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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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, 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, 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

The phenotype definition can be found in Phenotype KnowledgeBase.

2The eMerge network explores the use of EHR to obtain phenotypic information at multiple medical institutions.
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 and real EHR data. Finally, we summarize our work in Section 5.

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 \( A \), we denote the \((i, j)\)th element as \( a_{ij} \), the \(j\)th column as \( \mathbf{a}_j \), and the \(i\)th 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, \( \mathbf{X} \approx \mathbf{W} \mathbf{H} \). 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, \( \mathbf{a}^{(1)} \odot \mathbf{a}^{(2)} \odot \cdots \mathbf{a}^{(N)} \), produces a \( N \)th order tensor \( \mathbf{X} \) where each element \( x_{i_1,i_2,\ldots,i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \cdots a_{i_N}^{(N)} \).

Definition 2. The element-wise multiplication (and division) of two same-sized matrices \( \mathbf{A} \odot \mathbf{B} \) produces a matrix \( \mathbf{Z} \) of the same size such that the element \( z_{ij} = a_{ij} b_{ij} \) for all \( i \).

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 \( \mathbf{Z} = [a_1 \otimes b_1 \cdots a_R \otimes b_R] \), where \( \otimes \) represents the Kronecker product. The Kronecker product of two vectors \( \mathbf{a} \odot \mathbf{b} = \begin{bmatrix} a_1 b \\ a_2 b \\ \vdots \\ a_R b \end{bmatrix} \).

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha, \gamma )</td>
<td>scalar</td>
</tr>
<tr>
<td>( \lambda, u, a )</td>
<td>vector</td>
</tr>
<tr>
<td>( A, B, Z, \Lambda, \Pi, \Phi, \Psi )</td>
<td>matrix</td>
</tr>
<tr>
<td>( \mathbf{X}, \mathbf{M}, \mathbf{C}, \mathbf{V} )</td>
<td>tensor</td>
</tr>
<tr>
<td>( i, j )</td>
<td>tensor element index ((i_1, i_2, \ldots, i_N))</td>
</tr>
<tr>
<td>( x_{i} )</td>
<td>tensor element at index ( i )</td>
</tr>
<tr>
<td>( \mathbf{X}^{(n)}, \mathbf{M}^{(n)}, \mathbf{C}^{(n)} )</td>
<td>mode-( n ) matricization of tensor</td>
</tr>
<tr>
<td>( \odot )</td>
<td>element-wise multiplication</td>
</tr>
<tr>
<td>( \oslash )</td>
<td>element-wise division</td>
</tr>
<tr>
<td>( \odot )</td>
<td>outer product</td>
</tr>
<tr>
<td>( \odot )</td>
<td>Khatri-Rao product</td>
</tr>
<tr>
<td>( &lt; \mathbf{x}, \mathbf{y} &gt; )</td>
<td>inner product of ( \mathbf{x}, \mathbf{y} )</td>
</tr>
</tbody>
</table>
As lim

ffi

small

are achieved via an exponential update (i.e. sor factorization stages are required and sparsity constraints approach is computationally expensive because multiple ten-

linearly transformed gradient descent approach [7]. Yet, the ing KL divergence as an objective function. A multi-layer of existing work regarding sparse factor representations us-

been recently proposed [24]. Nonetheless, there is a paucity Bayesian approaches to sparse Tucker decomposition have the standard least-squares minimization. Non-parametric representations. However, additional sparsity may be desired

fit sparse count data [6]. Details of the algorithm and model can be found in [8]. Our paper will focus on the nonnega-

tive CP alternating Poisson regression (CP-APR) model to can be expressed as

\[ X \approx \sum_{r=1}^{R} \lambda_r a_r^{(1)} \otimes \ldots \otimes a_r^{(N)} \]

\[ = [\lambda; A^{(1)}; \ldots; A^{(N)}] \]

Note that \([\lambda; A^{(1)}; \ldots; A^{(N)}]\) is shorthand notation to de-

scribe the CP decomposition, where \(\lambda\) is a vector of the weights \(\lambda_r\) and \(a_r^{(n)}\) is the rth column of \(A^{(n)}\). The CP ten-

sor 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 compo-

nents, a higher-order generalization of NMF. Non-negative tensor factorization (NTF) requires the elements of the fac-

tor 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 represen-

tations. However, additional sparsity may be desired for the factor matrices. Traditional sparsity-inducing penal-
ties such as \(\ell_1\) and \(\ell_2\) 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 non-

linearly transformed gradient descent approach [7]. Yet, the approach is computationally expensive because multiple ten-

sor factorization stages are required and sparsity constraints are achieved via an exponential update (i.e. \(s \leftarrow s^{1+\gamma}\), for small \(\gamma\)). Sparse factor representation using KL divergence is difficult because (1) an incorrect zero in the factor rep-

resentation causes the objective function to be ill-defined as \(\lim_{t \to 0} \log t = -\infty\), 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 factoriza-

tion 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.

