symptoms
Complications of surgical and medical care
Arthropathies and related disorders
Other forms of heart disease
Dorsopathies
Disorders of the human eye
Diseases of other endocrine glands
Hypertensive disease
Other metabolic and immunity disorder
Other diseases of urinary system

Procedure Mode
Evaluation and Management of Other Outpatient Services
Diagnostic Radiology Procedures
Hospital Inpatient Services
Chemistry Pathology and Laboratory Tests
Physical Medicine and Rehabilitation Procedures
Surgical Procedures on the Cardiovascular System
Cardiovascular Procedures
Emergency Department Services
Nursing Facility Services
Hematology and Coagulation Procedures

tients with the second phenotype (titled severe heart failure) require hospital stays (inpatient services) and have added complications from lung disease. Table 5 depicts two other chronic disease phenotypes. The diabetes phenotype describes patients with complications resulting from diabetes, as the procedures include organ or disease oriented panels. The second phenotype, associated with arthritis, suggests that patients belong to this phenotype are undergoing rehabilitation to strengthen their joints. Furthermore, all four phenotypes shown are concise, easily interpretable, and map to known characteristics of the chronic disease.

5. CONCLUSION

This paper presented Marble, a novel sparse non-negative tensor factorization model to fit EHR count data. Our algorithm offers a data-driven solution to simultaneously generate multiple phenotypes from a diverse EHR population without expert supervision. The resulting phenotypes are concise, intuitive, and interpretable; and automatically reveal patient clusters on specific diagnoses and procedures. Furthermore, Marble captures the baseline characteristics of the overall population via an augmented bias tensor.

The experimental results on simulated data and 10,000 patient records from the CMS De-SYNPUF dataset demonstrate the conciseness, interpretability, and predictive power of Marble-derived phenotypes. They underscore the promise of Marble for high-throughput phenotyping with minimal human intervention. Marble can potentially be used to rapidly characterize, predict, and manage a large number of diseases, thereby promising a novel, data-driven solution that can benefit very large segments of the population. Future work will focus on generalizing the sparse non-negative tensor factorization to multi-relational tensors [20] to incorporate multiple EHR data sources and examine quasi-Newton methods to improve computational speed of the algorithm [12].

Acknowledgements

We would like to thank Yubin Park and Suriya Gunasekar for their comments. This research is supported by the Schlumberger Centennial Chair in Engineering; Army Research Office under grant W911NF-11-1-0258; and Department of Defense award under award number 60036907.

6. REFERENCES

[1] B. W. Bader and T. G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 2007.

[2] C. L. Byrne. Alternating Minimization as Sequential Unconstrained Minimization: A Survey. *Journal of Optimization Theory and Applications*, 156(3):554–566, Mar. 2013.

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

[4] Centers for Disease Control and Prevention (CDC). Chronic diseases at a glance 2009. Technical report, CDC, Feb. 2009.

[5] Y. Chen, R. J. Carroll, E. R. M. Hinz, A. Shah, A. E. Eyler, J. C. Denny, and H. Xu. Applying active learning to high-throughput phenotyping algorithms for electronic health records data. *JAMIA*, 20(e2):e253–e259, Dec. 2013.

Table 4: Two heart failure related phenotypes

Heart Failure Phenotype
Other forms of heart disease
Complications of surgical and medical care
Symptoms
Cardiovascular Procedures
Hematology and Coagulation Procedures
Evaluation and Management of Other Outpatient Services
Surgical Procedures on the Cardiovascular System
Chemistry Pathology and Laboratory Tests

Severe Heart Failure Phenotype
Other forms of heart disease
Pneumoconioses and other lung diseases
Ill-defined and unknown causes of morbidity and mortality
Hospital Inpatient Services
Cardiovascular Procedures

Table 5: Two chronic disease related phenotypes

Diabetes Phenotype
Diseases of other endocrine glands
Complications of surgical and medical care
Chemistry Pathology and Laboratory Tests
Organ or Disease Oriented Panels
Hematology and Coagulation Procedures
Surgical Procedures on the Cardiovascular System

Arthritis Phenotype
Arthropathies and related disorders
Physical Medicine and Rehabilitation Procedures
Evaluation and Management of Other Outpatient Services
Surgical Procedures on the Musculoskeletal System
Diagnostic Radiology Procedures

[6] E. C. Chi and T. G. Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

[7] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari. Novel multi-layer non-negative tensor factorization with sparsity constraints. In *ICANNGA 2007*, pages 271–280. Springer, 2007.

[8] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari. *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. Wiley, 2009.

[9] I. Davidson, S. Gilpin, O. Carmichael, and P. Walker. Network discovery via constrained tensor analysis of fMRI data. In *KDD 2013*, Aug. 2013.

[10] J. C. Denny. Mining electronic health records in the genomics era. *PLoS Computational Biology*, 8(12):e1002823–e1002823, Dec. 2012.

[11] N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural Computation*, 24(4), Apr. 2012.

[12] S. Hansen, T. Plantenga, and T. G. Kolda. Newton-Based Optimization for Nonnegative Tensor Factorizations. *arXiv*, Apr. 2013.

[13] R. 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.

[14] J. C. Ho, J. Ghosh, S. Steinhubl, W. Stewart, J. C. Denny, B. A. Malin, and J. Sun. Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of Biomedical Informatics*, accepted.

[15] G. Hripcsak and D. J. Albers. Next-generation phenotyping of electronic health records. *JAMIA*, 20(1):117–121, Dec. 2012.

[16] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos. Gigatensor: Scaling tensor analysis up by 100 times-algorithms and discoveries. In *KDD 2012*, pages 316–324, 2012.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct. 1999.

[19] C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596, Nov. 2007.

[20] Y.-R. Lin, J. Sun, H. Sundaram, A. Kelliher, P. Castro, and R. Konuru. Community discovery via metagraph factorization. *ACM Transactions on Knowledge Discovery from Data*, 5(3), Aug. 2011.

[21] M. Mørup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):24–40, 2011.

[22] K. M. Newton, P. L. Peissig, A. N. Kho, S. J. Bielinski, R. L. Berg, V. Choudhary, M. Basford, C. G. Chute, I. J. Kullo, R. Li, J. A. Pacheco, L. V. Rasmussen, L. Spangler, and J. C. Denny. Validation of electronic medical record-based phenotyping algorithms: results and lessons learned from the eMERGE network. *JAMIA*, 20(e1):e147–e154, June 2013.

[23] D. Wang and S. Kong. Feature selection from high-order tensorial data via sparse decomposition. *Pattern Recognition Letters*, 33(13):1695–1702, 2012.

[24] Z. Xu, F. Yan, Yuan, and Qi. Infinite Tucker Decomposition: Nonparametric Bayesian Models for Multiway Data Analysis. In *ICML 2012*, pages 1023–1030. Alan, 2012.