Kernelized Information-Theoretic Metric Learning for Cancer Diagnosis Using High-Dimensional Molecular Profiling Data

FEIYU XIONG, Drexel University
MOSHE KAM, New Jersey Institute of Technology
LEONID HREBIEN, Drexel University
BEILUN WANG and YANJUN QI, University of Virginia

With the advancement of genome-wide monitoring technologies, molecular expression data have become widely used for diagnosing cancer through tumor or blood samples. When mining molecular signature data, the process of comparing samples through an adaptive distance function is fundamental but difficult, as such datasets are normally heterogeneous and high dimensional. In this article, we present kernelized information-theoretic metric learning (KITML) algorithms that optimize a distance function to tackle the cancer diagnosis problem and scale to high dimensionality. By learning a nonlinear transformation in the input space implicitly through kernelization, KITML permits efficient optimization, low storage, and improved learning of distance metric. We propose two novel applications of KITML for diagnosing cancer using high-dimensional molecular profiling data: (1) for sample-level cancer diagnosis, the learned metric is used to improve the performance of *k*-nearest neighbor classification; and (2) for estimating the severity level or stage of a group of samples, we propose a novel set-based ranking approach to extend KITML. For the sample-level cancer classification task, we have evaluated on 14 cancer gene microarray datasets and compared with eight other state-of-the-art approaches. The results show that our approach achieves the best overall performance for the task of molecular-expression-driven cancer sample diagnosis. For the group-level cancer stage estimation, we test the proposed set-KITML approach using three multi-stage cancer microarray datasets, and correctly estimated the stages of sample groups for all three studies.

CCS Concepts: • **Computing methodologies → Machine learning algorithms**; *Supervised learning by classification*; • **Applied computing → Bioinformatics**

Additional Key Words and Phrases: Metric learning, cancer diagnosis, high-dimensional data

ACM Reference Format:
Feiyu Xiong, Moshe Kam, Leonid Hrebien, Beilun Wang, and Yanjun Qi. 2016. Kernelized information-theoretic metric learning for cancer diagnosis using high-dimensional molecular profiling data. ACM Trans. Knowl. Discov. Data 10, 4, Article 38 (May 2016), 23 pages.
DOI: http://dx.doi.org/10.1145/2789212

1. BACKGROUND AND MOTIVATION

Modern molecular profiling technologies have enabled researchers to query the expression values of thousands of genes simultaneously. Information derived from such genome-wide molecular profiling is important for the identification of cancer tumor types in patient samples [Ramaswamy et al. 2002]. In this article, we focus on the

Authors' addresses: F. Xiong and L. Hrebien, Drexel University, Electrical and Computer Engineering Department, 3141 Chestnut Street, Philadelphia, PA 19104; emails: fx26@drexel.edu, lhrebien@coe.drexel.edu; M. Kam, Office of the Dean, Newark College of Engineering, New Jersey Institute of Technology, 323 Dr. Martin Luther King, Jr. Boulevard, Newark, NJ 07102, USA; email: moshe.kam@njit.edu; B. Wang and Y. Qi, Department of Computer Science, University of Virginia, 85 Engineer's Way, P.O. Box 400740, Charlottesville, Virginia 22904-4740; emails: bw4mw@virginia.edu, yanjun@virginia.edu.

DOI: http://dx.doi.org/10.1145/2789212

task of cancer diagnosis based on the molecular signatures of patient samples like those from microarray. Recent advancements in monitoring of gene expression have enabled researchers to obtain enormous amounts of data across most common cancer types [Cancer Genome Atlas Research Network et al. 2011, 2012, 2013]. An important emerging medical application for molecular profiling technologies is in clinical decision support systems for diagnosing the existence of certain cancers as well as estimating of the severity levels or cancer stages of patient samples [Statnikov et al. 2005].

Machine learning techniques such as classification and clustering have been used for analysis and interpretation of data obtained from molecular profiling measurements [Statnikov et al. 2005; Cancer Genome Atlas Research Network et al. 2012]. These data are characterized by a very high number of measured variables (m genes) over a relatively small number of (n samples). The number of genes in a single sample is typically in the thousands and the number of samples is typically in the hundreds, so the number of feature variables (genes) greatly exceeds the number of samples. This data situation ($m >> n$) is referred as "high dimensionality" [Hastie et al. 2003] and makes machine learning quite challenging. Recent studies have tried to tackle the "high-dimensionality" issue when predicting the existence of cancer using molecular expressions, for example, through sparse-learning based approaches [Cawley and Talbot 2006]. As molecular signature data become available for more and more patient samples, e.g., from the national project The Cancer Genome Atlas [Hudson et al. 2010], measuring the similarity among patient samples grows to be a critical and necessary module for mining such signature data. For instance, such similarity measure could be used for molecular signature-based retrieval of similar cancer patient cases for a target. Therefore, relying on molecular profiling data, we aim to design an accurate cancer diagnosis system that is able to provide good assessments of patient similarity as well. Previous studies [Cawley and Talbot 2006; Cancer Genome Atlas Research Network et al. 2012] were not able to fully address such needs, especially when having the "curse-of-dimensionality" [Hastie et al. 2003] issue in place.

In this article, we extend a family of "distance metric learning"—"Information-Theoretic Metric learning (ITML)" [Kulis 2013]—for achieving the above goal. Having been studied over the past few years [Xing et al. 2003; Kulis 2013], distance metric learning was recently applied to practical areas such as image recognition [Kulis 2013] and information retrieval [Ying and Li 2012]. This article presents two novel extensions of metric learning on the tasks of sample-level cancer diagnosis and group-level stage diagnosis. The intrinsic data issue of "small sample, large feature" is addressed through "kernelizing" the learned metric from ITML method (i.e., Kernelized Information-Theoretic Metric Learning (KITML)).

Our approach has the following advantages:

—KNN algorithm with learned distance metric from KITML achieves better classification performance, for sample-level cancer classification using molecular features, than other state-of-the-art approaches (see Section 5.1.2).
—Designed to deal with high-dimensional data, KITML needs only to estimate $O(n^2)$ parameters, which are much smaller than the $O(m^2)$ parameters required for most other metric learning algorithms.
—The learned metric can be used to find similar patient cases for a target patient case using nearest neighbor (NN) search, which provides a great tool for physicians when designing relevant prognosis or treatment plans.

The rest of this article is organized as follows: Section 2 introduces the KITML algorithms under the cancer diagnosis setting. Section 3 discusses related research. Section 4 describes our experimental design and Section 5 shows performance evaluations. The last section states our conclusion.

2. KITML FOR HIGH-DIMENSIONAL MOLECULAR EXPRESSION DATA

The methods we present in this article belong to the family of "distance metric learning" algorithms that learn a distance metric when given a set of n training samples $\mathbf{x}_1, , \ldots, \mathbf{x}_n$ in which each $\mathbf{x}_i \in \mathbb{R}^m$ is a data vector of m features.

2.1. Basic Distance Metric Learning

Metric learning methods try to learn a Mahalanobis distance defined in Equation (1), where A is a positive semi-definite m by m matrix of parameters learned from data. The learning process usually relies on pairwise constraints between sample points as training signals: (1) equivalent constraints (Equation (2)), which state that a given pair of data points are semantically similar and should be close together in the learned metric; and (2) inequivalent constraints (Equation (3)), which indicate that the given pairs of samples are semantically dissimilar and should not be close together in the learned metric [Yang and Jin 2006].