3.1 Problem Formulation and Overview

Let \(X\) denote an observed tensor constructed from count data with size \(I_1 \times I_2 \times \ldots \times I_N\) and \(M\) represent a same-

sized tensor of Poisson parameters for \(X\). \(M\) is split into two terms, a rank one bias tensor \(C\) and a rank \(R\) interaction tensor \(\mathcal{V}\). The bias term, or baseline characteristics of the population, is composed of \(N\) positive vectors \(u^{(1)}, \ldots, u^{(N)}\) and a positive scalar \(\alpha\). 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 \(\lambda_r\). However, Marble constrains the feasible space of the vectors to either be zero or above some threshold value \(\gamma_n\). The optimization problem is defined as

\[
\min f(M) \equiv \sum_i (m_{i,\ldots,\ldots} - x_i, \log m_{i,\ldots,\ldots})
\]

subject to

\[
M = C + V
\]

\[
C = [\alpha; u^{(1)}; \ldots; u^{(N)}] \in \Omega_C
\]

\[
V = [\lambda; A^{(1)}; \ldots; A^{(N)}] \in \Omega_V
\]

\[
\Omega_C = \Omega_\alpha \times \Omega_{A1} \times \ldots \times \Omega_{AN}
\]

\[
\Omega_\alpha = (0, +\infty)
\]

\[
\Omega_{an} = \{u \in [0,1]^{I_i \times I_j} | ||u||_1 = 1\}
\]

\[
\Omega_V = \Omega_\lambda \times \Omega_{A1} \times \ldots \times \Omega_{AN}
\]

\[
\Omega_\lambda = (0, +\infty)^R
\]

\[
\Omega_{an} = \{A \in [0, \gamma_n, 1]^{I_i \times I_j} | ||A||_1 = 1 \forall r\}.
\]

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 3.4.2, 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.
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 $r$th rank one tensor, each non-zero element along the $n$th vector $a_{n(r)}$ 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 $n$th factor matrix, $A_{n}$, will have non-zero components that range from $\gamma_{n}$ 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, $\gamma_{n}$, 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 $x_{n}$. The bias tensor also provides computational stability for the “inadmissible zeros” resulting from the sparse factor representation. Under the assumption that $\gamma_{n}$ 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 $m_{r}$ is zero is high. If the corresponding observed tensor element is zero ($x_{r} = 0$), then $m_{r} = -\log m_{r} = 0$. However, for a non-zero $x_{r}$, the objective function is ill-defined as $\lim_{x_{r} \to 0} \log i = -\infty$, 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 $\alpha$ and the factor vectors $u^{(1)}$, $u^{(2)}$, …, $u^{(N)}$ 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 $V$ to $R + 1$. Each factor vector ($u^{(n)}$ 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 $\mathcal{M}$ and allows sparse factor matrices in the interaction tensor, $A^{(n)}$.

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 $n$th mode, we express the mode-$n$ matricization $V_{(n)} = B^{(n)} \Pi^{(n)}$, where

\[ B^{(n)} = A^{(n)} \Lambda, \quad \Lambda = \text{diag}(\Lambda) \]

\[ \Pi^{(n)} = \left( A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)} \right)^{T}. \]

The weights ($\Lambda$) are absorbed into the $n$th factor matrix $B^{(n)}$ in Equation (6) and we use the matrix $\Pi^{(n)}$ to denote the fixed parts in Equation (7). Note that the size of matrix $B^{(n)}$ is $I_{n} \times R$ while the size of matrix $\Pi^{(n)}$ 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:

\[ e^{T} \left[ C_{(n)} + B^{(n)} \Pi^{(n)} - X_{(n)} \ast \log \left( C_{(n)} + B^{(n)} \Pi^{(n)} \right) \right] e, \]

where $e$ is a vector of all ones, $C_{(n)}$ is the mode-$n$ matricization of the bias tensor and $X_{(n)}$ is the mode-$n$ matricization of the observed tensor.

