Supervised Patient Similarity Measure of Heterogeneous Patient Records

Jimeng Sun, Fei Wang, Jianying Hu, Shahram Edabollahi
IBM TJ Watson Research Center
{jimeng,fwang,jyhu,ebad}@us.ibm.com

ABSTRACT

Patient similarity assessment is an important task in the context of patient cohort identification for comparative effectiveness studies and clinical decision support applications. The goal is to derive clinically meaningful distance metric to measure the similarity between patients represented by their key clinical indicators. How to incorporate physician feedback with regard to the retrieval results? How to interactively update the underlying similarity measure based on the feedback? Moreover, often different physicians have different understandings of patient similarity based on their patient cohorts. The distance metric learned for each individual physician often leads to a limited view of the true underlying distance metric. How to integrate the individual distance metrics from each physician into a globally consistent unified metric?

We describe a suite of supervised metric learning approaches that answer the above questions. In particular, we present Locally Supervised Metric Learning (LSML) to learn a generalized Mahalanobis distance that is tailored toward physician feedback. Then we describe the interactive metric learning (iMet) method that can incrementally update an existing metric based on physician feedback in an online fashion. To combine multiple similarity measures from multiple physicians, we present Composite Distance Integration (Comdi) method. In this approach we first construct discriminative neighborhoods from each individual metrics, then combine them into a single optimal distance metric. Finally, we present a clinical decision support prototype system powered by the proposed patient similarity methods, and evaluate the proposed methods using real EHR data against several baselines.

1. INTRODUCTION

With the tremendous growth of the adoption of Electronic Health Records (EHR), various sources of information are becoming available about patients. A key challenge is to identify the appropriate and effective secondary uses of EHR data for improving patient outcome without incurring additional effort from physicians. To achieve the goal of the meaningful reuse of EHR data, patient similarity becomes an important concept. The objective of patient similarity is to derive a similarity measure between a pair of patients based on their EHR data. With the right patient similarity in place, many applications can be enabled: 1) case-based retrieval of similar patients for a target patient; 2) treatment comparison among the cohorts of similar patients to a target patient; 3) cohort comparison and comparative effectiveness research.

One of the key challenges to deriving meaningful patient similarity measure is how to leverage physician input. In this work, we present a suit of approaches to encode physician input as supervised information to guide the similarity measure to address the following questions:

- How to adjust the similarity measure according to physician feedback?
- How to interactively update the existing similarity measure efficiently based on new feedback?
- How to combine different similarity measures from multiple physicians?

First, to incorporate physician feedback, we present an approach of using locally supervised metric learning (LSML) [20] to learn a generalized Mahalanobis measure to adjust the distance measure according to the target labels. The main approach is to construct two sets of neighborhoods for each training patient based on an initial distance measure. In particular, the homogeneous neighborhood of the index patient is the set of retrieved patients that are close in distance measure to the index patient and are also considered similar by the physician; the heterogeneous neighborhood of the index patient is the set of retrieved patients that are close in distance measure to the index patient but are considered NOT similar by the physician. Given these two definitions, both homogeneous and heterogeneous neighborhoods are constructed for all patients in the training data. Then we formulate an optimization problem that tries to maximize the homogeneous neighborhoods while at the same time minimizing the heterogeneous neighborhoods.

Second, to incorporate additional feedback to the existing similarity measure, we present the interactive Metric learning (iMet) method that can incrementally adjust the underlying distance metric based on latest supervision information [25]. iMet is designed to scale linearly with the data set size based on the matrix perturbation theory, which allows the derivation of sound theoretical guarantees. We show empirical results demonstrating that iMet outperforms the baseline by three orders of magnitude in speed while obtaining comparable accuracy on several benchmark datasets.

Third, to combine multiple similarity measures (one from each physician), we develop an approach that first constructs discriminative neighborhoods from each individual metrics, then we combine them into a single optimal distance metric. We formulate this problem as a quadratic optimization problem and propose an efficient alternating strategy to find the optimal solution [24]. Besides learning a globally consistent metric, this approach provides an elegant way to share knowledge across multiple experts (physicians) without sharing the underlying data, which enables the privacy preserving collaboration. Through our experiments on real claim datasets, we show improvement of classification accuracy as we incorporate feedback from multiple physicians.

All three techniques address different aspects of operationalizing patient similarity in the clinical application: The first technique locally supervised metric learning can be used to learn the distance metric in the batch mode where large amount of evidence first need to be obtained to form the training data. In particular, the training data should consist of 1) clinical features of patients such as diagnosis, medication, lab results, demographics and vitals, and 2) physician feedback about whether pair of patients are similar or not. For example, one simple type of feedback is binary indicator about each retrieved patient, where 1 means the retrieved patient is similar to the index patient and 0 means not similar. Then the supervised similarity metric can be learned over the training data using LSML algorithm. Finally, the learned similarity can be used in various applications for retrieving a cohort of similar patients to a target patient. The second and third techniques address other related challenges of using such a supervised metric, namely how to update the learned similar metric with new evidence efficiently and how to combine multiple physicians' opinions.

Obtaining high quality training data is very important but often challenging, since it typically imposes overhead on users, who are busy physicians in our case. An important benefit of our approaches is that the supervision required can come from various sources besides direct physician feedback, and could be implicitly collected without any additional overhead. For example, for some use case scenarios the training data could be simply information about patients such as diagnoses, which physicians routinely provide in every encounter.

We have conducted preliminary evaluation of all the proposed methods using claims data consisting of 200K patients over 3 years from a healthcare group consisting of primary care practices. A target diagnosis code assigned by physicians is considered as the feedback, while all other information (e.g., other diagnosis codes) are used as input features. The goal is to learn the similarity that push patients of the same diagnosis closer, and patient of different diagnosis far away from each other. Classification performance based on the target diagnosis is used as the evaluation metric. Our initial results show significant improvements over many baseline distance metrics.

The rest of the paper is organized as the follows: Section 2 describes the EHR and patient representation; Section 3 presents the locally supervised metric learning (LSML) method; Section 4 describes an extension of LSML that enables incremental updates of an existing similarity metric based on physician feedback; Section 5 presents an extension of LSML that can combine multiple supervised similarity metrics learned using LSML; Section 6 presents the experiments; section 7 presents the related works, and we conclude in section 8.