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T A(\mathbf{x}_i - \mathbf{x}_j)} \tag{1}$$

$$\mathbf{S} = \{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i \textit{ and } \mathbf{x}_j \textit{ are similar}\} \tag{2}$$

$$\mathbf{D} = \{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i \textit{ and } \mathbf{x}_j \textit{ are dissimilar}\}. \tag{3}$$

A seminal formulation of distance metric learning [Xing et al. 2003] converts the above constraints to a convex programming task to learn the parameter matrix A.

$$\begin{aligned} \min_{A \in \mathbb{R}^{m \times m}} & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{S}} d_A(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}} d_A(\mathbf{x}_i, \mathbf{x}_j) \geq 1, \quad \text{and} \quad A \succeq 0. \end{aligned} \tag{4}$$

The positive semi-definite constraint $A \succeq 0$ is required to guarantee that the learned distance between any two points (parameterized by A) cannot be negative and satisfies the triangle inequality.

2.2. Kullback–Leibler (KL) Divergence and Connecting to Gaussian Distribution

Given a distance metric parameterized by A, a corresponding multivariate Gaussian distribution can be expressed for describing samples (assuming $\mathbf{x}$ has been centered) where A^{-1} is the covariance matrix of the distribution, that is,

$$Pr(x|A) = \frac{1}{(2\pi)^{m/2}|A|^{1/2}} \exp\left(-\frac{1}{2}x^T A^{-1} x\right). \tag{5}$$

Considering the Euclidean distance (i.e., distance metric with identity matrix $A_0 = I$) works well as a baseline empirically, we regularize the learned metric matrix A with A_0. Probabilistically, this serves to minimize the distance between the two corresponding Gaussian distributions, denoted by $Pr(x|A)$ and $Pr(x|A_0)$. Typically, KL divergence [Kullback and Leibler 1951] is used to measure the distance between two distributions, thus the distance between $Pr(x|A)$ and $Pr(x|A_0)$ is given by

$$\begin{aligned} d(A_0||A) &= KL(Pr(x|A_0)||Pr(x|A)) \\ &= \int Pr(x|A_0) \log \frac{Pr(x|A_0)}{Pr(x|A)} d\mathbf{x}. \end{aligned} \tag{6}$$

Then, the log determinate (LogDet) formulation is used to simplify the $d(A_0||A)$ in a closed form

$$d(A_0||A) = \frac{1}{2}(tr(A^{-1}A_0) + log|A| - log|A_0| - m), \tag{7}$$

where m is the dimensionality of the data. The proof of Equation (7) is in Appendix A.

2.3. ITML

Based on the above formulation, Davis et al. proposed ITML [Davis et al. 2007; Davis, Kulis, Jain, Sra, and Dhillon Davis et al.] to tackle metric learning by minimizing the LogDet divergence (Equation (7)) along with side constraints (equivalent or inequivalent). The constraints used in ITML are similar to those described in Section 2.1, in which for two similar samples, their learned distance is constrained to be smaller than a given upper bound, that is, $d_A(\mathbf{x}_i, \mathbf{x}_j) \leq u$ for a hyperparameter u, and, for two samples that are known to be dissimilar, $d_A(\mathbf{x}_i, \mathbf{x}_j) \geq l$ for a hyperparameter l. The objective is to learn a distance metric parameterized by parameter matrix A. To solve this optimization, ITML uses the so-called Bregman projections (also called Bregmans algorithm) for solving a strictly convex optimization with respect to multiple linear inequality constraints. Using this simple first-order technique developed in Bregman [1967], ITML repeatedly computes Bregman projections of the current solution onto a single constraint via the following update

$$A_{t+1} = A_t + \beta A_t(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T A_t, \tag{8}$$

where A_0 is chosen as the identity matrix $\mathbb{I}$ and β is the projection parameter (Lagrange multiplier) corresponding to the current constraint. It is positive for similar pairs and negative for dissimilar pairs.

2.4. KITML for High-Dimensional Data

When given a dataset $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]$, where each $\mathbf{x}_i \in \mathbb{R}^m$, ITML will learn a distance metric parameterized by a $m \times m$ matrix A. If the dataset is high dimensional (e.g., m is relatively large in gene microarray datasets), ITML needs to estimate m^2 parameters in A, which is not ideal when the sample size n is small. To adapt ITML for datasets with $n \ll m$, we employ the kernel trick and present the KITML for learning a kernel matrix $K = X^T AX$. Under this formulation, we only need to estimate $n \times n$ parameters in the matrix K, which is much smaller than $m \times m$ parameters in the original A matrix. The distance between two points based on K can be denoted as

$$d_A(\mathbf{x_i}, \mathbf{x_j})^2 = (\mathbf{e}_i - \mathbf{e}_j)^T K(\mathbf{e}_i - \mathbf{e}_j), \tag{9}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the unit basis vectors in which only the entry i or j is 1 and the rest are 0.

The optimization problem is to search for K that satisfies the similar/dissimilar side constraints as well as minimizing the KL divergence. Similarly, A_0 is transformed to kernelized $K_0 = X^T A_0 X$ for the regularization distribution.

$$\begin{aligned} \min_A \quad & d(K_0||K) \qquad\qquad (10)\\ \text{s.t.} \quad & (\mathbf{e}_i - \mathbf{e}_j)^T K(\mathbf{e}_i - \mathbf{e}_j) \leq u, (i, j) \in S,\\ & (\mathbf{e}_i - \mathbf{e}_j)^T K(\mathbf{e}_i - \mathbf{e}_j) \geq l, (i, j) \in D,\\ & K \succeq 0. \end{aligned}$$

The hyperparameters, upper bound u and lower bound l, are determined by the distribution of the data. In this article, l and u are the 5th and 95th percentiles of the

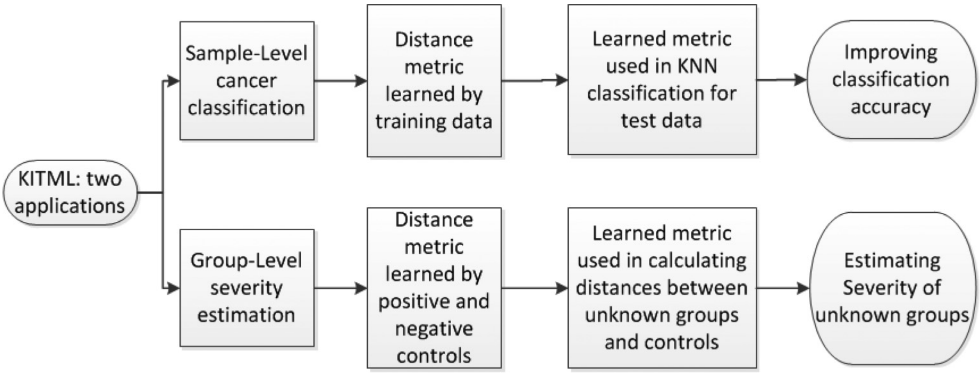
KITML: two applications
Sample-Level cancer classification
Distance metric learned by training data
Learned metric used in KNN classification for test data
Improving classification accuracy
Group-Level severity estimation
Distance metric learned by positive and negative controls
Learned metric used in calculating distances between unknown groups and controls
Estimating Severity of unknown groups

Fig. 1. A flow chart for KITML in two applications for cancer diagnosis using high-dimensional molecular profiling data.

observed distribution of distances between pairs of samples within each dataset. The optimization is again solved through Bregman projections [Bregman 1967], where in each iteration, a constraint $(i, j) \in S$ or $(i, j) \in D$ is picked to update the matrix K. The Bregman projection update is similar to Equation (8) and can be denoted as

$$K_{t+1} = K_t + \beta K_t(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T K_t, \tag{11}$$

where $K_0 = X^T A_0 X = X^T \mathbb{I} X = X^T X$ and β is the same as that in Equation (8).

