eTREE: Learning Tree-structured Embeddings

Faisal M. Almutairi[1,2], Yunlong Wang[2], Dong Wang[2], Emily Zhao[2], and Nicholas D. Sidiropoulos[3]

[1]Department of ECE, University of Minnesota, MN, USA
[2]Advanced Analytics, IQVIA Inc., Plymouth Meeting, PA, USA
[3]Department of ECE, University of Virginia, VA, USA
almut012@umn.edu, {yunlong.wang, dong.wang, emily.zhao}@iqvia.com, nikos@virginia.edu

Abstract

Matrix factorization (MF) plays an important role in a wide range of machine learning and data mining models. MF is commonly used to obtain item embeddings and feature representations due to its ability to capture correlations and higher-order statistical dependencies across dimensions. In many applications, the categories of items exhibit a hierarchical tree structure. For instance, human diseases can be divided into coarse categories, e.g., bacterial, and viral. These categories can be further divided into finer categories, e.g., viral infections can be respiratory, gastrointestinal, and exanthematous viral diseases. In e-commerce, products, movies, books, etc., are grouped into hierarchical categories, e.g., clothing items are divided by gender, then by type (formal, casual, etc.). While the tree structure and the categories of the different items may be known in some applications, they have to be learned together with the embeddings in many others. In this work, we propose eTREE, a model that incorporates the (usually ignored) tree structure to enhance the quality of the embeddings. We leverage the special uniqueness properties of Nonnegative MF (NMF) to prove identifiability of eTREE. The proposed model not only exploits the tree structure prior, but also learns the hierarchical clustering in an unsupervised data-driven fashion. We derive an efficient algorithmic solution and a scalable implementation of eTREE that exploits parallel computing, computation caching, and warm start strategies. We showcase the effectiveness of eTREE on real data from various application domains: healthcare, recommender systems, and education. We also demonstrate the meaningfulness of the tree obtained from eTREE by means of domain experts interpretation.

Introduction

Matrix Factorization (MF) plays an important role in a wide range of machine learning models, for various applications such as dimensional reduction and embedding. A popular task is matrix completion, where the goal is to infer the unknown/missing matrix entries from the observed ones. A common approach is to employ MF to find a reduced-dimension representation (embedding) of each element corresponding to the matrix dimensions (e.g., users and items) These embeddings capture the essential information due to the ability of MF to capture correlations and higher-order
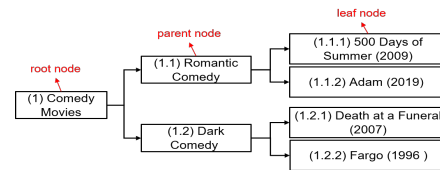
Figure 1: An example of hierarchical movie categories.

statistical dependencies across dimensions. The entry corresponding to the ith user and j^{th} item can be inferred by the inner product of their embeddings. Matrix completion finds a wide range of applications including collaborative filtering in recommender systems (Koren 2008), disease and treatment prediction and patient subtyping (Wang et al. 2019) in healthcare analytics, student performance prediction and course recommendation in learning analytics (Almutairi, Sidiropoulos, and Karypis 2017), and image processing (Liu et al. 2015).

Incorporating side contextual information or priors, e.g., sparsity (Hoyer 2004), smoothness, and latent clustering (Yang, Fu, and Sidiropoulos 2016), is well-motivated in matrix factorization and completion of sparse data. This is because a major challenge stems from the fact that we aim to find latent representations from very few samples. In this paper, we present a principled approach that incorporates the *unknown* implicit tree structure prior. In many applications, categories of items display a hierarchical tree structure. In higher education, for instance, courses form multiple trees via their prerequisite hierarchy. Movie genres, e.g., comedy, action, and fantasy, comprise different fine subcategories as illustrated in the example in Fig. 1. Another example appears in Electronics Health Records (EHR) in healthcare analytics, where medical service (diagnoses, procedures, and prescription) can be clustered into subcategories, and these subcategories can also be grouped into coarse categories (examples are provided in the experimental results in Fig. 3). Individuals, e.g., users, students, and patients, also exhibit hierarchical clusters where the common traits between people increase as we move down from the root nodes to the leaf nodes in a tree (Maleszka, Mianowska, and Nguyen 2013; Wang, Pan, and Xu 2014). In many applications, the categorical hierarchy is either unknown, or requires manual labeling of massive amounts of data.

Incorporating tree structures in machine learning models has been recently considered, mostly in recommender sys-

tems (Nikolakopoulos, Kouneli, and Garofalakis 2015; Li et al. 2019; Zhong, Fan, and Yang 2012) and also in other applications such as image processing (Fan and Cheng 2018), clustering and classification (Trigeorgis et al. 2016), and natural language processing (NLP) (Shen et al. 2018). For example, the recommender system model in (Yang et al. 2016) penalizes MF with the distance between users who share common traits based on hierarchically-organized features. In another MF model (Sun et al. 2017), the item embeddings are assumed to form a tree, where each leaf node represents a single item and the parent nodes contain subsets of items (categories). The embeddings of parent nodes and leaf nodes are learned jointly. The final item feature vector is modeled as a weighted sum of its embedding and those of the categories it belongs to. Regularizing MF with a pre-defined tree prior has been also explored in response prediction in online advertising (Menon et al. 2011). In this example, the tree groups the set of ads according to their campaigns, and the campaigns are further grouped based on the advertisers running them.

All the aforementioned methods assume that the tree structure is known apriori, or learned separately via side information. Recently, (Wang et al. 2018, 2015) proposed to capture the unknown implicit tree structure via a model based on nonnegative matrix factorization (NMF). In a three-layer tree, the embedding of a leaf node (item/user) is assumed to be a linear combination of *all* the parent nodes (subcategories) in the intermediate layer, and each subcategory is a linear combination of all the categories in the root nodes. The weights that determine the memberships of a child node to the parent nodes are non-negative and learned by the model. This results in a fully connected tree, thus, a clear tree clustering can not be obtained. Moreover, (Wang et al. 2018, 2015) imposes the implicit tree as a hard constraint, which can be restrictive if the data do not exactly follow the imposed prior.

In this paper, we propose eTREE (Learning Tree-structured Embeddings), a framework that integrates the unknown implicit tree structure into a low-rank nonnegative factorization model to improve the quality of embeddings. eTREE does not require any extra information and jointly learns: i) the embeddings of all the tree nodes (items, subcategories, and main categories), and ii) the tree clustering in an unsupervised fashion. Unlike (Wang et al. 2015, 2018), the obtained tree provides clear hierarchical clusters as each node belongs to exactly one parent node, e.g., an item belongs to one subcategory, and a subcategory belongs to one main category. The formulation of eTREE handles partially observed data matrices, which appear often in real-world applications. We derive an efficient algorithm to compute eTREE with a scalable implementation that leverages parallel computing, computation caching, and warm-start strategies. Our contributions can be summarized as follows:

1. Formulation: eTREE provides an intuitive formulation that: i) exploits the tree structure, and ii) learns the hierarchical clustering in an unsupervised data-driven fashion.
2. Identifiability: We leverage the special uniqueness properties of NMF to prove identifiability of eTREE.
3. Effectiveness: eTREE significantly improves the quality

of the embeddings in terms of matrix completion error on data from recommender systems, healthcare, and education.
4. Interpretability: We demonstrate the meaningfulness of the tree clusters learned by eTREE using real-data interpreted by domain experts.