2. DATA AND PATIENT REPRESENTATION

We adopt a feature-based framework that serves as the basis for implementing different similarity algorithms. In particular, we systematically construct features from different data sources, recognizing that longitudinal data on even a single variable (e.g., blood pressure) can be represented in a variety of ways. The objective of our feature construction effort is to capture sufficient clinical nuances of heterogeneity among patients. A major challenge is in data reduction and in summarizing the temporal event sequences in EHR data into features that can differentiate patients.

We construct features from longitudinal sequences of observable measures based on demographics, diagnoses, medication, lab, vital signs, and symptoms. For the evaluation results presented in this paper, only diagnosis information is used. However other types of features can be generated and used in the similarity measure in a similar fashion.

Different types of clinical events arise in different frequency and in different orders. We construct summary statistics for different types of event sequences based on the feature characteristics: For static features such as gender and ethnicity, we will use a single static value to encode the feature. For temporal numeric features such as lab measures, we will use summary statistics such as point estimate, variance, and trend statistics to represent the features. For temporal discrete features such as diagnoses, we will use the event frequency (e.g., number of occurrences of a ICD9 code). For other measures such as blood pressure, we construct variance and trend in value. For other variables, we construct counting statistics such as number of encounters or number of symptoms at different time intervals. For complex variables, like medication prescribed, we model medication use as a time dependent variable and also express medication usage (i.e., percent of days pills may have been used) at different time intervals.

Essentially, each patient is represented by a feature vector, which serves as the input to the similarity measure. Our goal next is to design a similarity measure that operates on patient feature vectors and are consistent with physician feedback in terms of whether two patients are clinically similar or not.

3. SUPERVISED PATIENT SIMILARITY

In this section, we present a supervised metric learning algorithm that can incorporate physician feedback as supervision information. We use $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ to represent a feature matrix of a set of patients, and $\mathbf{y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^n$ is the corresponding label vector with $y_i \in \{1, 2, \cdots, C\}$ denoting the label of $\mathbf{x}_i$, and C is the number of classes. In particular, $\mathbf{x}_i$ corresponds to the feature vector of patient i, and the label y_i captures the supervision information from a physician. More specifically, if two patients have the same label information, it means that they are considered similar.

Our goal is to learn a *generalized Mahalanobis distance* as follows

$$d_\Sigma(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{\Sigma} (\mathbf{x}_i - \mathbf{x}_j)} \quad (1)$$

where $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ is a *Symmetric Positive Semi-Definite* (*SPSD*) matrix. Following [26], we define the Homogeneous Neighborhood and Heterogeneous Neighborhood around each data point as

DEFINITION 3.1. *The homogeneous neighborhood of $\mathbf{x}_i$, denoted as $\mathcal{N}_i^o$, is the $|\mathcal{N}_i^o|$-nearest data points of $\mathbf{x}_i$ with the same label.*

DEFINITION 3.2. *The heterogeneous neighborhood of $\mathbf{x}_i$, denoted as $\mathcal{N}_i^e$, is the $|\mathcal{N}_i^e|$-nearest data points of $\mathbf{x}_i$ with different labels.*

In the above two definitions we use $|\cdot|$ to denote set cardinality. Intuitively, $\mathcal{N}_i^o$ consists of true similar patients, who are considered similar by both our algorithm and the physician (because of the label agreement). Likewise, $\mathcal{N}_i^e$ consists of falsified similar patients, who are considered similar by the algorithm but not by the physician (because of label disagreement). The falsified similar patients are false positives that should be avoided by adjusting the underlying distance metric.

In order to learn the right distance metric based on the label information, we need to first construct both neighborhoods $\mathcal{N}_i^o$ and $\mathcal{N}_i^e$. Then we can define the local compactness and scatterness measures

around a feature vector $\mathbf{x}_i$ as

$$\mathcal{C}_i = \sum\nolimits_{j:\mathbf{x}_j \in \mathcal{N}_i^o} d_\Sigma^2(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$

$$\mathcal{S}_i = \sum\nolimits_{k:\mathbf{x}_k \in \mathcal{N}_i^e} d_\Sigma^2(\mathbf{x}_i, \mathbf{x}_k) \quad (3)$$

Ideally, we want small compactness and large scatterness simultaneously. To do so, we can want to minimize the following *discrimination* criterion:

$$\mathcal{J} = \sum\nolimits_{i=1}^{n} (\mathcal{C}_i - \mathcal{S}_i) \quad (4)$$

which makes the data in the same class compact while data in different class diverse. As $\mathbf{\Sigma}$ is SPSD, we can factorize it using incomplete Cholesky decomposition as

$$\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^\top \quad (5)$$

Then $\mathcal{J}$ can be expanded as[1]

$$\mathcal{J} = tr\left(\mathbf{W}^\top \left(\mathbf{\Sigma}_\mathcal{C} - \mathbf{\Sigma}_\mathcal{S}\right)\mathbf{W}\right) \quad (6)$$

where $tr(\cdot)$ is the matrix trace, and

$$\mathbf{\Sigma}_\mathcal{C} = \sum_i \sum\nolimits_{j:\mathbf{x}_j \in \mathcal{N}_i^o} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \quad (7)$$

$$\mathbf{\Sigma}_\mathcal{S} = \sum_i \sum\nolimits_{k:\mathbf{x}_k \in \mathcal{N}_i^e} (\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^\top \quad (8)$$

are the local *Compactness* and *Scatterness* matrices. Hence the distance metric learning problem can be formulated as

$$\min\nolimits_{\mathbf{W}:\mathbf{W}^\top\mathbf{W}=\mathbf{I}} tr\left(\mathbf{W}^\top \left(\mathbf{\Sigma}_\mathcal{C} - \mathbf{\Sigma}_\mathcal{S}\right)\mathbf{W}\right) \quad (9)$$

Note that the orthogonality constraint $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ is imposed to reduce the information redundancy among different dimensions of $\mathbf{W}$, as well as control the scale of $\mathbf{W}$ to avoid some arbitrary scaling. To further simplify the notations, let us define two symmetric square matrices