2.5. Calculating Distance Using KITML on Samples with High-Dimensional Features

The kernel matrix K learned above is about the n training samples. During the testing phase, we need to calculate distances among points that might not be covered by the kernel K, thus, we cannot use Equation (9) directly. Through derivation and a theorem from Kulis [2013], we can conclude that A can be constructed in a closed form from K as following

$$A = \alpha I + XTX^T, \tag{12}$$

$$\text{where } \; T = K_0^{-1}(K - \alpha K_0)K_0^{-1}. \tag{13}$$

Here, α is suggested to take the value 1 in Kulis [2013] and $K_0 = X^T X$ since we use a Euclidean distance as prior distance function. Therefore, we can calculate the distance between any two "high-dimensional" sample points using an *implicit* representation of A through kernel evaluation, as follows:

$$\begin{aligned} d_A(\mathbf{x}_i^?, \mathbf{x}_j^?)^2 &= (\mathbf{x}_i^? - \mathbf{x}_j^?)^T A(\mathbf{x}_i^? - \mathbf{x}_j^?) \\ &= (\mathbf{x}_i^? - \mathbf{x}_j^?)^T (I + XTX^T)(\mathbf{x}_i^? - \mathbf{x}_j^?) \\ &= (\mathbf{x}_i^? - \mathbf{x}_j^?)^T (\mathbf{x}_i^? - \mathbf{x}_j^?) + (\mathbf{x}_i^? - \mathbf{x}_j^?)^T XTX^T (\mathbf{x}_i^? - \mathbf{x}_j^?) \\ &= (\mathbf{x}_i^? - \mathbf{x}_j^?)^T (\mathbf{x}_i^? - \mathbf{x}_j^?) + (\mathbf{x}_i^? - \mathbf{x}_j^?)^T X(K_0^{-1}(K - \alpha K_0)K_0^{-1})X^T (\mathbf{x}_i^? - \mathbf{x}_j^?) \end{aligned} \tag{14}$$

where matrix X with size $m * n$ describes the training set, $\mathbf{x}_i^?$ and $\mathbf{x}_j^?$ are two testing samples whose distance is under interest. Clearly instead of learning m^2 parameters in A, only n^2 parameters need to be learned by using the kernel formulation K.

Figure 1 provides a flow chart about how we adapt KITML on two applications for cancer diagnosis using high-dimensional molecular profiling data.

As mentioned in Section 1, one important emerging medical applications for molecular profiling technologies such as microarray or RNA-Sequencing is in clinical decision support systems. Possible tasks include (but are not limit to) diagnosing the existence of certain cancers, estimating the severity levels or cancer stages of patient samples, estimating the degree of effectiveness from certain treatments, or searching for similar patient cases for a target patient. All of these tasks may be handled through NN-based learning strategies and are very easy to interpret (since biomed-field favors models that are straightforward and easy to understand). For such NN-based systems, learning effective distance metrics is, therefore, critical for providing useful tools to physicians and researchers to help them design relevant prognosis or treatment plans. Owing to the availability of current relevant datasets, this article just shows two sample cases of adapting KITML on cancer diagnosis applications using molecular signature features. Considering the rapidly growing biomedical data, the proposed computational framework will grow to become more crucial, especially for modern healthcare systems where searching or retrieving similar patients, disease, symptoms or treatments are key functions.

On "high-dimensional" ($m >> n$) samples, KITML provides several major computational advantages over ITML and many other non-kernelized metric learning methods.

(1) *Parameter Storage*: KITML only needs to learn and store $O(n^2)$ parameters, which is much smaller than the $O(m^2)$ parameters required by ITML metric learning.
(2) *Optimization Complexity*: Assuming the max-number of side constraints is c, Equation (11) repeatedly computes Bregman projections of the current solution onto a single constraint, which means optimizing Equation (11) for K will be run c times. Therefore, the time complexity for KITML optimization is $O(cn^3)$ (matrix computation $O(n^3)$), while it is $O(cm^3)$ for ITML (Equation (8)).

In summary, KITML permits efficient optimization and lower storage need in learning through Equation (11). Equations (13) and (14) make the evaluation of the learned distance metric (i.e., calculating distances) efficient as well.

2.6. Sample-Level Cancer Classification with KNN KITML

K-Nearest Neighbor (KNN) classification is a natural candidate for using the distance metric we have just learned from KITML for predicting the existence of certain cancers in patent samples. For high-dimensional molecular signature data, when using metrics like Euclidean distance, KNN is often inferior to more sophisticated approaches such as Support Vector Machines (SVMs). In this article, we use KITML to actively learn a distance metric to improve the performance of KNN-driven cancer classification. KITML also reduces the heavy computation burden of distance metric learning through kernelization. The process works as following:

(1) use cross validation to find the optimal number of NNs, k, based on training samples;
(2) compute the distances of a test sample $\mathbf{x}_i$ to the labeled training samples $\mathbf{y}_i$ using Equation (14);
(3) order the training samples by increasing distances from the test sample;
(4) use majority vote or inverse distance weighted average based on KNNs to determine the class of the test sample $\mathbf{x}_i$.

2.7. Group-Level Severity/Stage Estimation with Set-Ranking KITML

Another important task for molecular-profiling-based cancer diagnosis is the ability to further quantify/classify blood or tumor samples into subtypes, which have distinct biomedical properties and result in varied prognoses. For instance, samples of "blood

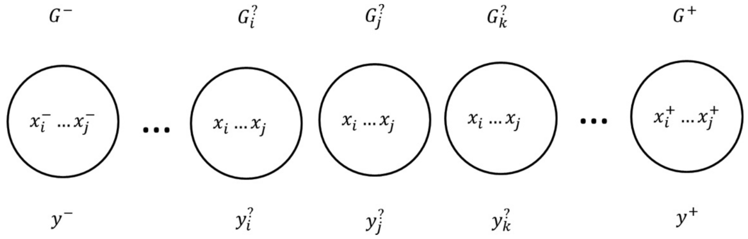

Fig. 2. Problem definition: the severity levels of positive control group $\mathbf{G}^+$ and negative control group $\mathbf{G}^-$ are known. The severity level $y_i^?$ of an unknown sample group $\mathbf{G}_i^?$ is estimated based on its learned distances to the two controls.

cancers"—Diffuse Large B-Cell Lymphomas (DLBCLs)—are indistinguishable based on histological methods yet are clinically heterogeneous: some patients respond well and exhibit prolonged survival while others do not [Alizadeh et al. 2000]. It has been shown that utilizing expression profiling techniques to stratify DLBCL to two subtypes is necessary [Alizadeh et al. 2000]. Researchers have developed expert-based diagnostic scores for tracking disease states and predicting clinical outcomes [Birkner et al. 2007]. However, the process is time consuming and expensive [Birkner et al. 2007]. For most cases of disease severity/stage estimation in practice, the reference data normally include only positive (e.g., most severe disease state) and negative controls (e.g., least severe disease state) since in many experiments such as blood assay or clinical trials only positive and negative controls were labeled to verify the success of their experiments.