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

\[ \frac{\partial f}{\partial b_{gh}} = \sum_{\ell} \pi_{gh \ell} - \sum_{\ell} x_{gL} \frac{\pi_{gh \ell}}{c_{gL} + \sum_{r} b_{gr} \pi_{r \ell}}. \]

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

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

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

\[ B^{(n)} = B^{(n)} \ast \left[ X \otimes (C_{(n)} + B^{(n)} \Pi^{(n)}) \right] \Pi^{(n)} \Pi^{(n)^{T}}. \]

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

\[ C_{(n)} = \alpha u^{(n)} \Psi^{(n)} \]

\[ \Psi^{(n)} = \left( u^{(N)} \otimes \cdots \otimes u^{(n+1)} \otimes u^{(n-1)} \otimes \cdots \otimes u^{(1)} \right)^{T}. \]
where the following conditions:

\[
\begin{align*}
\text{where } A_{(n)} &= \begin{bmatrix} X_{(n)} \otimes (\alpha u_{(n)}^{(n)} + V(n)) \end{bmatrix} \Psi_{11}^{(n)}^T.
\end{align*}
\]

3.4.2 Gradual Projection

The alternating minimization updates satisfy the non-negative constraints of the weights (\(\lambda\)), factor matrices \(A_{(n)}\), and basis vectors \(u_{(n)}\). 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 \(\xi\) is introduced to calculate the gradual projection, where each iteration uses the threshold \(\xi_{\gamma_n}\). The range of \(\xi\) 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 \(\xi\) as follows:

\[
\begin{align*}
\kappa^{(k)} &= 1 - \left\{ f(M_{(k-1)}) - f(M_{(k)}) \right\} \left\{ f(M_{(k-1)}) \right\}^{-1} = \text{max}(\xi(k), 1 - \frac{1}{2}\kappa^{(k)}) + \frac{1}{2}\kappa^{(k)}.
\end{align*}
\]

Marble uses a moving average of the \(\xi\) 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 \(A_{(n)}\) and the bias vector \(u_{(n)}\). 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:

\[
\begin{align*}
\Phi_{(n)} &= X_{(n)} \otimes (\alpha u_{(n)}\Psi_{(n)} + B_{(n)}\Pi_{(n)}) \Pi_{(n)}^T \Psi_{(n)}^{(n)}(12) \\
\Phi_{(n)} &= X_{(n)} \otimes (\alpha u_{(n)}^{(n)}\Psi_{(n)} + B_{(n)}^{(n)}\Pi_{(n)}^{(n)}) \Psi_{(n)}^{(n)}(13)
\end{align*}
\]

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:

\[
\begin{align*}
\min(A_{(n)}, E - \Phi_{(n)}) = 0 \\
\min(u_{(n)}, E - Z_{(n)}) = 0.
\end{align*}
\]

where \(E\) 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. \(\min(u_{(n)}, E - Z^{(n)}) \leq \text{kktTol}\). The detailed Marble algorithm is presented in Algorithm 2.

Sparse Implementation. When \(\mathcal{X}\) is a sparse tensor, we construct \(\mathbf{V}\) and \(\mathbf{C}\) using the non-zero elements of the observed tensor and avoid constructing \(\mathbf{V}\) and \(\mathbf{C}\) 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 \(\mathcal{X}\). We can store \(\mathcal{X}\) 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 \(\Pi\) and \(Z\) that correspond to each non-zero element of \(\mathcal{X}\). The qth vector of \(\Pi\) and \(Z\) are:

\[
\begin{align*}
u^{(q)} &= a^{(1)}(q) \cdots a^{(n+1)}(q) \cdots a^{(n-1)}(q) \cdots a^{(N)}(q) \\
y^{(q)} &= u^{(1)}(q) \cdots u^{(n+1)}(q) \cdots u^{(n-1)}(q) \cdots u^{(N)}(q).
\end{align*}
\]
where $a^{(1)}_{i, j}$ is the $i$th row of $A^{(1)}$. Thus, we can calculate element $(i, r, \tau) = v_i j$ of $\Phi$ and $Z$ as:

$$v_i j = \frac{x_i (r)}{\langle \epsilon_j y_i (r), v_i u_i (r) \rangle} + w_i j, B_i (r)$$