Background

In this section, we provide the background needed before presenting eTREE. We review NMF and related recent identifiability results, followed by a brief background on the alternating direction method of multipliers (ADMM).

Notation: x. $\mathbf{x}$, $\mathbf{X}$ denote scalars, vectors, and matrices, respectively; $\mathbf{X}(i,:)$ ($\mathbf{X}(:,j)$) refers to the ith row (j^{th} column) of $\mathbf{X}$; $\mathbf{X}(\mathcal{J},:)$ denotes the rows of $\mathbf{X}$ in the set $\mathcal{J}$. The product $\odot$ is the Hadamard (element-wise) product.

Non-negative Matrix Factorization

Assume we have a healthcare data matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ indexed by (patient, medical service), where $\mathbf{X}(i,j)$ denotes the number of times the ith patient has received the jth service. In other applications, $\mathbf{X}$ may contain the ratings given by users to items, or the grades received by students in their courses. In some parts of this paper, we refer to patients, users, students, etc. as individuals, and to medical services, products, etc. as items. NMF models aim to decompose the data matrix into low-rank latent factor matrices as $\mathbf{X} = \mathbf{A}\mathbf{B}^T$, where $\mathbf{A} \in \mathbb{R}^{N \times R}$, $\mathbf{B}^{M \times R}$ only have non-negative values, and $R \leq \min(N, M)$ is the matrix rank. NMF has gained considerably special attention as it tends to produce interpretable representations. For instance, it has been shown that the columns of $\mathbf{A}$ produce clear parts of human faces (e.g., nose, ears, and eyes) when NMF in applied on a matrix $\mathbf{X}$ whose columns are vectorized face images (Lee and Seung 1999). In practice, NMF is often formulated as a bilinear optimization problem:

$$\min_{\mathbf{A} \geq \mathbf{0}, \mathbf{B} \geq \mathbf{0}} \quad \frac{1}{2}\mathcal{F}(\mathbf{A}, \mathbf{B}) = \|\mathbf{W} \odot (\mathbf{X} - \mathbf{A}\mathbf{B}^T)\|_F^2 \qquad (1)$$

where $\mathbf{W} \in \{0,1\}^{N \times M}$ has ones at the indices of the observed entries in $\mathbf{X}$, and zeros otherwise. Each row of $\mathbf{A}$ corresponds to the embedding/latent representation of the corresponding individual, whereas the rows of $\mathbf{B}$ are the embeddings of the items.

Identifiability of NMF: The interpretability of NMF is intimately related to its uniqueness properties – the latent factors are identifiable under some conditions (up to trivial ambiguity, e.g., scaling/counter-scaling or permutation). To facilitate our discussion of the uniqueness of eTREE, we present the following definitions and established identifiability results.

Definition 1 (Identifiability) *The NMF of* $\mathbf{X} = \mathbf{A}\mathbf{B}^T$ *is said to be (essentially) unique if* $\mathbf{X} = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}^T$ *implies* $\widetilde{\mathbf{A}} = \mathbf{A}\mathbf{\Pi}\mathbf{D}$ *and* $\widetilde{\mathbf{B}} = \mathbf{B}(\mathbf{\Pi}\mathbf{D})^{-1}$*, where* $\mathbf{\Pi}$ *is a permutation matrix, and* $\mathbf{D}$ *is a diagonal positive matrix.*

Definition 2 (Sufficiently Scattered) *A nonnegative matrix* $\mathbf{B} \in \mathbb{R}^{M \times R}$ *is said to be sufficiently scattered if: 1)* $cone\{\mathbf{B}^T\} \supseteq \mathcal{C}$*, and 2)* $cone\{\mathbf{B}^T\} \cap bd\{\mathcal{C}^\star\} = \{\lambda\mathbf{e}_k | \lambda \geq 0, k = 1, \ldots, R\}$*, where* $\mathcal{C} = \{\mathbf{x} | \mathbf{x}^T\mathbf{1} \geq \sqrt{R-1}\|\mathbf{x}\|_2\}$*,*

$\mathcal{C}^\star = \{\mathbf{x}|\mathbf{x}^T\mathbf{1} \geq \|\mathbf{x}\|_2\}$, $cone\{\mathbf{B}^T\} = \{\mathbf{x}|\mathbf{x} = \mathbf{B}^T\boldsymbol{\theta}, \forall\boldsymbol{\theta} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\theta} = 1\}$, and $cone\{\mathbf{B}^T\}^\star = \{\mathbf{y}|\mathbf{x} = \mathbf{x}^T\mathbf{y}, \forall\mathbf{x} \in cone\{\mathbf{H}^T\}\}$ are the conic hull of $\mathbf{B}^T$ and its dual cone, respectively, and bd is the boundary of a set.

The works in (Fu et al. 2015; Lin et al. 2015) prove that the so-called volume minimization (VolMin) criterion can identify the factor matrices if $\mathbf{A}$ is full-column rank, and the rows of $\mathbf{B}$ are sufficiently scattered (Definition 2) and sum-to-one (row stochastic). Recently, Fu *et al.* shifted the row stochastic condition on rows of $\mathbf{B}$ to the *columns* of $\mathbf{B}$.

Theorem 1 (NMF Identifiability) *(Fu, Huang, and Sidiropoulos 2018)* $\mathbf{A}$ *and* $\mathbf{B}$ *are essentially unique under the criterion of minimizing* $det(\mathbf{A}^T\mathbf{A})$ *w.r.t.* $\mathbf{A} \in \mathbb{R}^{N \times R}$ *and* $\mathbf{B} \in \mathbb{R}^{M \times R}$, *subject to* $\mathbf{X} = \mathbf{A}\mathbf{B}^T$ *and* $\mathbf{B}^T\mathbf{1} = \mathbf{1}, \mathbf{B} \geq \mathbf{0}$ *if* $\mathbf{B}$ *is sufficiently scattered, and rank($\mathbf{X}$) = rank($\mathbf{A}$) = R.*

Theorem 1 provides an intriguing generalization of NMF, as it pertains to a more general factorization. Note that $\mathbf{A}$ is *not* restricted to be non-negative. Also note that the column-sum-to-one constraint on $\mathbf{B}$ is without loss of generality, as one can always assume the columns of $\mathbf{B}$ are scaled by a diagonal matrix $\mathbf{D}$, and compensate for this scaling in the columns of $\mathbf{A}$, i.e., $\mathbf{X} = (\mathbf{A}\mathbf{D}^{-1})(\mathbf{B}\mathbf{D})^T$.

Alternating Direction Method of Multipliers

ADMM is a primal-dual algorithm that solves convex optimization problems in the form

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c} \end{aligned} \quad (2)$$

by iterating the following updates

$$\mathbf{x} \leftarrow \arg\min_{\mathbf{x}} \quad f(\mathbf{x}) + \rho/2\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}\|_2^2$$

$$\mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \quad g(\mathbf{z}) + \rho/2\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}\|_2^2 \quad (3)$$

$$\mathbf{u} \leftarrow \mathbf{u} + (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c})$$

where $\mathbf{u}$ is a scaled version of the dual variable, and $\rho > 0$ is a Lagrangian parameter. A comprehensive review of ADMM can be found in (Boyd, Parikh, and Chu 2011).