Therefore, we feel it is important to design more advanced computational methods for categorizing subtypes of cancer samples using molecular expression data. Here, we propose a set-based ranking method using metrics learned from KITML for severity estimation. Normally given a dataset with multiple sample groups associated with different severity levels of a type of cancer, we assume that the positive and negative control groups' severity levels are known. The goal is to estimate the severity levels of unknown sample groups based on their relationship to the known control groups.

The schematic definition of set-KITML is illustrated in Figure 2. When analyzing a set of sample groups corresponding to a range of severity levels, it is natural to think that one can calculate the distances between samples with unknown severity to samples with known severity, in order to estimate the unknown severity. The basic idea of set-based KITML is maximizing the distances between dissimilar sample groups, and minimizing the distances between samples in the same group or among similar groups. The learned metric based on positive control and negative control should give a maximum distance $d(G^+, G^-)$ between these two controls. The distances, between a sample group $G^?$ with unknown severity level and two controls, can then be measured using this learned distance. These distances should be proportional to $d(G^+, G^-)$ and can be combined to locate the position of the unknown sample group between the two controls, where the position indicates the severity level.

In a high-dimensional setting, using the parameter T learned from Equation (13), we can calculate the distance measure between any data samples. Therefore, we define and calculate so the distances between an unknown severity sample $\mathbf{x}_i^?$ (within $\mathbf{G}^?$) to $\mathbf{G}^+$, and to $\mathbf{G}^-$, distance between $\mathbf{x}_i^?$ and $\mathbf{G}^+$ is defined as

$$d_A\big(\mathbf{x}_i^?, \mathbf{G}^+\big)^2 = \left(\frac{\sum_{x_k^+ \in \mathbf{G}^+} \mathbf{x}_k^+}{|\mathbf{G}^+|} - \mathbf{x}_i^?\right)^T (I + XTX^T)\left(\frac{\sum_{x_k^+ \in \mathbf{G}^+} \mathbf{x}_k^+}{|\mathbf{G}^+|} - \mathbf{x}_i^?\right). \tag{15}$$

Similarly, the distance between $\mathbf{x}_i^?$ and $\mathbf{G}^-$ is defined as

$$d_A\big(\mathbf{x}_i^?, \mathbf{G}^-\big)^2 = \left(\frac{\sum_{x_k^- \in \mathbf{G}^-} \mathbf{x}_k^-}{|\mathbf{G}^-|} - \mathbf{x}_i^?\right)^T (I + XTX^T) \left(\frac{\sum_{x_k^- \in \mathbf{G}^-} \mathbf{x}_k^-}{|\mathbf{G}^-|} - \mathbf{x}_i^?\right). \tag{16}$$

These two distances are then used to determine the predicted severity level $y_{x_i}^?$ of $\mathbf{x}_i^?$ (Equation (17)). When $y_{x_i}^?$ is close to 0, the severity of $\mathbf{x}_i^?$ is similar to that of the negative controls. On the other hand, if $y_{x_i}^?$ is close to 1, the severity of $\mathbf{x}_i^?$ is similar to that of the positive controls.

$$y_{x_i}^? = \frac{d_A\big(\mathbf{x}_i^?, \mathbf{G}^-\big)}{\big(d_A\big(\mathbf{x}_i^?, \mathbf{G}^+\big) + d_A\big(\mathbf{x}_i^?, \mathbf{G}^-\big)\big)}. \tag{17}$$

The severity $y^?$ of $\mathbf{G}^?$ is then defined as

$$y^? = \frac{\sum_{\mathbf{x}_i^? \in \mathbf{G}^?} y_{x_i}^?}{|\mathbf{G}^?|}. \tag{18}$$

2.7.1. Using Mean Points in Set-KITML. In general, measuring the distance between a sample $\mathbf{x}^?$ and a sample group G may be defined by one of the following methods:

(1) the distance between the mean point in the group to this sample: $d_A(\mathbf{x}^?, \mathbb{E}[G])$;
(2) the distance between the closest point in the group to this sample: $\operatorname{argmin}_{\mathbf{z} \in G} d_A(\mathbf{x}^?, \mathbf{z})$;
(3) the distance between the farthest point in the group to this sample: $\operatorname{argmax}_{\mathbf{z} \in G} d_A(\mathbf{x}^?, \mathbf{z})$.

Essentially, Equations (15) and (16) take the above option (1), which uses the distance between the sample and the mean point of the group.

This is because we assume that data samples in the same group follow the same distribution. Under this assumption, the Multidimensional Chebyshev's inequality [Durrett 2010] holds as

$$\mathbb{P}\big(\sqrt{(Z-\mu)^T V^{-1} (Z-\mu)} > t\big) \leq \frac{Q}{t^2}, \tag{19}$$

where Z is an m-dimensional random variable with the expected value $\mu = \mathbb{E}[Z]$ and covariance matrix $V = Cov(Z) = \mathbb{E}[(Z-\mu)(Z-\mu)^T]$. Here, $t > 0$ is a given number and $Q = \text{trace}(V^{-1}V)$. Chebyshev's inequality indicates that most data samples will be centralized around the mean point μ under the Mahalanobis distance measure $d_{V^{-1}}(x, y) = \sqrt{(x-y)^T V^{-1} (x-y)}$. Under our assumption that samples in the same group follow the same probability distribution, it is natural to choose the mean point μ for measuring the distance between a group and another sample, i.e.,

$$d_A\big(\mathbf{x}_i^?, \mathbf{G}^+\big)^2 = d_A\big(\mathbf{x}_i^?, \mathbb{E}[\mathbf{G}^+]\big)^2 = d_A\left(\mathbf{x}_i^?, \frac{\sum_{x_k^+ \in \mathbf{G}^+} \mathbf{x}_k^+}{|\mathbf{G}^+|}\right). \tag{20}$$

Combining Equations (20) and (14) results in Equation (15), and similarly Equation (16). Then, we can calculate the predicted severity level $y_{x_i}^?$ of sample $\mathbf{x}_i^?$ using Equation (17).

Assuming $Y^?$ represents a random variable, which represents the severity level of group $G^?$ when corresponding to severity levels of G^+ and G^-, then using the Law of Large Numbers [Durrett 2010], we choose the following unbiased estimator as the predicted severity level for a sample group $G^?$ (Equation (18)):

$$y^? = \mathbb{E}[Y^?] = \frac{\sum_{\mathbf{x}_i^? \in \mathbf{G}^?} y^?_{x_i}}{|\mathbf{G}^?|}. \tag{21}$$

3. CONNECTING TO RELATED STUDIES

Our proposed work is closely related (but not limited) to the following research disciplines.