DEFINITION 3.3. **(Homogeneous Adjacency Matrix)** *The homogeneous adjacency matrix $\mathbf{H}^o$ is an $n \times n$ symmetric matrix with its (i,j)-th entry*

$$h_{ij}^o = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}_i^o \text{ or } \mathbf{x}_i \in \mathcal{N}_j^o \\ 0, & \text{otherwise} \end{cases}$$

DEFINITION 3.4. **(Heterogeneous Adjacency Matrix)** *The heterogeneous adjacency matrix $\mathbf{H}^e$ is an $n \times n$ symmetric matrix with its (i,j)-th entry*

$$h_{ij}^e = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}_i^e \text{ or } \mathbf{x}_i \in \mathcal{N}_j^e \\ 0, & \text{otherwise} \end{cases}$$

We also define $g_{ii}^o = \sum_j h_{ij}^o$ and $\mathbf{G}^o = \text{diag}(g_{11}^o, g_{22}^o, \cdots, g_{nn}^o)$[2]. Likewise, we define $g_{ii}^e = \sum_j h_{ij}^e$ and $\mathbf{G}^e = \text{diag}(g_{11}^e, g_{22}^e, \cdots, g_{nn}^e)$. Then we refer to

$$\mathbf{L}^o = \mathbf{G}^o - \mathbf{H}^o \quad (10)$$

$$\mathbf{L}^e = \mathbf{G}^e - \mathbf{H}^e \quad (11)$$

as the *Homogeneous Laplacian* and *Heterogeneous Laplacian*, respectively..

[1]Note that this is a *trace difference* criterion which has some advantages over optimizing the *trace quotient* criterion as adopted in [26], such as easy to manipulate, convexity, and avoid the singularity problem.

[2]$\text{diag}(\mathbf{x})$ creates a diagonal matrix with the entries in $\mathbf{x}$.

With definition 3.3, we can rewrite Eq.(2) as

$$\begin{aligned} \mathcal{C} &= \sum\nolimits_i \sum\nolimits_{\substack{j:\mathbf{x}_j \in \mathcal{N}_i^o \\ or\ \mathbf{x}_i \in \mathcal{N}_j^o}} \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2 \\ &= \sum\nolimits_i \sum\nolimits_j \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2 h_{ij}^o \\ &= 2\sum\nolimits_i g_{ii}^o \|\hat{\mathbf{x}}_i\|^2 - 2\sum\nolimits_{ij} \hat{\mathbf{x}}_i{}^\top \hat{\mathbf{x}}_j h_{ij}^o \\ &= 2tr\left(\mathbf{W}^\top\mathbf{X}\mathbf{L}^o\mathbf{X}^\top\mathbf{W}\right) \end{aligned} \quad (12)$$

Similarly, by combining definition 3.4 and Eq.(3), we can get

$$\begin{aligned} \mathcal{S} &= 2tr\left(\mathbf{W}^\top\mathbf{X}(\mathbf{G}^e - \mathbf{H}^e)\mathbf{X}^\top\mathbf{W}\right) \\ &= 2tr\left(\mathbf{W}^\top\mathbf{X}\mathbf{L}^e\mathbf{X}^\top\mathbf{W}\right) \end{aligned} \quad (13)$$

Then the optimization problem becomes

$$\min_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} tr\left(\mathbf{W}^\top\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top\mathbf{W}\right) \quad (14)$$

With the following *Ky Fan* theorem, we know that optimal solution of the above solution would be $\mathbf{W}^* = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d]$, where $\mathbf{w}_1 \ldots, \mathbf{w}_d$ are the eigenvectors of matrix $\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top$, whose corresponding eigenvalues are negative.

THEOREM 3.1. *(Ky Fan)[31]. Let $\mathbf{H} \in \mathbb{R}^{d\times d}$ be a symmetric matrix with eigenvalues*

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_d$$

and the corresponding eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d]$. Then

$$\lambda_1 + \lambda_2 + \cdots + \lambda_k = \max_{\mathbf{P}^\top\mathbf{P}=\mathbf{I}_k} tr(\mathbf{P}^\top\mathbf{H}\mathbf{P})$$

Moreover, the optimal $\mathbf{P}^* = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k]$ subject to orthonormal transformation.

The complete algorithm of *Locally Supervised distance Metric Learning* (*LSML*) is summarized in Algorithm 1.

Algorithm 1 LSML ALGORITHM

Require: Data matrix $\mathbf{X}$, Data label vector $\mathbf{y}$, Homogeneous neighborhood size $|\mathcal{N}_i^o|$, Heterogeneous neighborhood size $|\mathcal{N}_i^e|$, Projected dimensionality k of matrix $\mathbf{W}$

1: Construct homogeneous Laplacian $\mathbf{L}^o$ using Eq.(10)
2: Construct heterogeneous Laplacian $\mathbf{L}^e$ using Eq.(11)
3: Find the number of columns k of $\mathbf{W}$ as the total number of negative eigenvalues.
4: Set $\mathbf{W}$ as the k eigenvectors of $\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top$ with the k smallest eigenvalues.

There are some optimization tricks that can be applied to make Algorithm 1 more efficient, e.g., (1) when $\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top$ is sparse, we can resort to Lanczos iteration to accelerate it; (2) according to Ky Fan theorem, the optimal objective value of problem (14) is simply the sum of all negative eigenvalues of $\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top$. Therefore, we can automatically determine the number of columns of $\mathbf{W}$ to be the number of negative eigenvalues of $\mathbf{X}(\mathbf{L}^o - \mathbf{L}^e)\mathbf{X}^\top$.

4. PATIENT SIMILARITY UPDATE

LSML is particularly relevant for patient similarity, since it provides a natural way to encode the physician feedback. However, to really make this patient similarity applicable, we have to be able to efficiently and effectively incorporate new feedback from

physicians into the existing model. In other words, the learned distance metric needs to be incrementally updated without expensive rebuilding. We next present an efficient update algorithm that can adjust an existing distance metric when additional label information becomes available. In particular, we present the update algorithm which links changes to the projection matrix $\mathbf{W}$ to changes to the existing homogeneous and heterogeneous Laplacian matrices.

4.1 Metric Update Algorithm

The updates that we consider here are in the form of label changes of $\mathbf{y}$, which consequently leads to changes to the homogeneous and heterogeneous Laplacian matrices $\mathbf{L}^o$ and $\mathbf{L}^e$. The key idea here is to relate the metric update to eigenvalue and eigenvector updates of these Laplacian matrices.