Proposed Framework: eTREE

In this section, we present our proposed framework. We start with the mathematical formulation of eTREE, then we discuss the theoretical uniqueness of the proposed model. Next, we work out some design considerations of eTREE, then we derive the algorithmic solution.

eTREE: Formulation

In many applications, categories of items exhibit a hierarchical tree structure – as we showed in the introduction. For ease of notation, let us denote the embedding matrix of items resulting from NMF in (1) as $\mathbf{B}_1 \in \mathbb{R}^{M_1 \times R}$, where M_1 is the number of items. Assume that the embeddings of the M_1 items (rows of $\mathbf{B}_1$) are the leaf nodes at the very bottom layer in a tree. A subset of items that belong to the same category is grouped together via one parent node, where the parent node is the embedding of the corresponding category. Assuming that the embeddings are fully inherited (replicated verbatim) from one's parent category, we can further decompose $\mathbf{B}_1$ into

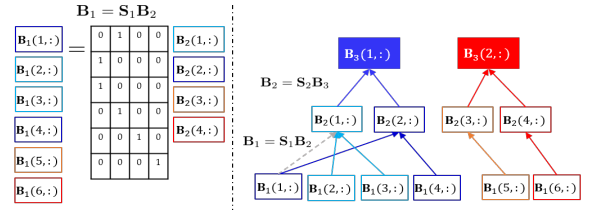$$\mathbf{B}_1 = \mathbf{S}_1\mathbf{B}_2 \quad (4)$$

Figure 2: Illustration of the tree prior in eTREE.

where each row of $\mathbf{B}_2 \in \mathbb{R}^{M_2 \times R}$ is the embedding of one category, M_2 is the number of categories with $M_2 \leq M_1$, and $\mathbf{S}_1 \in \{0,1\}^{M_1 \times M_2}$, $\|\mathbf{S}_1(i,:)\|_0 = 1, \forall i \in [M_1]$, i.e., values in $\mathbf{S}_1$ are binary with only one 1 per row to ensure that each item belongs to exactly one category. Note that M_2 is the number of parent nodes (categories) in the second from bottom layer. The M_2 categories can be grouped into coarser categories, i.e., we decompose $\mathbf{B}_2$ into $\mathbf{B}_2 = \mathbf{S}_2\mathbf{B}_3$, where rows of $\mathbf{B}_3 \in \mathbb{R}^{M_2 \times M_3}$ represent the embeddings of the coarse categories, and $\mathbf{S}_2$ maps the M_2 fine-level categories into the M_3 coarse-level categories in the same fashion as $\mathbf{S}_1$. Up to here, we have constructed a three-layer tree, and we can use the same concept to create a Q-layer tree. Fig. 2 illustrates the mapping between $\mathbf{B}_1$ and $\mathbf{B}_2$ in matrix notation (left), and shows a 3-layer tree (right). Generalizing to Q layers, we obtain

$$\mathbf{B}_1 = \mathbf{S}_1\mathbf{S}_2\ldots\mathbf{S}_{Q-1}\mathbf{B}_Q \quad (5)$$

Substituting the embedding matrix of items in (1) with the right term in (5) above may seem natural, however there is a solution ambiguity in the cases where $Q > 2$. To see this, the route $\mathbf{B}_1(1,:) \rightarrow \mathbf{B}_2(2,:) \rightarrow \mathbf{B}_3(1,:)$ in Fig. 2 (right) would give the same cost value as $\mathbf{B}_1(1,:) \rightarrow \mathbf{B}_2(1,:) \rightarrow \mathbf{B}_3(1,:)$ (the dotted gray arrow). Moreover, imposing the tree structure as a hard constraint can be too intrusive when the data do not exactly follow the assumed prior. Thus, we propose to: i) incorporate the tree prior as a soft constraint, and ii) explicitly solve for the embedding of the intermediate layers to resolve the (immaterial for our purposes) solution ambiguity. This yields the following formulation:

$$\begin{aligned} \min_{\mathcal{U}} \quad & \mathcal{F}(\mathbf{A}, \mathbf{B}_1) + \frac{\mu}{2}\sum_{q=1}^{Q-1}\|\mathbf{B}_q - \mathbf{S}_q\mathbf{B}_{q+1}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}_q \in \{0,1\}^{M_q \times M_{q+1}}, q \in [Q-1] \\ & \|\mathbf{S}_q(i,:)\|_0 = 1, \forall i \in [M_q], q \in [Q-1] \\ & \mathbf{A} \geq \mathbf{0}, \mathbf{B}_1 \geq \mathbf{0} \end{aligned} \quad (6)$$

where $\mathcal{U} := \{\mathbf{A}, \{\mathbf{B}_q\}_{q=1}^Q, \{\mathbf{S}_q\}_{q=1}^{Q-1}\}$ is the set of all variables, and $\mathcal{F}$ is the NMF cost function as defined in (1) with $\mathbf{B}_1$ as the embedding matrix of items. The second term is to minimize the difference between the embedding of each child node and its parent node in the tree structure. In other words, it minimizes the difference between the embeddings of each item and its category, or between each fine category and its coarse category. $\mu \geq 0$ is a regularization parameter to balance the data fidelity and the tree prior.

There is an intriguing connection between the proposed tree regularizer and k-means formulation. The variables $\{\mathbf{S}_q\}_{q=1}^{Q-1}$ are equivalent to the assignment variables in k-means for clustering the rows of $\{\mathbf{B}_q\}_{q=1}^{Q-1}$, respectively, and

$\mathbf{B}_Q$ is equivalent to the centroid variable in k-means for clustering the rows of $\mathbf{B}_{Q-1}$. On the other hand, each variable in $\{\mathbf{B}_q\}_{q=2}^{Q-1}$ is involved in two terms: 1) $\|\mathbf{B}_{q-1} - \mathbf{S}_{q-1}\mathbf{B}_q\|_F^2$ where its rows are centroids, and 2) $\|\mathbf{B}_q - \mathbf{S}_q\mathbf{B}_{q+1}\|_F^2$ where its rows are the points to be clustered. This can be thought of as a *regularized* k-means. Interestingly, the joint NMF and latent k-means model in (Yang, Fu, and Sidiropoulos 2016) is a special case of eTREE with $Q = 2$.

Note that if the tree structure is known apriori, it can be seamlessly incorporated using our formulation. A direct way is to fix $\{\mathbf{S}_q\}_{q=1}^{Q-1}$ to the known tree and solve (6) w.r.t. the rest of the variables. This way we learn the embedding of the categories and penalize the distance between items and their corresponding categories. When the tree is *partially* known, one can fix the known parts and learn the unknowns. Another way to integrate a known tree is to penalize the difference between the embeddings of items that share similar paths to the root nodes. The latter method does not require us to learn $\{\mathbf{B}_q\}_{q=2}^{Q}$. In this work, we focus on the more challenging scenario where the tree structure is unknown and to be learned from the data.

eTREE has the following advantages: i) it incorporates the tree structure to improve the quality of embeddings, ii) unlike most methods in the literature, it assumes the tree is unknown and learns it through the solution of $\{\mathbf{S}_q\}_{q=1}^{Q-1}$ in an unsupervised fashion, and iii) it provides the embedding of the parent nodes (categories) in addition to the item embeddings. The tree clustering can be useful in broader applications such as classification and data labeling. The embeddings of categories provide extra information for some applications, e.g., web personalization and category-based recommendation (He, Li, and Liao 2017).

eTREE: Theoretical Identifiability

Identifiability in machine learning problems that require parameter estimation is essential in guaranteeing sensible results, especially in applications where model identifiability is entangled with interpretability, such as topic modeling (Arora et al. 2013), image processing (Lee and Seung 1999), and social network clustering (Mao, Sarkar, and Chakrabarti 2017). Nevertheless, the majority of MF-based methods in practice do not have known identifiability guarantees. In the following theorem, we establish the identifiability of eTREE for the case where $\mathbf{X}$ is fully observed.