3.1. Distance Metric Learning

Distance metric learning methods have caught much attention in the recent literature and are summarized in a series of recent surveys [Yang and Jin 2006; Kulis 2013; Bellet et al. 2013]. As summarized by the survey [Kulis 2013], metric learning has been extensively applied on applications like enhancing bag-of-words models, automated image tagging, image retrieval or face identification. Targeting the rapidly growing biomedical datasets, this article is the first attempt (to the authors' knowledge) to adapt metric learning framework on high-dimensional bio-data analysis. Since functions like searching or retrieving similar patients, diseases or treatments are key to modern healthcare systems, the proposed framework potentially will become more and more important.

Since there exist a number of papers about metric learning, we will cover just a few representative methods, such as ITML and large margin nearest neighbor (LMNN) [Weinberger and Saul 2009]. Mahalanobis metric learning can be seen as learning a linear transformation followed by the calculation of the Euclidean metric in the transformed space. Such learning methods normally vary by the regularization functions they use. For example, LMNN uses the regularization function $tr(AU)$ where $U = \sum_{(x_i,x_j)\in S}(x_i - x_j)(x_i - x_j)^T$. This may result in a low-rank A matrix. As another example, ITML uses a log-determinant based regularization function $tr(A) - logdet(A)$ to constrain A to be strictly positive definite. Accordingly, we uses LMNN and ITML as baselines to compare with KITML in our experiments.

Under the "high-dimensional" setting, KITML essentially assumes the original matrix A has a low rank plus a diagonal matrix structure through Equation (12). The baseline ITML method itself will also generate a low rank plus diagonal matrix structure under the high-dimensional situation. This is because that at each update step of Bregman's projection (Equation (8))

$$\text{Rank}(A_t(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T A_t) \leq \min\{\text{Rank}(A_t), \text{Rank}((\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T)\}, \tag{22}$$

which means that $\text{Rank}(A_{t+1}) \leq \text{Rank}(A_t) + 1$. The maximum number of iterations has an exact upper bound as n^2 so we can conclude that $\text{Rank}(A) < n^2$. Considering that $n \ll d$ and that diagonal matrix $A_0 = \mathbb{I}$, we can see that ITML also learns the parameter matrix A as a low rank plus diagonal structure under a high-dimensional setting.

Other kernel-based metric learning methods have been proposed, as well. Wang et al. [2013] generalized LMNN and ITML into a kernel classification framework through explicit polynomial kernel functions on doublets and triplets from the training samples. Hertz et al. [2004] utilized a boosting-style strategy "DistBoost" to learn distance functions over the product space of sample pairs with a weak learner based on partitioning the original feature space. We tried to use this method in our experiments, but abandoned it due to its slow speed. Nguyen and Guo [2008] explored local neighborhood constraints to address the metric learning problem through a margin-based approach and the learning is formulated as a quadratic semi-definite programming problem (QSDP). Wang et al. [2011] proposed a method based on the multiple kernel learning strategy under the metric learning setting by learning a linear combination of a number of predefined kernels.

A number of recent studies consider sparsity, low-rank, robustness or structured constraints in metric learning. For instance, Ying et al. [2009] proposed a mixed-norm regularized metric learning algorithm to learn a low-dimensional (sparse) distance matrix. Later, Ying and Li [2012] developed an eigenvalue optimization framework for learning a Mahalanobis metric. As a prior work of KITML, Davis and Dhillon [2008] proposed a structured metric learning method based on the log-determinant matrix divergence, which enables efficient optimization of structured, low-parameter Mahalanobis distances. Very recently, Wang et al. [2014] proposed a new objective for distance metric learning using the L1-norm distances. This robust distance metric learning strategy is solved through simultaneous L1-norm minimization and maximization. Furthermore, Zhang et al. [2010] proposed a general kernelization framework for learning algorithms via a two-stage procedure, that is, transforming data by kernel principal component analysis (PCA), and then, directly performing the learning algorithm with the transformed data. Enlightened by this framework, we also include one more baseline using PCA+KNN to compare the effectiveness of our proposed methods.

3.2. Learning with Side Constraints

Most metric learning algorithms aim to generalize the standard squared Euclidean distance, by optimizing a target distance under various types of side constraints. Similar ideas of learning from side information have also been explored in a number of other important machine learning topics. For example, the classic "Semi-Supervised Clustering" [Basu et al. 2004] uses the pairwise constraints, that is, pairs of instances labeled as belonging to the same or different clusters, to improve unsupervised clustering. Others studies [Xiang et al. 2008] have performed better clustering or classification by considering pairwise constraints in the form of must-links and cannot-links. In their formulation, a must-link indicates the two data points must be in the same class, while a cannot-link indicates that the two data points must be in different classes. We similarly construct pairwise constraints in **S** (Similar) and **D** (Dissimilar) sets. Unlike [Xiang et al. 2008], which focused on the clustering or classification of samples, our framework performs efficient metric learning on a high-dimensional feature space using kernelization.

The idea of "learning from side constraints" has been explored within the "deep learning" community as well. A Dimensionality Reduction by Learning an Invariant Mapping (DRLIM) [Hadsell et al. 2006] approach learns a parametric mapping $A : \mathbf{x} \in \mathbb{R}^m \mapsto \mathbf{z} \in \mathbb{R}^D$, such that the embeddings of similar samples attract each other in the low-dimensional space, while the embeddings of dissimilar samples push each other away in the low-dimensional space. When using a margin-based loss, the learning of this embedding encourages similar examples to be close, and dissimilar ones have a distance of at least m from each other. The embedding representations learned from DRLIM can be used to calculate the distance measure between samples. The proposed KITML directly computes the distances without the need of embedding though.

3.3. Ordinal Regression

Our set-ranking KITML method is closely related to the task of "ordinal regression," which is a type of regression analysis predicting an ordinal variable. The value of an ordinal variable exists on an arbitrary scale where only the relative ordering between different values matters. There exists a large body of methods for ordinal regression in the literature (a thorough tutorial provided in Agresti [2010]). Similar to "ordinal regression," our problem of group-level severity estimation aims to figure out the relative biomedical severity among sample groups. However, since each of the sample group is a set of biomedical samples, our framework is more a set "ordinal regression" task. Due to the small number of groups normally appearing in the biomedical dataset, estimating regression parameters can rely on only a few number of group instances. Therefore, we argue that our ranking with metric learning framework provides a more robust solution for the target group-level estimation task.

3.4. Molecular-Expression-Based Cancer Diagnosis

Genome-scale monitoring technologies, especially microarray platforms, have enabled the measurement of thousands of molecular signatures in cancer cells. Generally, computational analyses on such datasets could be categorized into three types.

—Using machine learning classifiers on such data has been shown to improve the diagnosis of patients with cancer [Dudoit et al. 2002] and our KITML framework belongs to this category. Various studies [Ramaswamy et al. 2002; Perou et al. 2000] have shown convincing experimental results that gene expression data can provide more reliable means to diagnose and predict the existence of cancers than traditional clinical methods. Several studies have conducted comprehensive comparisons [Ramaswamy et al. 2002; Perou et al. 2000; Hudson et al. 2010] of popular learning classifiers for this task and SVM and random forest are the top ranked classifiers from such studies. Therefore, we use these two methods as baselines to compare with KITML in the experimental section of this article. In addition, decision tree is also a popular candidate for such patient diagnosis due to its interpretability [Hudson et al. 2010], so we include it in our analysis as well.