Definition and Setup: To facilitate the discussion, we define the Laplacian matrix as

$$\mathbf{L} = \mathbf{L}^o - \mathbf{L}^e.$$

Next we introduce an efficient technique based on matrix perturbation [19] to adjust the learned distance metric according to changes of $\mathbf{L}$. Suppose that after adjustment, $\mathbf{L}$ becomes

$$\widetilde{\mathbf{L}} = \mathbf{L} + \Delta\mathbf{L}.$$

We define $\mathbf{M} = \mathbf{X}\mathbf{L}\mathbf{X}^\top$, and define $(\lambda_i, \mathbf{w}_i)$ as one eigeivalue-eigenvector pair of matrix $\mathbf{M}$. Similarly, we have $\widetilde{\mathbf{M}} = \mathbf{X}\widetilde{\mathbf{L}}\mathbf{X}^\top$ and define $(\widetilde{\lambda}_i, \widetilde{\mathbf{w}}_i)$ as one eigenvalue-eigenvector pair of $\widetilde{\mathbf{M}}$. Then we can rewrite $(\widetilde{\lambda}_i, \widetilde{\mathbf{w}}_i)$ as

$$\begin{aligned} \widetilde{\lambda}_i &= \lambda_i + \Delta\lambda_i \\ \widetilde{\mathbf{w}}_i &= \mathbf{w}_i + \Delta\mathbf{w}_i \end{aligned}$$

Next we can obtain

$$\mathbf{X}(\mathbf{L} + \Delta\mathbf{L})\mathbf{X}^\top(\mathbf{w}_i + \Delta\mathbf{w}_i) = (\lambda_i + \Delta\lambda_i)(\mathbf{w}_i + \Delta\mathbf{w}_i). \quad (15)$$

Now the key questions are how to compute changes to the eigenvalue $\Delta\lambda_i$ and eigenvector $\Delta\mathbf{w}_i$, respectively.

Eigenvalue update: Expanding Eq.(15), we obtain

$$\begin{aligned} &\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{w}_i + \mathbf{X}\Delta\mathbf{L}\mathbf{x}^\top\mathbf{w}_i + \mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i + \mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\Delta\mathbf{w}_i \\ &= \lambda_i\mathbf{w}_i + \lambda_i\Delta\mathbf{w}_i + \Delta\lambda_i\mathbf{w}_i + \Delta\lambda_i\Delta\mathbf{w}_i \end{aligned}$$

In this paper, we concentrate on first-order approximation, i.e., we assume all high order perturbation terms (such as $\mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\Delta\mathbf{w}_i$ and $\Delta\lambda_i\Delta\mathbf{w}_i$ in the above equation) are neglectable. By further using the fact that $\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{w}_i = \lambda_i\mathbf{w}_i$, we can obtain the following equation

$$\mathbf{X}\mathbf{L}\mathbf{X}^\top\Delta\mathbf{w}_i + \mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i = \lambda_i\Delta\mathbf{w}_i + \Delta\lambda_i\mathbf{w}_i$$

Now multiplying both sides of Eq.(15) with $\mathbf{w}_i^\top$ and because of the symmetry of $\mathbf{X}\mathbf{L}\mathbf{X}^\top$, we get

$$\Delta\lambda_i = \mathbf{w}_i^\top\mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i \quad (16)$$

Eigenvector update: Since the eigenvectors are orthogonal to each other, we assume that the change of the eigenvector $\Delta\mathbf{w}_i$ is in the subspace spanned by those original eigenvectors, i.e.,

$$\Delta\mathbf{w}_i \approx \sum\nolimits_{j=1}^{d} \alpha_{ij}\mathbf{w}_j \quad (17)$$

where $\{\alpha_{ij}\}$ are small constants to be determined. Bringing Eq.(17) into Eq.(15), we obtain

$$\mathbf{X}\mathbf{L}\mathbf{X}^\top\sum_{j=1}^{d}\alpha_{ij}\mathbf{w}_j + \mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i = \lambda_i\sum_{j=1}^{d}\alpha_{ij}\mathbf{w}_j + \Delta\lambda_i\mathbf{w}_i$$

which is equivalent to

$$\sum_{j=1}^{d}\lambda_j\alpha_{ij}\mathbf{w}_j + \mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i = \lambda_i\sum_{j=1}^{d}\alpha_{ij}\mathbf{w}_j + \Delta\lambda_i\mathbf{w}_i$$

Multiplying $\mathbf{w}_k^\top$ $(k \neq i)$ on both side of the above equation, we get

$$\lambda_k\alpha_{ik} + \mathbf{w}_k^\top\mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i = \lambda_i\alpha_{ik}$$

Therefore,

$$\alpha_{ik} = \frac{\mathbf{w}_k^\top\mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i}{\lambda_i - \lambda_k}$$

To get α_{ii}, we use the fact that

$$\begin{aligned} & \widetilde{\mathbf{w}}_i^\top\widetilde{\mathbf{w}}_i = 1 \\ \Longleftrightarrow \quad & (\mathbf{w}_i + \Delta\mathbf{w}_i)^\top(\mathbf{w}_i + \Delta\mathbf{w}_i) = 1 \\ \Longleftrightarrow \quad & 1 + 2\mathbf{w}_i^\top\Delta\mathbf{w}_i + O(\|\Delta\mathbf{w}_i\|^2) = 1 \end{aligned}$$

Discarding the high order term, and bringing in Eq.(17), we get $\alpha_{ii} = -\frac{1}{2}$. Therefore

$$\Delta\mathbf{w}_i = -\frac{1}{2}\mathbf{w}_i + \sum_{j\neq i}\frac{\mathbf{w}_j^\top\mathbf{X}\Delta\mathbf{L}\mathbf{X}^\top\mathbf{w}_i}{\lambda_i - \lambda_j}\mathbf{w}_j \quad (18)$$

Algorithm 2 METRIC UPDATE

Require: Data matrix $\mathbf{X}$, Initial label vector $\mathbf{y}$, Learned optimal projection matrix $\mathbf{W}$ as well as the eigenvalues λ, Expert feedback

1: Construct $\Delta\mathbf{L}$ based on $\mathbf{y}$ and expert feedback
2: **for** $i = 1$ to k **do**
3: Compute $\Delta\lambda_i$ using Eq.(16), and compute $\widetilde{\lambda}_i = \lambda_i + \Delta\lambda_i$
4: Compute $\Delta\mathbf{w}_i$ using Eq.(18), and compute $\widetilde{\mathbf{w}}_i = \mathbf{w}_i + \Delta\mathbf{w}_i$
5: **end for**

5. INTEGRATE MULTIPLE SOURCES

The above LSML algorithm and its extension on incremental update face a key challenge in distributed secure environments. For example in healthcare applications, different physicians or practices may be responsible for different cohorts of patients, and all the information of these patients (demographic, diagnosis, lab tests, pharmacy, etc.) should be kept confidential. Typically in a health network we have a number of physicians or practices. If we want to learn an objective distance metric to compare pairwise patient similarity across all providers in the network, LSML cannot be applied as it needs to input all the patient features. How to learn a good distance metric in this distributed environment?