Theorem 2 *Assume that a data matrix follows* $\mathbf{X} = \mathbf{AB}_1^T$, *where* $\mathbf{A} \in \mathbb{R}^{N \times R}$, *and* $\mathbf{B}_1 \in \mathbb{R}^{M_1 \times R}$ *are the ground-truth factors, and assume that* $\mathbf{B}_1 = \mathbf{S}_1\mathbf{S}_2\ldots\mathbf{S}_{Q-1}\mathbf{B}_Q$, *where* $\mathbf{S}_q \in \{0,1\}^{M_q \times M_{q+1}}$, $\|\mathbf{S}_q(i,:)\|_0 = 1, \forall i \in [M_q], q \in [Q-1]$. *Let* $\mathbf{S} = \mathbf{S}_1\mathbf{S}_2\ldots\mathbf{S}_{Q-1}$, *then,* $\mathbf{B}_1 = \mathbf{SB}_Q$. *Also, assume that* $rank(\mathbf{X}) = rank(\mathbf{A}) = R$, *and, without loss of generality,* $\mathbf{M}_Q \geq R$. *If* $\mathbf{A}$ *and* $\mathbf{S}$ *are full-column rank, and rows of* $\mathbf{B}_Q$ *are sufficiently scattered, then rows of* $\mathbf{B}_1$ *are sufficiently scattered, and* $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_Q$, *and* $\mathbf{S}$ *are essentially unique.*

Proof Sketch: The factors $\mathbf{A}$ and $\mathbf{B}_Q$ in $\mathbf{M} = \mathbf{AB}_Q^T$ are essentially unique by Theorem 1, since $\mathbf{A}$ is full-column rank, and rows of $\mathbf{B}_Q$ are sufficiently scattered (Definition 2). If $\mathbf{S}$ is full-column rank, then all the rows of $\mathbf{B}_Q$ will appear in

$\mathbf{B}_1 = \mathbf{SB}_Q$. Thus, rows of $\mathbf{B}_1$ are sufficiently scattered *iff* the rows of $\mathbf{B}_Q$ are sufficiently scattered. Now, $\mathbf{A}$ and $\mathbf{B}_1$ in $\mathbf{X} = \mathbf{AB}_1^T$ are essentially unique by Theorem 1, since $\mathbf{A}$ is full-column rank, and rows of $\mathbf{B}_1$ are sufficiently scattered. Next, the factor $\mathbf{S}$ in $\mathbf{B}_1 = \mathbf{SB}_Q$ is also essentially unique because the rows of $\mathbf{B}_1$ are rows of $\mathbf{B}_Q$, and every row of $\mathbf{B}_Q$ appears in $\mathbf{B}_1$, hence $\mathbf{S}$ can be determined based on the correspondence (identifiability of $\mathbf{S}$ also follows as a very special instance of Theorem 1).

In plain words, in addition to the NMF identifiability conditions ($\mathbf{A}$ to be full-rank, and rows of $\mathbf{B}_Q$ to be sufficiently scattered), we only require $\mathbf{S}$ to be full-column rank, which means that every root node (main category) must have at least one leaf node — this is a natural condition in a tree. Interestingly, the rows of $\mathbf{B}_Q$ are likely to be sufficiently scattered as they are the embeddings of the coarsest categories and encouraged to be distant (think, e.g., in a 3-layer tree, each row is the centroid of multiple subcategories, where each subcategory is the centroid of a set of items). We point out that there is an inherent column permutation ambiguity in $\{\mathbf{S}_q\}_{q=1}^{Q-1}$ in $\mathbf{S} = \mathbf{S}_1\mathbf{S}_2\ldots\mathbf{S}_{Q-1}$, however, this is immaterial in our context.

eTREE: Model Engineering

In this section, we discuss some caveats that need to be addressed in the formulation (6) before moving to the algorithmic derivation.

The first point is the scaling between the low-rank factors $\mathbf{A}$ and $\mathbf{B}_1$. The tree structure regularizer implicitly favors $\mathbf{B}_1$ to have a small norm. On the other hand, the first term is not affected by the scaling of $\mathbf{B}_1$, as long as this scaling is compensated for in $\mathbf{A}$. This motivates introducing norm regularization on $\mathbf{A}$, i.e., $\|\mathbf{A}\|_F$.

The second consideration is regarding the tree structure term. It has been shown that the cosine similarity metric is superior over the Euclidean distance in clustering (Strehl, Ghosh, and Mooney 2000) and latent clustering (Yang, Fu, and Sidiropoulos 2016) in many applications. We also observed that constraining the rows of $\{\mathbf{B}\}_{q=1}^{Q-1}$ to be in the unit l_2-norm ball, which is equivalent to using cosine similarity in clustering, gives better performance. Taking these points into account, we obtain the following formulation:

$$\min_{\mathcal{Y}} \; \mathcal{F}_d(\mathbf{A}, \mathbf{B}, \mathbf{D}) + \frac{\mu}{2}\sum_{q=1}^{Q-1}\|\mathbf{B}_q - \mathbf{S}_q\mathbf{B}_{q+1}\|_F^2 + \frac{\lambda}{2}\|\mathbf{A}\|_F^2$$

$$\begin{aligned}
\text{s.t.} \quad & \|\mathbf{B}_q(i,:)\|_2 = 1 \; \forall i \in [M_q], \; \forall q \in [Q-1] \\
& \mathbf{S}_q \in \{0,1\}^{M_q \times M_{q+1}}, \forall q \in [Q-1] \\
& \|\mathbf{S}_q(i,:)\|_0 = 1, \; \forall i \in [M_q], q \in [Q-1] \\
& \mathbf{D} = \text{Diag}(d_1, \ldots, d_{M_1}) \\
& \mathbf{A} \geq 0, \; \mathbf{B}_1 \geq 0
\end{aligned}$$

(7)

where $\mathcal{Y} := \{\mathbf{A}, \mathbf{D}, \{\mathbf{B}_q\}_{q=1}^{Q}, \{\mathbf{S}_q\}_{q=1}^{Q-1}\}$ is the set of all variables, $\lambda \geq 0$, $\mathcal{F}_d := 1/2\|\mathbf{W} \odot (\mathbf{X} - \mathbf{AB}^T\mathbf{D})\|_F^2$, and $\mathbf{D}$ is a diagonal matrix that is introduced to allow us to fix the rows of $\mathbf{B}_1$ onto the unit l_2-norm ball without loss of generality of the factorization model.

eTREE: Algorithm

The optimization problem in (7) is NP-hard (as it contains both NMF and k-means as special cases, and both are known to be NP-hard). We therefore present a carefully designed alternating optimization (AO) algorithm. The proposed algorithm leverages ADMM (reviewed in the background section above) and utilizes parallel computing, computation caching, and warm-start to provide a scalable implementation. The high level algorithmic strategy is to employ AO to update $\mathbf{A}$, $\mathbf{D}$, $\{\mathbf{B}_q\}_{q=1}^Q$, and $\{\mathbf{S}_q\}_{q=1}^{Q-1}$ one at a time, while fixing the others. The resulting sub-problems w.r.t. a single variable can be solved optimally.