—The second category of studies focus on "biomarker" discovery when analyzing such molecular profiling datasets. For instance, the studies of differentially expressed genes have enabled the identification of single gene markers implicated in different types of cancers such as breast [Perou et al. 2000], lung cancer [Cancer Genome Atlas Research Network et al. 2012], and other cancers [Tusher et al. 2001]. This group of significance-test-based analyses may indicate possible gene targets for more detailed molecular studies or drug treatments. Considering data of "high-dimensionality," several studies [Leng 2008; Lu et al. 2010; Min et al. 2013] explored sparse-learning-based frameworks to detect gene or gene groups that contribute to human disease, by enforcing sparsity at the feature or feature group level in a supervised regression framework.

—A third type of analysis is to perform unsupervised clustering, especially the hierarchical clustering algorithms [Neve et al. 2006; de Souto et al. 2008] on cancer/patient samples (tissues). The goal is to find groups of samples that share similar expression patterns that can lead to identification cancer subtypes. Differently, our set-based KITML framework focuses on detect the severity-level/stage of the patient sample groups.

In addition, one previous article [Xiong and Chen 2006] from bioinformatics literature claimed to propose "Kernel-based distance metric learning for microarray data classification." Different from our parameterization with matrix K (Equation (11)) (i.e., we are learning K), this article aimed to learn weights for combining multiple handcrafted kernel matrices, which is not comparable to our method.

4. EXPERIMENTAL SETUP

4.1. Sample-Level Cancer Classification Experimental Setup

4.1.1. Algorithms Compared. We compared KITML performance with following algorithms:

(1) *KNN Classification with Distance Metric Learned by ITML*.* Directly learning distance metric from high-dimensional dataset by ITML is quite slow. Thus, for each dataset, we first use a variance feature selection to obtain a reduced feature set of size 100. The metric learning process and KNN classification are based on these 100 features with highest variance in the dataset. We denote the combination of feature selection and ITML as ITML*.
(2) *KNN Classification With PCA.* Dimensionality reduction was conducted by a linear transformation learned by PCA using the first l principal components that accounted for more than 90% of the variance of the data. The KNN classification is applied to the transformed data using Euclidean distance as distance metric.
(3) *KNN Classification with Euclidean Distance.* Here, we use Euclidean distance as a baseline and show that KITML can improve the basic KNN classification.
(4) *Multi-Class SVM With Linear Kernel.* SVM often achieves superior classification performance compared to other learning algorithms across most domains and tasks [Statnikov et al. 2005]. In this article, we compare SVMs implemented by libsvm v3.18 [Chang and Lin 2011] using a linear kernel $K(x, y) = x^T y$, where x and y are samples.
(5) *Multi-Class SVM With Radial Basis Function (RBF) Kernel.* In addition, we compare SVM using an RBF kernel $K(x, y) = \exp(-\gamma|x-y|^2)$, where x and y are samples and γ is kernel parameter.
(6) *LMNN Classification.* LMNN [Weinberger and Saul 2009] is a popular metric learning algorithm that learns a Mahalanobis distance metric for KNN classification from labeled examples.
(7) *Decision Tree Classification (DTC).* The C4.5 DTC algorithm [Quinlan 1993] implemented in the Weka 3.6.6 software [Hall et al. 2009] (University of Waikato, Hamilton, New Zealand) was used here to analyze the data.
(8) *Random Forest.* Random forest is an ensemble learning method for classification and regression using multiple decision tree models. We used the Random Forest algorithm implemented in the Weka 3.6.6 software [Hall et al. 2009] (University of Waikato, Hamilton, New Zealand) as one baseline to classify the data.

4.1.2. High-Dimensional Microarray Datasets. Fourteen publicly available microarray datasets shown in Table I are used to evaluate our sample-based KITML approach. These datasets were obtained using two microarray technologies: single-channel Affymetric chips (six sets) and double-channel cDNA chips(eight sets). For each dataset, total number of samples, number of features, number of classes, number of samples in each class, type of microarray chip, and tissue type are presented in Table I. In summary, the 14 datasets had 2–10 distinct classes, 28–248 samples, and 1,095–4,553 features.

4.1.3. Experimental Setup. The experimental setup is designed to obtain reliable performance estimates and avoid over-fitting using two loops. The inner loop is used to determine the best parameters of the classifier using cross-validation sets. The outer loop is used to estimate the performance of the classifiers built using the parameters found by the inner loop. The test datasets used in the outer loop are independent from cross-validation sets. The outer loop uses a 10-fold cross validation and the inner loop uses a 4-fold cross validation. We run each of the 14 datasets through both our KITML and the test algorithms baseline five times and average the classification results.

Table I. Dataset Description for Sample-Level Cancer Classification

Dataset Name	Total Samples	Num. of Features	Num. of Classes	Num. of Samples in Each Class	Tissue
Alizadeh et al. [2000]	42	1095	2	21, 21	Blood
Bittner et al. [2000]	38	2201	2	19, 19	Skin
Bredel et al. [2005]	50	1739	3	31, 14, 5	Brain
Garber et al. [2001]	66	4553	4	17, 40,4, 5	Lung
Golub et al. [1999]	72	1877	2	47, 25	Bone marrow
Golub et al. [1999]	72	1877	3	38, 9, 25	Bone marrow
Gordon et al. [2002]	181	1626	2	31, 150	Lung
Nutt et al. [2003]	28	1070	2	14,14	Brian
Pomeroy et al. [2002]	42	1379	5	10, 10, 10, 4, 8	Brian
Su et al. [2001]	174	1571	10	26, 8, 26, 23,12, 11, 7, 27, 6, 28	Multi-tissue
Tomlins-v1 [Tomlins et al. 2007]	104	2315	5	27, 20, 32, 13, 12	Prostate
Tomlins-v2 [Tomlins et al. 2007]	92	1288	4	27, 20, 32, 13	Prostate
Yeoh-v1 [Yeoh et al. 2002]	248	2526	2	43, 205	Bone marrow
Yeoh-v2 [Yeoh et al. 2002]	248	2526	6	15, 27, 64, 20, 79, 43	Bone marrow

4.1.4. KITML Setting. To construct constrained pairs (Section 2.1), we consider the pairs of samples in the same class are similar and pairs of samples in different classes are dissimilar. A total of $20C^2$ constrained pairs are randomly chose in the learning process, where C is the number of classes in each dataset. The lower and upper bounds of the right-hand side of the constraints (l and u) in Equation (10) are the 5th and 95th percentiles of the observed distribution of distances between pairs of samples within each dataset.

4.1.5. Performance Metrics. We use two classification performance metrics. The first performance metric is Macro-averaged F1 (F-measure). The F-measure is a weighted combination of precision and recall. It is defined as

$$F = \frac{(\beta^2 + 1)P_{\text{macro}}R_{\text{macro}}}{\beta^2 P_{\text{macro}} + R_{\text{macro}}}, \tag{23}$$

where β is typically set to 1. The multi-class precision and recall are defined as

$$P_{\text{macro}} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FP_i}, \tag{24}$$

$$R_{\text{macro}} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FN_i}, \tag{25}$$

where TP_i is the number of true positives for class i, FP_i is the number of false positives for class i, FN_i is the number of false negatives for class i, and C is the number of classes.