$$\Phi = \sum q_i j = v_i j, w_i j$$

$$z_i j = \sum q_i j = v_i j, y_i (r)$$

Note that we require storage for the augmented tensor, which entails storing $y_i (r)$. 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 $\Phi$ requires on the order of the decomposition rank $\tau$ 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 $O(D^3)$. 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 $V$ is then used to define $R$ candidate phenotypes, similar to Figure 1. Thus, the $r$th candidate phenotype is defined using the non-zero elements of the $r$th 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 $r$th element denotes the probability the patient exhibits characteristics consist with the $r$th phenotype. For notation purpose, we will assume the patient mode is the first mode of the tensor. Thus given a new patient’s tensor $X$, we want to find $\hat{X}$ and $\hat{A}^{(1)}$ that provides that best approximates the new patient’s tensor. Our projection also needs to determine $\hat{\alpha}$, the strength of the bias in $\hat{\lambda}$. 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 $\hat{B}^{(1)}$ across all $R$ phenotypes instead of the standard column normalization for each phenotype, (2) setting the augmented vector $(\hat{a}^{(1)}_\alpha)$ to 1 and absorbing the weight into $\hat{\alpha}$, and (3) ignoring the projection onto the feasible set defined in the optimization problem.

### 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.

1. **Non-zero Ratio**
   \[
   \text{Non-zero Ratio} = \frac{\# \text{Non-zero in Computed Solution}}{\# \text{Non-zero in Actual Solution}}
   \]

2. **Similarity**
   \[
   \text{Similarity}(a, b) = \frac{a \cdot b}{||a|| ||b||}
   \]

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 $100 \times 80 \times 60$ with rank of 10 ($R = 10$). We generate the model $\mathbf{M} = \mathbf{C} + \mathbf{V}$, where $\mathbf{C} = [\alpha; u^{(1)}; \ldots; u^{(N)}]$ and $\mathbf{V} = [\lambda; A^{(1)}; \ldots; A^{(N)}]$. Each factor matrix $A^{(n)}$ 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 $\lambda$. Each augmented vector is chosen in a similar fashion except the weight is set to $\alpha = 2$. The full tensor $\mathbf{M}$ is calculated from the factor matrices and the augmented vectors. Then each tensor element $x_{ijk}$ is sampled from the Poisson distribution.
with the parameter set to $m_{ij,k}$. Ten observation tensors are generated from the tensor $\mathcal{M}$.

### 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 $2 \times$ 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 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. 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 finer-grained level. Thus, similar diagnosis codes (procedure codes) were grouped using the Unified Medical Language System to aggregate the individual ICD-9 codes and HCPCS codes to higher level medical hierarchies. Therefore,
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 $\gamma_n$. 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 occurs at the start of the second bin. Thus, for the remainder of the paper, the thresholds used are $\gamma = [0.0001, 0.01, 0.01]$.

4.2.2 Predictive performance

The phenotypes are evaluated on a classification task of predicting high cost (above 75\% of medical care) on the treatment of chronic diseases such as heart failure, diabetes, and cancer can help medical professionals tailor treatment options based on the patient’s 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\(^6\).

4.2.3 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. Patients’s chronic condition flag cannot be perfectly reproduced due to the synthetic claim process used.

\(^{6}\)The latest disease reports are located at http://www.cdc.gov/chronicdisease/resources/publications/saag.htm

\(^{7}\)A patient’s chronic condition flag cannot be perfectly reproduced due to the synthetic claim process used.
patients 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

Table 4: Two heart failure related phenotypes

<table>
<thead>
<tr>
<th>Other forms of heart disease</th>
<th>Complications of surgical and medical care</th>
<th>Symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiovascular Procedures</td>
<td>Hematology and Coagulation Procedures</td>
<td>Evaluation and Management of Other Outpatient Services</td>
</tr>
<tr>
<td>Surgery Procedures on the Cardiovascular System</td>
<td>Chemistry Pathology and Laboratory Tests</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Two chronic disease related phenotypes

<table>
<thead>
<tr>
<th>Diabetes Phenotype</th>
<th>Arthritis Phenotype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diseases of other endocrine glands</td>
<td>Arthropathies and related disorders</td>
</tr>
<tr>
<td>Complications of surgical and medical care</td>
<td>Physical Medicine and Rehabilitation Procedures</td>
</tr>
<tr>
<td>Symptoms</td>
<td>Evaluation and Management of Other Outpatient Services</td>
</tr>
<tr>
<td>Chemistry Pathology and Laboratory Tests</td>
<td>Surgical Procedures on the Musculoskeletal System</td>
</tr>
<tr>
<td>Organ or Disease Oriented Panels</td>
<td>Diagnostic Radiology Procedures</td>
</tr>
<tr>
<td>Hematology and Coagulation Procedures</td>
<td></td>
</tr>
<tr>
<td>Surgical Procedures on the Cardiovascular System</td>
<td></td>
</tr>
</tbody>
</table>