We propose a variable-splitting strategy by introducing slack variables $\{\mathbf{Z}_q \in \mathbb{R}^{M_q \times R}\}_{q=1}^{Q-1}$ to handle the unit l_2 norm ball constraints on $\{\mathbf{B}_q \in \mathbb{R}^{M_q \times R}\}_{q=1}^{Q-1}$ in (7). Specifically, we consider the following optimization surrogate:

$$
\begin{aligned}
\min_{\mathcal{H}} \quad & \mathcal{F}_d(\mathbf{A}, \mathbf{B}, \mathbf{D}) + \frac{\mu}{2} \sum_{q=1}^{Q-1} \|\mathbf{B}_q - \mathbf{S}_q \mathbf{B}_{q+1}\|_F^2 \\
& + \frac{\eta}{2} \sum_{q=1}^{Q-1} \|\mathbf{B}_q - \mathbf{Z}_q\|_F^2 + \frac{\lambda}{2} \|\mathbf{A}\|_F^2 \\
\text{s.t.} \quad & \|\mathbf{Z}_q(i,:)\|_2 = 1, \forall i \in [M_q], q \in [Q-1] \\
& \mathbf{S}_q \in \{0,1\}^{M_q \times M_{q+1}}, \forall q \in [Q-1] \\
& \|\mathbf{S}_q(i,:)\|_0 = 1, \ \forall i \in [M_q], q \in [Q-1] \\
& \mathbf{D} = \mathrm{Diag}(d_1, \ldots, d_{M_1}) \\
& \mathbf{A} \geq 0, \ \mathbf{B}_1 \geq 0
\end{aligned} \tag{8}
$$

where $\mathcal{H} := \{\mathbf{A}, \mathbf{D}, \{\mathbf{B}_q\}_{q=1}^Q, \{\mathbf{Z}_q\}_{q=1}^{Q-1}, \{\mathbf{S}_q\}_{q=1}^{Q-1}\}$ is the set of all the variables, and $\eta \geq 0$. Note that when $\eta = +\infty$, then (8) is equivalent to (7). In practice, we choose a large η to enforce $\mathbf{B}_q \approx \mathbf{Z}_q$ (we set it to $\eta = 1000$ in all experiments). We handle problem (8) as follows. First, we update $\mathbf{A}$ by solving the following non-negative least squares

$$
\min_{\mathbf{A} \geq \mathbf{0}} \quad \frac{1}{2} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{A} \mathbf{B}_1^T \mathbf{D})\|_F^2 + \frac{\lambda}{2} \|\mathbf{A}\|_F^2 \tag{9}
$$

using ADMM. Due to space limitation, we use the update of $\mathbf{A}$ as a working example for the two updates that uses ADMM ($\mathbf{A}$ and $\mathbf{B}_1$). Problem (9) can be reformulated by introducing an auxiliary variable $\widetilde{\mathbf{A}}$

$$
\begin{aligned}
\min_{\mathbf{A}, \widetilde{\mathbf{A}}} \quad & \frac{1}{2} \|\mathbf{W}^T \odot (\mathbf{X}^T - \widetilde{\mathbf{B}}\widetilde{\mathbf{A}})\|_F^2 + \frac{\lambda}{2} \|\widetilde{\mathbf{A}}\|_F^2 + \mathcal{R}(\mathbf{A}) \\
\text{s.t.} \quad & \mathbf{A} = \widetilde{\mathbf{A}}^T
\end{aligned} \tag{10}
$$

where $\widetilde{\mathbf{B}} = \mathbf{D}\mathbf{B}_1$, and $\mathcal{R}(.)$ is the indicator function of the nonnegative orthant. Next, we derive the ADMM updates

$$
\begin{aligned}
\widetilde{\mathbf{A}}(:,i) \leftarrow & \big(\widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \widetilde{\mathbf{B}}(\mathcal{J}_i,:) + c\mathbf{I}_R\big)^{-1}\big(\widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \cdot \\
& \mathbf{X}(i,\mathcal{J}_i)^T + \rho(\widetilde{\mathbf{A}}(:,i) + \mathbf{U}(i,:)^T)\big)
\end{aligned} \tag{11a}
$$

$$
\mathbf{A}(i,:) \leftarrow [\widetilde{\mathbf{A}}(:,i)^T - \mathbf{U}(i,:)]_+ \tag{11b}
$$

$$
\mathbf{U}(i,:) \leftarrow \mathbf{U}(i,:) + \mathbf{A}(i,:) - \widetilde{\mathbf{A}}(:,i)^T \tag{11c}
$$

where $c := \lambda + \rho$, $[.]_+$ is the projection on $\mathbb{R}_+$ by zeroing the negative entries, and $\mathcal{J}_i$ is the set of items that have observations for the ith individual. We use the adaptive

$\rho = \|\widetilde{\mathbf{B}}\|_F^2/NR$, which is a scaled version of ρ suggested in (Huang, Sidiropoulos, and Liavas 2016). The ADMM steps in (11) are performed until a termination criterion is met. We adopt the criterion in (Boyd, Parikh, and Chu 2011; Huang, Sidiropoulos, and Liavas 2016), namely, the primal and dual residuals

$$
\begin{aligned}
p_i &= \|\mathbf{A}(i,:) - \widetilde{\mathbf{A}}(:,i)^T\|_F^2 / \|\mathbf{A}(i,:)\|_F^2; \\
d_i &= \|\mathbf{A}(i,:) - \mathbf{A}_0(i,:)\|_F^2 / \|\mathbf{U}(i,:)\|_F^2;
\end{aligned} \tag{12}
$$

where $\mathbf{A}_0$ is $\mathbf{A}$ from the previous iteration. We iterate between the ADMM updates until p and d are smaller than a predefined threshold, or we reach the maximum number of iterations K – in our experiments we set $K = 5$.