Receiver operating characteristics (ROC) graphs are useful for organizing classifiers and visualizing their performance [Fawcett 2006]. To compare classifiers, a common method is to calculate the area under the ROC curve, abbreviated AUC. Therefore, the second metric we use is multi-class AUC in the R package "pROC" [Robin et al. 2011].

Table II. Dataset Description for Estimating Severity of Sample Subgroups

Dataset Name	Total Samples	Num. of Features	Num. of Features after Pre-processing	Staging	Num. of Sample in Each Stage
Bjladder [Dyrskjot et al. 2003]	40	7129	3036	Ta, T1, T2+	20, 11, 9
Prostate [True et al. 2006]	31	15488	9491	Gleason patterns 3, 4, 5	11, 12, 9
Ovary [Wu et al. 2007]	37	22283	18091	T1, T2, T3	18, 5, 14

A multi-class AUC is defined as an average AUC [Hand and Till 2001]

$$AUC = \frac{2}{|C|(|C|-1)} \sum_{i=1}^{|C|} auc_i \tag{26}$$

where C is the number of the classes.

4.1.6. Statistical Comparison Among Classifiers. Statistical comparison is used to verify that the differences in accuracy between algorithms are non-random. Since we have only 14 datasets, we cannot assume that the difference between results are normally distributed [Demšar 2006], we used Wilcoxon signed-rank test [Wilcoxon 1945], which is a non-parametric alternative to paired t-test. Wilcoxon signed-rank test ranks the difference in performance of two classifiers for each dataset, ignoring the signs and compares the ranks for positive and negative differences.

4.2. Experimental Setup for Estimating Stage/Severity for Sample Subgroups

4.2.1. High-Dimensional Microarray Datasets. Three microarray datasets from bladder, prostate, and ovarian multi-stage cancer patient studies (Table II) are used here [Chen 2012]. (1) The bladder dataset contains gene expression data of human bladder tumor samples from a clinical specimen bank. There are 20 Ta (stage 1) samples, 11 T1 (stage 2) samples, and 9 T2+ (stage 3) samples, which contain a total of 7,129 genes. After pre-processing according to Chen [2012], we removed genes having missing data, leaving 3,036 genes for our analysis. (2) The prostate cancer dataset was created in an attempt to characterize gene expression profiles of specific Gleason patterns. The dataset contains gene expression data of 11 Gleason pattern three (stage 1) samples, 12 Gleason pattern four (stage 2) samples, and 8 Gleason pattern five (stage 3) samples. After removing the data with missing values, there are 9,491 genes left for our analysis. (3) The ovary dataset is from genetically engineered mouse models, which are used to demonstrate the mutations of certain signaling pathways in woman and mouse ovarian endometrioid adenocarcinomas [Wu et al. 2007]. There are 18 T1 (stage 1) samples, 5 T2 (stage 2) samples and 14 T3 (stage 3) samples. After pre-processing, there are 18,091 genes left for our analysis.

4.2.2. KITML Setting. Since there are three stages in each dataset, 75% of the stage 1 samples are used as negative control $\mathbf{G}^{-}$ and 75% of the stage 3 samples are used as positive control $\mathbf{G}^{+}$. These controls are used to learn distance metric parameters. The remaining 25% of both stage 1 samples $\mathbf{G}_1^{?}$ and stage 3 samples $\mathbf{G}_3^{?}$, and all of the stage 2 samples $\mathbf{G}_2^{?}$, are then used as test groups to evaluate the learned metric parameters. This process is repeated 4 times and averaged. For all three datasets, the constrained pairs used are formulated by the samples within negative controls $\mathbf{G}^{-}$ and positive controls $\mathbf{G}^{+}$. The lower and upper bounds of the right-hand side of the constraint (l and u) in Equation (10) are the 5th and 95th percentiles of the observed distribution of distances between pairs of samples within each dataset.

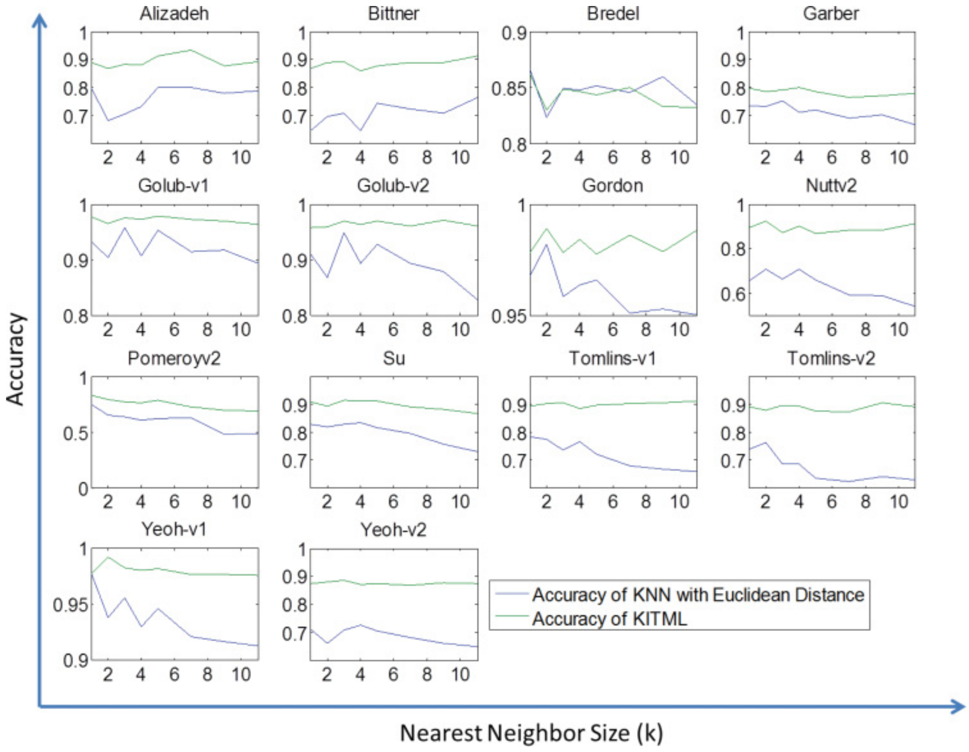

Fig. 3. Comparing cross-validation classification accuracy when varying neighbor size k.

5. EXPERIMENTAL RESULTS

5.1. Sample-Level Cancer Classification Results

5.1.1. Cross-Validation Accuracy When Varying Neighbor Sizes in KNN. In KNN classification, NN size k is a user-defined parameter and the choice of k is very critical to the classification performance. The optimal value of k is based on the cross-validation accuracy of training samples. Generally, the larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct [Cover and Hart 1967]. Figure 3 shows the cross-validation accuracy of KNN KITML and KNN with Euclidean distance as a function of different neighbor sizes (from $k = 1$ to 11) for all 14 datasets listed in Table I. The purpose of this analysis is finding the k value that leads to the highest classification accuracy for training data. Then, when testing KNN KITML and KNN with Euclidean distance, this optimal value of k will be used in the classification. For example, for Alizadeh dataset, when $k = 7$ KNN KITML has the highest cross-validation accuracy and when $k = 5$ KNN with Euclidean distance has the highest cross-validation accuracy. Moreover, from Figure 3, we can see that the cross-validation accuracy of KNN KITML does not have an obvious trend when k gets larger. For KNN with Euclidean Distance, it can be seen that except Alizadel, Bitterner, and Bredel datasets, the cross-validation accuracy decreases when k gets larger.