In this section, we present a *Composite Distance Integration* (*Comdi*) algorithm to solve such problem. The goal of Comdi is to learn an optimal integration of those individual distance metrics on each group of patients. We follow the same framework as LSML to develop the Comdi algorithm.

Now we present how to integrate neighborhood information from multiple parties. First, we generalize the optimization objective; second, we present an alternating optimization scheme; at last, we provide the theoretical analysis on the quality of the final solution.

5.1 Objective function

We still aim at learning a generalized Mahalanobis distance as in Eq.(1) but integrating the neighborhood information from all parties. Here the q-th party constructs homogeneous neighborhood

$\mathcal{N}_i^o(q)$ and heterogeneous neighborhood $\mathcal{N}_i^e(q)$ for the i-th data point in it. Correspondingly, the compactness matrix $\mathbf{\Sigma}_{\mathcal{C}}^q$ and the scatterness matrix $\mathbf{\Sigma}_{\mathcal{S}}^q$ are computed and shared by the q-th party:

$$\begin{aligned}\mathbf{\Sigma}_{\mathcal{C}}^q &= \sum\nolimits_{i\in\mathcal{X}_q}\sum\nolimits_{j:\mathbf{x}_j\in\mathcal{N}_i^o(q)}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top\\ \mathbf{\Sigma}_{\mathcal{S}}^q &= \sum\nolimits_{i\in\mathcal{X}_q}\sum\nolimits_{k:\mathbf{x}_k\in\mathcal{N}_i^e(q)}(\mathbf{x}_i-\mathbf{x}_k)(\mathbf{x}_i-\mathbf{x}_k)^\top\end{aligned}$$

Similar to one party case presented in Eq.(6), we generalize the optimization objective as

$$\mathcal{J}=\sum\nolimits_{q=1}^m\alpha_q\mathcal{J}^q=\sum\nolimits_{q=1}^m\alpha_q tr\left(\mathbf{W}^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}\right) \quad (19)$$

where the importance score vector $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_m)^\top$ is constrained to be in a simplex as $\alpha_q\geqslant 0,\ \sum_q\alpha_q=1$, and m is the number of parties. Note that by minimizing Eq.(19), Comdi actually leverages the local neighborhoods of all parties to get a more powerful discriminative distance metric. Thus Comdi aims at solving the following optimization problem.

$$\begin{aligned}\min_{\boldsymbol{\alpha},\mathbf{W}}\quad & \sum\nolimits_{q=1}^m\alpha_q tr\left(\mathbf{W}^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}\right)+\lambda\Omega(\boldsymbol{\alpha})\\ s.t.\quad & \boldsymbol{\alpha}\geqslant 0,\ \boldsymbol{\alpha}^\top\mathbf{e}=1\\ & \mathbf{W}^\top\mathbf{W}=\mathbf{I}\end{aligned} \quad (20)$$

Here $\Omega(\boldsymbol{\alpha})$ is some regularization term used to avoid trivial solutions, and $\lambda\geqslant 0$ is the tradeoff parameter. In particular, when $\lambda=0$, i.e., without any regularization, only $\alpha_q=1$ for the best party, while all the others have zero weight. The best λ can be selected through cross-validation.

5.2 Alternating Optimization

It can be observed that there are two groups of variables $\boldsymbol{\alpha}$ and $\mathbf{W}$. Although the problem is not jointly convex with respect to both of them, it is convex with one group of variables with the other fixed. Therefore we can apply *block coordinate descent* to solve it. Specifically, if $\Omega(\boldsymbol{\alpha})$ is a convex regularizer with respect to $\boldsymbol{\alpha}$, then the objective is convex with respect to $\boldsymbol{\alpha}$ with $\mathbf{W}$ fixed, and is convex with respect to $\mathbf{W}$ with $\boldsymbol{\alpha}$ fixed.

Solving $\mathbf{W}$ with $\boldsymbol{\alpha}$ Fixed: Starting from $\boldsymbol{\alpha}=\boldsymbol{\alpha}^0$, at step t we can first solve the following optimization problem to obtain $\mathbf{W}^{(t)}$ with $\boldsymbol{\alpha}=\boldsymbol{\alpha}^{(t-1)}$

$$\begin{aligned}\min_{\mathbf{W}}\quad & \sum\nolimits_{q=1}^m\alpha_q^{(t-1)}tr\left(\mathbf{W}^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}\right)+\lambda\Omega(\boldsymbol{\alpha})\\ s.t.\quad & \mathbf{W}^\top\mathbf{W}=\mathbf{I}\end{aligned} \quad (21)$$

Note that the second term of the objective is irrelevant to $\mathbf{W}$, therefore we can discard it. For the first term of the objective, we can rewrite it as

$$\begin{aligned}&\sum\nolimits_{q=1}^m\alpha_q^{(t-1)}tr\left(\mathbf{W}^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}\right) \quad (22)\\ &=tr\left(\mathbf{W}^\top\left[\sum_{q=1}^m\alpha_q^{(t-1)}\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\right]\mathbf{W}\right)\end{aligned}$$

The optimal $\mathbf{W}$ is obtained by the Ky Fan theorem. In particular, we can solve problem (21), and set $\mathbf{W}^{(t)}=[\mathbf{w}_1^{(t)},\mathbf{w}_2^{(t)},\cdots,\mathbf{w}_k^{(t)}]$ with $\mathbf{w}_i^{(t)}$ being the eigenvector of

$$\mathbf{E}^{(t-1)}=\sum\nolimits_{q=1}^m\alpha_q^{(t-1)}\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)$$

whose eigenvalue is the i-th smallest. The worst computational complexity can reach $O(d^3)$ if $\mathbf{E}^{(t-1)}$ is dense.