Scalability Considerations: There are some important observations regarding the implementation of the ADMM updates in (11). First, we do not compute the matrix inversion in (11a) explicitly. Instead, the *Cholesky decomposition* of the Gram matrix $\mathbf{G}_i := \widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \widetilde{\mathbf{B}}(\mathcal{J}_i,:) + c\mathbf{I}_R$ is computed, i.e., $\mathbf{G}_i = \mathbf{L}_i \mathbf{L}_i^T$, where $\mathbf{L}_i$ is a lower triangular matrix. Then, at each ADMM iteration, we only need to perform a forward and a backward substitution to get the solution of $\widetilde{\mathbf{A}}(:,i)$. Thus, the step in (11a) is replaced with:

$$
\mathbf{G}_i \leftarrow \widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \widetilde{\mathbf{B}}(\mathcal{J}_i,:) + c\mathbf{I}_R; \ \mathbf{L}_i \leftarrow \mathrm{Cholesky}(\mathbf{G}_i) \tag{13a}
$$

$$
\mathbf{F}_i \leftarrow \widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \mathbf{X}(i,\mathcal{J}_i)^T \tag{13b}
$$

$$
\widetilde{\mathbf{A}}(:,i) \leftarrow \mathbf{L}_i^{-T}\mathbf{L}_i^{-1}\big(\mathbf{F}_i + \rho(\widetilde{\mathbf{A}}(:,i) + \mathbf{U}(i,:)^T)\big) \tag{13c}
$$

Computing the Cholesky decomposition requires $\mathcal{O}(R^3)$ flops, and the back and forward substitution steps cost $\mathcal{O}(NR^2)$. The matrix multiplication in $\widetilde{\mathbf{B}}(\mathcal{J}_i,:)^T \widetilde{\mathbf{B}}(\mathcal{J}_i,:)$ and in computing $\mathbf{F}_i$ in (13b) takes $\mathcal{O}(|\mathcal{J}_i|R^2)$ and $\mathcal{O}(|\mathcal{J}_i|R)$, respectively, where, $|\mathcal{J}_i| \leq N$ is the cardinality of the set $\mathcal{J}_i$. An important implication is that $\mathbf{L}_i$ and $\mathbf{F}_i$ do not change throughout the ADMM iterations, thus can be cached to save computation. The overall complexity to update $\mathbf{A}$ is $\mathcal{O}(NR^2)$. Moreover, the ADMM updates enjoy row separability, allowing parallel computation. In the case where $\mathbf{X}$ is fully observed, $\mathbf{G} := \widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}} + c\mathbf{I}_R$ and $\mathbf{F} := \widetilde{\mathbf{B}}^T\mathbf{X}^T$ are shared not only across the ADMM iterations, but also among the N parallel sub-problems corresponding to the rows of $\mathbf{A}$. Finally, the outer AO routine naturally provides a good initial point (warm-start) to the inner ADMM iterations (for both $\mathbf{A}$ and its dual variable $\mathbf{U}$), resulting in a faster convergence.

Next, we update $\mathbf{B}_1$ using ADMM in the same fashion as $\mathbf{A}$. The updates of $\mathbf{Z}_1$ and $\mathbf{D}$ admit closed form solutions

$$
\mathbf{Z}_1(j,:) = \mathbf{B}_1(j,:)/\|\mathbf{B}_1(j,:)\|_2; \ d_j = \mathbf{h}_j^T \mathbf{X}(\mathcal{I}_j, j)/\mathbf{h}_j^T \mathbf{h}_j \tag{14}
$$

where $\mathbf{h}_j = (\mathbf{B}_1(j,:)\mathbf{A}^T)^T$, and $\mathcal{I}_j$ is the set of individuals that have observations for the jth item.

In the next step, we perform few inner iterations to alternate between updating the tree structure triplets $\{\mathbf{S}_{q-1}, \mathbf{B}_q, \mathbf{Z}_q\}_{q=2}^{Q-1}$ and $\mathbf{B}_Q$ in a cyclic fashion (we call it the tree loop) – in the experiments we set the maximum number of tree iterations $T = 5$. The updates w.r.t. $\{\mathbf{B}_q\}_{q=2}^{Q-1}$ are unconstrained least squares problems. These

Algorithm 1: Algorithmic Solution to eTREE

---

1 Initialize: $\mathbf{A}, \mathbf{B}_1 \leftarrow$ NMF;

2 $\{\mathbf{B}_q\}_{q=2}^Q$ and $\{\mathbf{S}\}_{q=1}^{Q-1} \leftarrow$ random; $\mathbf{D} \leftarrow \mathbf{I}$; $\mathbf{U} \leftarrow \mathbf{0}$

3 repeat

4   Compute $\mathbf{L}_i$, and $\mathbf{F}_i$, $\forall i$ using (13a) and (13b)

5   Set $k = 1$ // counter of ADMM loop

6   **while** p_i, d_i *in* (12) $> \epsilon$ *and* $k < K$ **do**

7   Update $\widetilde{\mathbf{A}}(:,i)$, $\mathbf{A}(i,:)$, and $\mathbf{U}(i,:)$, $\forall i$ using (13c), (11b), and (11c), respectively

8 $k = k + 1$

9   **end**

10   Update $\mathbf{B}_1$ using ADMM loop (similar to $\mathbf{A}$)

11   Update $\mathbf{D}$ and $\mathbf{Z}_1$ using (14)

12   Set $t = 1$ // counter of tree loop

13   **while** $t < T$ **do**

14   **for** $q = 2, \ldots, Q - 1$ **do**

15   compute $\mathbf{L}$ using (15a)

16   update $\mathbf{B}_q(:,j)$, $\forall j$ using (15b)

17   update $\mathbf{S}_{q-1}(i,:)$, $\forall i$ using (17)

18 $\mathbf{Z}_q(i,:) = \mathbf{B}_q(i,:)/\|\mathbf{B}_q(i,:)\|_2$, $\forall i$

19   **end**

20   update $\mathbf{B}_Q(i,:)$, $\forall i$ using (16)

21   t = t + 1

22   **end**

23 until *convergence*

---

problems are column separable with a common mixing matrix. Thus, the complexity can be reduced by computing *one* Cholesky decomposition. Then, at each iteration, the update of each column only requires a forward and a backward substitution as follows

$$\mathbf{H} \leftarrow \mu\mathbf{S}_{q-1}^T\mathbf{S}_{q-1} + v\mathbf{I}_{M_q}; \ \mathbf{L} \leftarrow \text{Cholesky}(\mathbf{H}) \quad (15a)$$

$$\mathbf{B}_q(:,j) \leftarrow \mathbf{L}^{-T}\mathbf{L}^{-1}\big(\mu\mathbf{S}_{q-1}\mathbf{B}_{q-1}(:,j) + \mu\mathbf{S}_q\mathbf{B}_{q+1}(:,j)\cdot$$
$$+ \eta\mathbf{Z}_q(:,j)\big) \quad (15b)$$

where $v := \mu + \eta$. The updates of $\mathbf{B}_Q$ and the matrices $\{\mathbf{S}\}_{q=1}^{Q-1}$ are similar to solving for the centroids and the assignment variables in the k-means algorithm, respectively. Let $\mathcal{T}_{m_Q} = \{i|\mathbf{S}_{Q-1}(i, m_Q) = 1\}$, then each row in $\mathbf{B}_Q$ is

$$\mathbf{B}_Q(m_Q,:) \leftarrow \sum_{i \in \mathcal{T}_{m_Q}} \mathbf{B}_{Q-1}(i,:)/|\mathcal{T}_{m_Q}| \quad (16)$$

And the ith row of the assignment matrices is updated using

$$\mathbf{S}_q(i,k) \leftarrow \begin{cases} 1, & k = \arg\min_{m_q} \|\mathbf{B}_{q-1}(i,:) - \mathbf{B}_q(m_q,:)\|_2 \\ 0, & \text{otherwise} \end{cases}$$
$$(17)$$

The overall algorithm is summarized in Algorithm 1. One nice property of the proposed algorithm is that all the updates are row separable and can be computed in a distributed fashion (with the exception of $\{\mathbf{B}_q\}_{q=2}^{Q-1}$, which are *column* separable).