Solving $\boldsymbol{\alpha}$ with $\mathbf{W}$ Fixed: After $\mathbf{W}^{(t)}$ is obtained, we can get $\boldsymbol{\alpha}^{(t)}$ by solving the following optimization problem.

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \sum\nolimits_{q=1}^m\alpha_q tr\left(\left(\mathbf{W}^{(t)}\right)^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}^{(t)}\right)+\lambda\Omega(\boldsymbol{\alpha})\\ s.t.\quad & \boldsymbol{\alpha}\geqslant 0,\ \boldsymbol{\alpha}^\top\mathbf{e}=1\end{aligned}$$

Here $\mathbf{e}$ is an all one vector. Now we analyze how to solve it with different choices of $\Omega(\boldsymbol{\alpha})$. For notational convenience, we denote $\mathbf{r}^{(t)}=(r_1^{(t)},r_2^{(t)},\cdots,r_m^{(t)})^\top$ with

$$r_q^{(t)}=tr\left(\left(\mathbf{W}^{(t)}\right)^\top\left(\mathbf{\Sigma}_{\mathcal{C}}^q-\mathbf{\Sigma}_{\mathcal{S}}^q\right)\mathbf{W}^{(t)}\right) \quad (23)$$

L2 regularization: Here $\Omega(\boldsymbol{\alpha})=\|\boldsymbol{\alpha}\|_2^2=\boldsymbol{\alpha}^\top\boldsymbol{\alpha}$, which is a common choice for regularization as adopted in SVM [17] and *Ridge Regression* [13] to avoid *overfitting*. In this case, the problem becomes

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \boldsymbol{\alpha}^\top\mathbf{r}^{(t)}+\lambda\|\boldsymbol{\alpha}\|_2^2\\ s.t.\quad & \boldsymbol{\alpha}\geqslant 0,\ \boldsymbol{\alpha}^\top\mathbf{e}=1\end{aligned} \quad (24)$$

which is a standard *Quadratic Programming* (QP) problem which can be solved by many mature softwares (e.g., the `quadprog` function in MATLAB). However, solving a QP problem is usually time consuming. Actually the objective of problem (24) can be reformulated as

$$\begin{aligned}&\boldsymbol{\alpha}^\top\mathbf{r}^{(t)}+\lambda\|\boldsymbol{\alpha}\|_2^2\\ &=\left\|\sqrt{\lambda}\boldsymbol{\alpha}+\frac{1}{\sqrt{2\lambda}}\mathbf{r}^{(t)}\right\|_2^2+\frac{1}{2\lambda}\left(\mathbf{r}^{(t)}\right)^\top\mathbf{r}^{(t)}\end{aligned}$$

As the second term $\frac{1}{2\lambda}\left(\mathbf{r}^{(t)}\right)^\top\mathbf{r}^{(t)}$ is irrelevant with $\boldsymbol{\alpha}$, we can discard it and rewrite problem (24) as

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \left\|\boldsymbol{\alpha}-\widetilde{\mathbf{r}}^{(t)}\right\|_2^2\\ s.t.\quad & \boldsymbol{\alpha}\geqslant 0,\ \boldsymbol{\alpha}^\top\mathbf{e}=1\end{aligned} \quad (25)$$

Here $\widetilde{\mathbf{r}}^{(t)}=\frac{1}{\sqrt{2\lambda}}\mathbf{r}^{(t)}$. Therefore this is just an *Euclidean projection* problem under the simplex constraint, several researchers have proposed *linear time* approaches to solve this type of problem [7; 16].

6. EVALUATION AND CASE STUDY

We first present a use case demonstration of patient similarity, then present all the quantitative evaluation of the algorithm.

6.1 Use Case Demostration

In this section we present a prototype system that uses patient similarity for clinical decision support. Fig.1 shows two snapshots of the system, where there are three tabs on the left side. When the physician inputs the ID of a patient and clicks the "Patient Summary" tab, the system displays all the information related to the index patient as shown in Fig.1(a). Once the physician clicks the "Similar Patient" tab, the system automatically retrieves N similar patients and visualize them as shown in Fig.1(b) according to some underlying patient similarity metric. The physician can further see the details of these retrieved patients by clicking the "Details" and "Comparison" tabs on the same page.

The underlying patient similarity metric is initially learned with the supervised metric learning method described in section 3. After the physician is presented with the view of N similar patients

(a) Patient Summary Tab

(b) Similar Patientsd Tab

Figure 1: Snapshots of a real world Physician decision support system developed by our team.

as described above, he/she can provide feedback using the "Similarity Evaluation" tab shown in Fig.2. The system then takes the input and updates the distance metric using the method described in section 4.

Currently, the system supports the following types of physician feedback.

- The selected patient $\mathbf{x}_u$ is highly similar to the query patient $\mathbf{x}_q$. This feedback is interpreted as indicating that $\mathbf{x}_u$ should be in the homogeneous neighborhood, and not in the heterogeneous neighborhood of $\mathbf{x}_q$.

- The selected patient $\mathbf{x}_v$ bears low similarity to the query patient $\mathbf{x}_q$. This feedback is interpreted as indicating that $\mathbf{x}_v$ should be in the heterogeneous neighborhood, and not in the homogeneous neighborhood of $\mathbf{x}_q$.

- The selected patient has medium similarity to the query patient, i.e., the physician is unsure whether the selected patient should be considered similar to the query patient. In this case, the selected patients is simply considered unlabeled and the corresponding elements in both homogeneous and heterogeneous adjacency matrices are set to 0.