Experiments[1]

In this section, we evaluate the proposed framework on real data from various application domains: healthcare analytics,

---

[1]Code is at: https://github.com/FaisalAlmutairi/eTREE

movie recommendations, and education. This section aims to answer the following questions:

Q1. Accuracy: Does eTREE improve the quality of embeddings for the downstream tasks?

Q2. Interpretability: How meaningful is the tree structure learned by eTREE from an application domain knowledge viewpoint?

Datasets: We evaluate eTREE and the competing baselines on the following real datasets: (i) **Med-HF:** These data are provided by IQVIA Inc. and include the counts of medical services performed on patients with heart failure (HF) conditions, including patients with preserved ejection fraction (pEF), and reduced ejection fraction (rEF). We include the $5,000$ patients with the most records in our experiments. The total number of medical services is 411. The majorities of the counts fall in the range of small numbers, with a small percentage of larger numbers, resulting in a "long tail" in the histogram of the data. To circumvent this, we apply a logarithmic transform on $\mathbf{X} + 1$ (we add the 1 to be slightly above the zero as we have nonnegativity constraints). The resulting range is $log(2) - 7.79$, and the sparsity of the data matrix is 78.01%, (ii) **Med-MCI:** These data are also provided by IQVIA Inc. and similar to Med-HF, but they include patients with mild cognitive impairment (MCI) conditions. Similarly, we include $5,000$ patients and the total number of medical services is 412. We also apply a logarithmic transform on the data. The final range of data is $log(2) - 6.98$, with a 77.76% sparsity, (iii) **Movielens:** Movielens (Harper and Konstan 2015) is a movie rating dataset and a popular baseline in recommender systems literature. It contains $\sim 10^5$ ratings. The data only include users with at least 20 ratings. We also filter out movies with less than 20 ratings. The total number of users is 943 and the total number of movies is $1,152$. The rating range is $1 - 5$, with 0.5 increments. The sparsity of this dataset is 90.98%, and (iv) **College Grades:** These data contain the grades of students from the College of Science and Engineering at the University of Minnesota spanning from Fall 2002 to Spring 2013. The total number of students is $5,703$, and the number of courses is 837. The grades take 11 discrete values (0, and 1 to 4 with increments of $1.\overline{33}$), and the sparsity of the data matrix is 96.28%.

Baselines: We compare to the plain NMF and following state-of-the-art methods from the literature: (i) **NMF:** non-negative Matrix factorization (1) regularized with ($\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2$), and implemented using ADMM (Huang, Sidiropoulos, and Liavas 2016), (ii) **BMF:** matrix factorization with rank-1 factors specified to capture items' and individuals' biases (Paterek 2007; Koren 2008); implemented using Stochastic Gradient Descent (SGD). The is a well-known approach in recommender systems and is considered a state-of-the-art method in student grade prediction (Almutairi, Sidiropoulos, and Karypis 2017), (iii) **AdaError:** a collaborative filtering model based on matrix factorization with learning rate that adaptively adjusts based on the prediction error (Li et al. 2018). AdaError is reported to have a state-of-the-art results on MovieLens (Rendle, Zhang, and Koren 2019), (iv) **HSR:** a hierarchical structure recommender system model that captures the tree structure in items (users) via factorizing the item (user) embeddings matrix into a prod-

	BMF		AdaError		HSR		NMF		eTREE		NMF+KM	
Data	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
Med-HF	0.9875	0.7721	0.9147	0.6858	0.9287	0.7094	<u>0.9031</u>	**0.6788**	**0.8873**	<u>0.6808</u>	1.0797	0.8094
Med-MCI	0.8034	0.6232	0.7468	0.5680	0.7807	0.5990	<u>0.7445</u>	<u>0.5612</u>	**0.7317**	**0.5611**	0.8781	0.6578
MovieLens	0.9300	0.7312	<u>0.9123</u>	<u>0.7165</u>	0.9216	0.7226	0.9286	0.7286	**0.9106**	**0.7136**	1.0182	0.8250
College Grades	0.5765	0.4254	0.5777	<u>0.4206</u>	0.5844	0.4339	<u>0.5755</u>	0.4229	**0.5601**	**0.4126**	0.5991	0.4476

Table 1: Matrix Completion Errors. eTREE significantly improves the prediction accuracy.

uct of matrices, i.e., $\mathbf{X} = \mathbf{A}_1\mathbf{A}_2\ldots\mathbf{A}_P(\mathbf{B}_1\mathbf{B}_2\ldots\mathbf{B}_Q)^T$ (Wang et al. 2018, 2015). $\mathbf{B}_Q \in \mathbb{R}^{M_Q \times R}$ can be interpreted as the embedding of the coarsest items categories, whereas the matrices $\{\mathbf{B}_q \in \mathbb{R}^{M_q \times M_{q+1}}\}_{q=1}^{Q-1}$ indicate the affiliation of the M_q subcategories (or items) with the M_{q+1} coarser categories. Note that an item can belong to all the subcategories with different scales since no constraints are imposed on $\mathbf{B}_q$'s matrices (except for nonnegativity). The same analysis also applies to user embeddings. We are unaware of other algorithms that incorporate the tree structure while learning the embeddings simultaneously. We used the Matlab code sample provided by the authors for a 3-layer tree and generalized it to handle Q layers, and (v) **NMF+KM:** is a simple two-stage procedure where we first apply NMF, then we obtain the embeddings of the root nodes $\mathbf{B}_Q$ and the product of the assignment matrices $\mathbf{S} = \mathbf{S}_1\ldots\mathbf{S}_{Q-1}$ via k-means' centroids and assignment variable, respectively – we include NMF+KM to demonstrate the advantage of learning the embeddings and tree structure simultaneously.
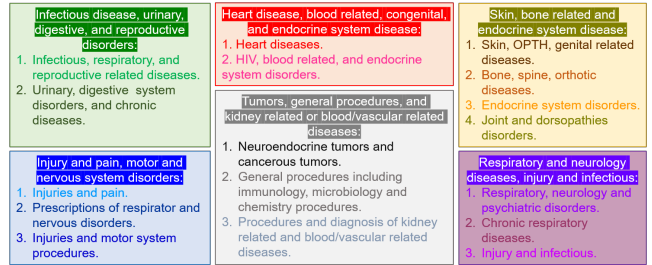
Q1. Accuracy of Embeddings: The quality of embeddings can be evaluated by testing their performance with a particular task, e.g., classification or regression. Here we take a more generic approach and evaluate the embedding quality on matrix completion. The philosophy is: *if the embeddings predict missing data with high accuracy, then they must be good representations of items and individuals.* We split each dataset into 5 equal folds. After training the models on 4 folds (80% of the data), we test the trained models on the held-out fold. The hyper-parameters of all methods are chosen via cross validation (10% of training data). Due to random initialization, the results can differ for different runs. Thus, after choosing the hyper-parameters, we run the training and testing on each fold 20 times and report the average error of the total 100 experiments.

Table 1 shows the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) of all methods on the different datasets. We highlight the smallest error in bold and underline the second smallest. eTREE significantly improves the best baseline with all datasets. Note that MovieLens and the grade datasets are challenging and an improvement in the second digit is considered significant in the literature. By comparing NMF and eTREE, we can conclude that the tree prior enhances the accuracy. Moreover, we can see the clear advantage of simultaneously learning the embeddings and tree clusters when we compare eTREE with MNF+KM. We point out that HSR baseline works better with MovieLens, compared to the medical datasets. This is likely because a movie usually belongs to a mix of genres, which suits the complete tree assumption in HSR. Nevertheless, the proposed tree formulation in eTREE provides better accuracy.

Q2. Interpretability of Learned Trees: We ran a 3-layer

(a) Sample of the 412 medical services in Med-MCI.

(b) A tree learned by eTREE and labeled by domain experts.

Figure 3: eTREE Provides Meaningful Clusters.

eTREE on Med-MCI with the following parameters: $R = 9$, $\lambda = 1$, $\mu = 50$ (more emphasis on the tree term), $M_2 = 27$ (number of subcategories), and $M_3 = 9$ (number of main categories). A sample of the 412 medical services is shown in Fig. 3 (a), where 'dx' stands for a diagnosis, 'rx' is a prescription, and 'px' is a procedure. Note that eTREE assigns each medical service to one subcategory, and each subcategory to a main category. Services with the *same* color in Fig. 3 (a) belong to the same subcategories, whereas services with *similar* colors (e.g., light and dark blue) belong to the same main category but to different subcategories. These unsupervised tree clusters were then shown to medical professionals. The domain experts were able to find coherence in the groups and they labeled both the main categories and their subcategories. The tables in Fig. 3 (b) shows the names of the main categories and their subcategories as labeled by the medical professionals – we show the top 6 coherent categories. Similar interpretability was observed on Med-HF data, but not shown due to space limitation.

Conclusion

In this paper, we proposed eTREE, a framework that incorporates the tree structure while learning the embeddings of a data matrix. eTREE not only exploits the tree structure, but also learns the hierarchical clustering in an supervised fashion. We leveraged the special properties of NMF to prove the uniqueness of the proposed model. We employed ADMM, parallel computing, and computation caching to derive a lightweight algorithm with scalable implementation to solve eTREE. We showed the effectiveness and interpretability of eTREE on real data.

Acknowledgements

References

Almutairi, F. M.; Sidiropoulos, N. D.; and Karypis, G. 2017. Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. *IEEE Journal of Selected Topics in Signal Processing* 11(5): 729–741.

Arora, S.; Ge, R.; Halpern, Y.; Mimno, D.; Moitra, A.; Sontag, D.; Wu, Y.; and Zhu, M. 2013. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, 280–288.

Boyd, S.; Parikh, N.; and Chu, E. 2011. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Fan, J.; and Cheng, J. 2018. Matrix completion by deep matrix factorization. *Neural Networks* 98: 34–41.

Fu, X.; Huang, K.; and Sidiropoulos, N. D. 2018. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters* 25(3): 328–332.

Fu, X.; Ma, W.-K.; Huang, K.; and Sidiropoulos, N. D. 2015. Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain. *IEEE Transactions on Signal Processing* 63(9): 2306–2320.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5(4): 1–19.

He, J.; Li, X.; and Liao, L. 2017. Category-aware Next Point-of-Interest Recommendation via Listwise Bayesian Personalized Ranking. In *IJCAI*, volume 17, 1837–1843.

Hoyer, P. O. 2004. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research* 5(Nov): 1457–1469.

Huang, K.; Sidiropoulos, N. D.; and Liavas, A. P. 2016. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing* 64(19): 5052–5065.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434.

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

Li, D.; Chen, C.; Lv, Q.; Gu, H.; Lu, T.; Shang, L.; Gu, N.; and Chu, S. M. 2018. Adaerror: An adaptive learning rate method for matrix approximation-based collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, 741–751.

Li, H.; Liu, Y.; Qian, Y.; Mamoulis, N.; Tu, W.; and Cheung, D. W. 2019. HHMF: hidden hierarchical matrix factorization for recommender systems. *Data Mining and Knowledge Discovery* 33(6): 1548–1582.

Lin, C.-H.; Ma, W.-K.; Li, W.-C.; Chi, C.-Y.; and Ambikapathi, A. 2015. Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Transactions on Geoscience and Remote Sensing* 53(10): 5530–5546.

Liu, M.; Luo, Y.; Tao, D.; Xu, C.; and Wen, Y. 2015. Low-rank multi-view learning in matrix completion for multi-label image classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2778–2784.

Maleszka, M.; Mianowska, B.; and Nguyen, N. T. 2013. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowledge-Based Systems* 47: 1–13.

Mao, X.; Sarkar, P.; and Chakrabarti, D. 2017. On mixed memberships and symmetric nonnegative matrix factorizations. In *International Conference on Machine Learning*, 2324–2333.

Menon, A. K.; Chitrapura, K.-P.; Garg, S.; Agarwal, D.; and Kota, N. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 141–149.

Nikolakopoulos, A. N.; Kouneli, M. A.; and Garofalakis, J. D. 2015. Hierarchical itemspace rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation. *Neurocomputing* 163: 126–136.

Paterek, A. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, 5–8.

Rendle, S.; Zhang, L.; and Koren, Y. 2019. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395* .

Shen, Y.; Tan, S.; Sordoni, A.; and Courville, A. 2018. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *International Conference on Learning Representations*.

Strehl, A.; Ghosh, J.; and Mooney, R. 2000. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, 64.

Sun, Z.; Yang, J.; Zhang, J.; and Bozzon, A. 2017. Exploiting both vertical and horizontal dimensions of feature hierarchy for effective recommendation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Trigeorgis, G.; Bousmalis, K.; Zafeiriou, S.; and Schuller, B. W. 2016. A deep matrix factorization method for learning attribute representations. *IEEE transactions on pattern analysis and machine intelligence* 39(3): 417–429.

Wang, S.; Tang, J.; Wang, Y.; and Liu, H. 2015. Exploring Implicit Hierarchical Structures for Recommender Systems. In *IJCAI*, 1813–1819.

Wang, S.; Tang, J.; Wang, Y.; and Liu, H. 2018. Exploring hierarchical structures for recommender systems. *IEEE*

Transactions on Knowledge and Data Engineering 30(6): 1022–1035.

Wang, X.; Pan, W.; and Xu, C. 2014. Hgmf: Hierarchical group matrix factorization for collaborative recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 769–778.

Wang, Y.; Wu, T.; Wang, Y.; and Wang, G. 2019. Enhancing model interpretability and accuracy for disease progression prediction via phenotype-based patient similarity learning. In *Pac Symp Biocomput*. World Scientific.

Yang, B.; Fu, X.; and Sidiropoulos, N. D. 2016. Learning from hidden traits: Joint factor analysis and latent clustering. *IEEE Transactions on Signal Processing* 65(1): 256–269.

Yang, J.; Sun, Z.; Bozzon, A.; and Zhang, J. 2016. Learning hierarchical feature influence for recommendation by recursive regularization. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 51–58.

Zhong, E.; Fan, W.; and Yang, Q. 2012. Contextual collaborative filtering via hierarchical matrix factorization. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, 744–755. SIAM.