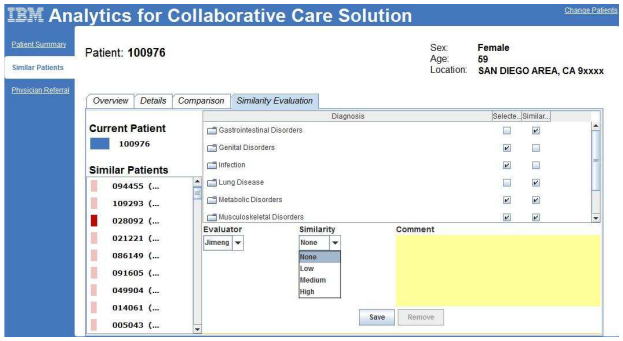

Figure 2: The "Similarity Evaluation" tab under "Similar Patients", where the physician can input his own feedback on whether a specific patient is similar to the query patient or not.

6.2 Effective Update Evaluation

In order to evaluate the performance of the update algorithm in a real world setting, we designed experiments that emulate physicians' feedback using existing diagnosis labels in a clinical data set containing records of 5000 patients. First, diagnosis codes were grouped according to their HCC category ([2], which resulted in a total of 195 different disease types. We select HCC019 which is diabetes without complication as the target condition. We use the presence and absence in patients' records of HCC019 code as the label and surrogates for physician feedback. That is, patients who had the same label were considered to be highly similar, and those who had different labels were considered to have low similarity.

The experiments were then set up as follows. First, the patient population was clustered into 10 clusters using Kmeans with the 194 dimensional features. An initial distance metric was then learned using LSML(described in section 3). For each round of simulated feedback, an index patient was randomly selected and 100 similar patients were retrieved using the current metric. Then 20 of these 100 similar patients were randomly selected for feedback based on the target label. These feedbacks were then used to update the distance metric using algorithm described in section 4.

The quality of the updated distance was then evaluated using the **precision@position** measure, which is defined as follows.

DEFINITION 6.1. **(Precision@Position)**. *On a retrieved list, the precision@position value is computed as the percentage of the patient with the same label as the query patient before some specific position.*

We calculated the precision values at different positions over the whole patient population. Specifically, after each round of feedback, the distance metric was updated. Then for each patient the 100 most similar patients were retrieved with the updated distance metric. we then computed the retrieval precision at different positions along this list and then averaged over all the patients.

Fig.3 illustrates the results on different diseases. From the figure we can see that with increasing number of feedback rounds, the retrieved precision becomes consistently higher.

6.3 Distance Integration Evaluation

We next evaluate the distance integration method described in section 5 through the clinical decision support scenario. We partition all the patients based on their primary care physicians. Each partition is called a patient cohort. We pick 30 patient cohorts to perform our experiments. We report the performance in terms of precision of different methods trained using all patients (shared version) and

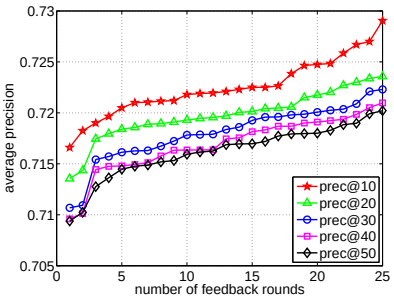

Figure 3: Precision variation at different positions with respect to the number of feedback rounds. The x-axis is the number of physician feedback rounds, which varies from 1 to 25. y-axis is the retrieved precision averaged over the whole population.

trained using only one patient cohort (secure version).

Besides our Comdi method described in section 5, we also present the performance of LSML. We also include *Principal Component Analysis* (**PCA**) [14], *Linear Discriminant Analysis* (**LDA**) [8] and *Locality Sensitive Discriminant Analysis* (**LSDA**) [5] as additional baselines. Moreover, the results using simple *Euclidean distance* (**EUC**) is also presented as a baseline.

For LSML, LSDA and Comdi, we fix $|\mathcal{N}_i^o| = |\mathcal{N}_i^e| = 5$. For Comdi, we use Algorithm 1 with m=30, and λ is set by cross validation. We report the classification performance for HCC019 in Fig.4 in terms of accuracy, recall, precision and F1 score, where the performance for secure version methods are averaged over 30 patient cohorts and we also show the standard deviation bars. From the figures we can see that Comdi significantly outperforms other secure version methods and can achieve almost the same performance as the shared version of LSML.

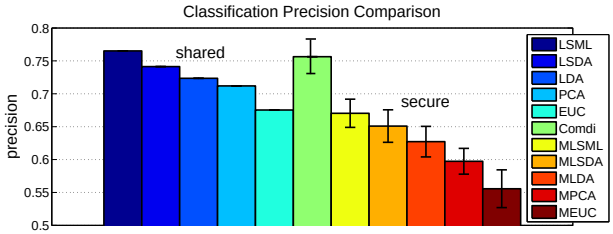

Figure 4: Classification performance comparison with different measurements on our data set with HCC019.

Effect of distance integration: In the second part of the experiments, we test how the performance of Comdi is affected by the choice of individual cohorts. For the evaluation purpose, we also hold-out one fixed set of 2000 patients for testing. The idea is that because some cohorts represent well the entire patient distribution, which often leads to good base metric. On the other hand, some cohorts do not represent the entire patient distribution, which often leads to bad base metric. We call the former *representative cohorts* and the latter *biased cohorts*. What is the effect of incorporating other metrics learned from a set of mixed cohorts? In particular, we want to find out 1) whether the base metric learned from a biased cohort will improve as incorporating other metrics; 2) whether the base metric learned from a representative cohort will improve as incorporating other metrics.

From each HCC code, we select a representative cohort and a biased cohort to build the base metric using LSML. Then we start adding other base metrics learned from other cohorts sequentially and check the accuracy changes during this process. We repeat the experiments 100 times and report the averaged classification accuracy as well as the standard deviation, which are shown in Fig.5. From the figure we clearly observe that by leveraging other metrics, the accuracy increases significantly for biased cohorts, and also still improves the accuracy for representative cohorts.

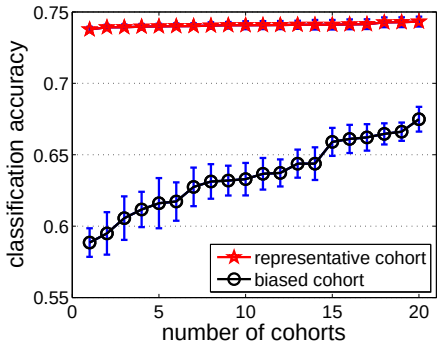

Figure 5: The affect of Combi on specific physicians on HCC019. The x-axis corresponds to the number of patient cohorts integrated. The y-axis represents the classification accuracy: the accuracy increases significantly for biased cohorts, and also still improves the accuracy for representative cohorts.

7. RELATED WORK

Distance Metric Learning (*DML*) [29] is a fundamental problem in data mining field. Depending on the availability of supervision information in the training data set (e.g., labels or constraints) , a DML algorithm can be classified as *unsupervised* [6][12][14], or *semi-supervised* [23][28] and *supervised*[8; 11; 27]. In particular, *Supervised DML* (*SDML*) constructs a proper distance metric that leads the data from the same class closer to each other, while the data from different classes far apart from each other.

SDML can further be categorized as *global* and *local methods*. A global SDML method attempts to learn a distance metric that keep *all* the data points within the same classes close, while separating *all* the data points from different classes far apart. Typical approaches in this category include *Linear Discriminant Analysis* (*LDA*) [8] and its variants [10][30]. Although global SDML approaches achieve empirical success in many applications, generally it is hard for a global SDML to separate data from different classes well [22], because the data distribution are usually very complicated such that data from different classes are entangled together. Local SDML methods, on the other hand, usually first construct some local regions (e.g., neighborhoods around each data points), and then in each local region, they try to pull the data within the same class closer, and push the data in different classes apart. Some representative algorithms include *Large Margin Nearest Neighbor* (*LMNN*) classifier [27], *Neighborhood Component Analysis* (*NCA*) [11], *Locality Sensitive Discriminant Analysis* (*LSDA*) [5], as well as the LSML method described in section 3. It is empirically observed that these local methods can generally perform much better than global methods. Most of the methods are offline methods that require model building on training data. However, the iMet described in section 4 can incrementally update the existing metric when feedback becomes available.

Another set of methods that closely related to Comdi is *Multiple Kernel Learning* (*MKL*) [15][3][18], which aims to learn an integration of kernel function from multiple base kernels. These approaches usually suppose that there is a initial set of "weak" kernels defined over the whole data set and the goal is to learn a "strong" kernel, which is some linear combination of these kernels. In MKL, all the weak kernels as well as the final strong kernel are required to defined on the same set of data, which cannot be used in the distributed environment as Comdi.

Comdi is also related to *Ensemble Methods*, such as *Bagging* [4] and *Boosting* [9]. What ensemble methods do is to obtain a strong learner via combining a set of weak learners, where each weak learner is learned from a sampled subset of the entire data set. At each step, the ensemble methods just sample from the whole data set according to some probability distribution with replacement and learn a weak learner on the sampled set. This is also different from the Comdi setting where the data in different parties are fixed.

Comdi is related to the area of privacy preserving data mining [1]. Different from of data perturbation and encrypted database schemes, Comdi share only models instead of data. Comdi falls into the general category of private distributed mining [21], which focus on building local mining models first before combining at the global level.

8. CONCLUSION

In this paper, we present a supervised patient similarity problem. The aim is to learn a distance metric between patients that are consistent with physician belief. We formulate the problem as a supervised metric learning problem, where physician input is used as the supervision information. First, we present locally supervised metric learning (LSML) algorithm that learns a generalized Mahalanobis distance with physician feedback as the supervision. The key there is to compute local neighborhoods to separate the true similar patients with other patients for an index patient. The problem is solved via the trace difference optimization. Second, we extend LSML to handle incremental updates. The goal is to enable online updates of the existing distance metric. Third, we generalize LSML to integrate multiple physician's similarity metrics into a consistent patient similarity measure. Finally, we demonstrated the use cases through a clinical decision support prototype and quantitatively compared the proposed methods against baselines, where significant performance gain is obtained. It is worth noting that the algorithms should work equally well with other sources of supervision besides direct physician input (e.g., labels derived directly from data).

For future work, we plan to use the patient similarity framework to address other clinical applications such as comparative effectiveness research and treatment comparison.

9. REFERENCES

[1] C. Aggarwal and P. S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[2] A. S. Ash, R. P. Ellis, G. C. Pope, J. Z. Ayanian, D. W. Bates, H. Burstin, L. I. Iezzoni, E. MacKay, and W. Yu. Using diagnoses to describe populations and predict costs. *Health care financing review*, 21(3):7–28, 2000. PMID: 11481769.

[3] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proc. of International Conference on Machine Learning*, pages 6–13, 2004.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 708–713, 2007.

[6] T. F. Cox and M. A. A. Cox. *Multimensional Scaling*. London, U. K., 2001.

[7] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[8] R. O. Duda, P. E. Hart, and D. H. Stork. *Pattern Classification (2nd ed.)*. Wiley Interscience, 2000.

[9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[10] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[11] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520, 2005.

[12] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press, 2002.

[13] A. E. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[14] I. Jolliffe. *Principal Component Analysis (2nd ed.)*. Springer Verlag, Berlin, Germany, 2002.

[15] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[16] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *International Conference on Machine Learning*, pages 657–664, 2009.

[17] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, Cambridge, MA, 2002.

[18] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[19] G. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1990.

[20] J. Sun, D. Sow, J. Hu, and S. Ebadollahi. Localized supervised metric learning on temporal physiological data. In *ICPR*, 2010.

[21] J. Vaidya, C. Clifton, and M. Zhu. *Privacy preserving data mining*. Springer, 2005.

[22] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[23] F. Wang, S. Chen, T. Li, and C. Zhang. Semi-supervised metric learning by maximizing constraint margin. In *Proceedings of ACM 17th Conference on Information and Knowledge Management*, pages 1457–1458, 2008.

[24] F. Wang, J. Sun, and S. Ebadollahi. Integrating distance metrics learned from multiple experts and its application in patient similarity assessment. In *SDM*, 2011.

[25] F. Wang, J. Sun, J. Hu, and S. Ebadollahi. imet: Interactive metric learning in healthcare application. In *SDM*, 2011.

[26] F. Wang, J. Sun, T. Li, and N. Anerousis. Two heads better than one: Metric+active learning and its applications for it service classification. In *IEEE International Conference on Data Mining*, pages 1022–1027, 2009.

[27] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[28] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing System 15*, pages 505–512, 2003.

[29] L. Yang. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

[30] J. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. volume 7, pages 1183–1204, 2006.

[31] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*, pages 1057–1064, 2001.