5.1.2. Overall Accuracy and Marco-Average F1. The results of the algorithm comparison with macro-average F1 as the performance metrics are shown in Table III. KNN KITML has the best average performance among all nine classification algorithms for macro-averaged F1. Specifically, KNN KITML generated best macro-average F1 in 9 out of 14 datasets (Table III). For Nutt, Tomlins-v2, and Yeoh-v1 datasets, LMNN outperformed

Table III. Classification Algorithm Comparison with Macro-Averaged F1

Dataset	KNN KITML	KNN ITML*	KNN PCA	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	**0.9386±0.0316**	0.8762±0.0199	0.7958±0.0554	0.8196±0.0437	0.9240±0.0309	0.9242±0.0260	0.7357±0.0513	0.7042±0.0507	0.9296±0.0351
Bittner	0.9174±0.0219	**0.9371±0.0301**	0.7339± 0.021	0.7766±0.0627	0.8183±0.0253	0.8199±0.0692	0.5882±0.0686	0.6215±0.1011	0.8300±0.055
Bredel	**0.7707±0.0288**	0.7442±0.0546	0.7247±0.0217	0.7138±0.0352	0.7254±0.0153	0.7238±0.0168	0.4062±0.1466	0.4640±0.0706	0.7560±0.0195
Garber	0.6118±0.0156	0.6005±0.0425	0.5873± 0.0444	0.5246±0.0301	0.6062±0.0421	0.5796±0.0411	0.5682±0.1329	**0.6819±0.0933**	0.5823±0.0386
Golub-v1	**0.9849±0.0185**	0.9637±0.0107	0.9639±0.0162	0.9697±0.0000	0.9758±0.0134	0.9697±0.0106	0.8478±0.0109	0.8609±0.0274	0.9848±0.0107
Golub-v2	**0.9662±0.0076**	0.9589±0.0078	0.9554±0.0229	0.9472±0.01587	0.9057±0.0167	0.9047±0.0135	0.8256±0.0447	0.8153±0.0247	0.9637±0.0094
Gordon	**0.9863±0.0053**	0.9823±0.0147	0.9840±0.0006	0.9705±0.0000	0.9824±0.0082	0.9824±0.0044	0.8850±0.0068	0.8888±0.0309	0.9755±0.007
Nutt	0.9402±0.0347	0.9086±0.0253	0.7376±0.0000	0.8517±0.0324	0.9207±0.0169	0.8714±0.0319	0.8309±0.0457	0.7934±0.0951	**0.9655±0.0065**
Pomeroy	**0.8451±0.0306**	0.8031±0.0556	0.7814±0.0163	0.7241±0.0379	0.7708±0.0287	0.7698±0.0140	0.2663±0.0390	0.2716±0.0250	0.7621±0.0285
Su	**0.9123±0.0100**	0.8351±0.0169	0.8325±0.0129	0.8165±0.0048	0.8733±0.0041	0.8640±0.0162	0.3995±0.0642	0.4718±0.0297	0.8326±0.0046
Tomlins-v1	**0.9205±0.0186**	0.8768±0.0220	0.7891±0.0323	0.8089±0.0085	0.9193±0.0192	0.9065±0.0202	0.5523±0.0317	0.5198±0.0685	0.8770±0.0286
Tomlins-v2	0.9026±0.0232	0.8801±0.0256	0.7477±0.0178	0.7776±0.0153	0.8994±0.0045	0.8818±0.0186	0.5693±0.0564	0.5574±0.0341	**0.9215±0.0195**
Yeoh-v1	0.9887±0.0039	0.9516±0.0064	0.9453±0.0077	0.9601±0.0065	0.9715±0.0102	0.9599±0.0065	0.9877±0.0030	0.9864±0.0000	**0.9930±0.0256**
Yeoh-v2	**0.8411±0.0204**	0.7798±0.0071	0.7020±0.0133	0.6979±0.0082	0.8049±0.0128	0.8047±0.0111	0.4814±0.0146	0.5323±0.0714	0.8264±0.0237
Average	**0.8947**	0.8656	0.8094	0.8079	0.8641	0.8545	0.6389	0.6550	0.8714

KNN KITML. For the Garber dataset, Random Forest outperformed KNN KITML. KNN ITML* has the best macro-averaged F1 for the Bittner dataset. In general, KNN ITML*, SVM Linear, and SVM RBF have similar average classification performance. Our KNN KITML has around 3%–4% performance increase compared to these three algorithms. KNN PCA used a dimensionality reduction method (PCA) to reduce features before applying KNN and achieved comparable performance as the baseline classifier KNN Euclidean. But the performance gap between KNN PCA and KNN KITML remains large.

The multi-class AUC comparison results are shown in Table IV. KNN KITML has the highest average multi-class AUC and achieved the best performance in 7 out of 14 datasets. LMNN has comparable results to KNN KITML—it achieved the best performance in 7 out of 14 datasets and its average multi-class AUC is only a little lower than KNN KITML. For the Golub-v2 dataset, KNN KITML, KNN ITML*, and LMNN have the same highest multi-class AUC. For the Su dataset, DTC has the highest multi-class AUC.

5.1.3. Wilcoxon Signed-Rank Test Results. Wilcoxon signed-ranks test is used to verify that the differences in accuracy between algorithms are non-random. The Wilcoxon signed-rank test ranks the difference in performance of two classifiers for each dataset, ignoring the sign, and compares the ranks for positive and negative differences. The results of right-sided Wilcoxon signed-ranks test are shown in Table V. The p-values of the tests for macro-averaged F1 between KITML and the other eight classification algorithms indicate that KNN KITML achieved better performance than all the other algorithms in terms of macro-average F1 at 5% significance level. Similarly, for multi-class AUC, the p-values also indicate KNN KITML is better than the other algorithms except LMNN. KNN KITML and LMNN have similar classification performance in terms of multi-class AUC. But in the next section, we show that KNN KITML is much faster than LMNN in the classification process.

5.1.4. Time Complexity Analysis. The objective of the learning process in KITML is to learn the n by n parameter matrix in the distance metric, where n is the number of samples. Therefore, for each constraint defined in Equation (10) (l or u), the time complexity is $O(n^3)$ for matrix multiplication. For the entire learning process looping through all the constraints, the time complexity is $O(cn^3)$, where c is the number of constraints. We further analyzed the execution times for the two best classification algorithms, KNN KITML and LMNN, and KNN ITML* was served as an execution time baseline. Our experimental setup was designed to obtain reliable performance estimates and avoid over-fitting using two loops. The inner loop is used to determine the best parameters of the classifier using cross-validation sets. The outer loop is used to estimate the performance of the classifiers built using the parameters found by the inner loop. In the execution time analysis, we only ran outer loop for each of the algorithms. For the inner loop, we used default parameters for each algorithm. Figure 4 shows the execution time analysis. KNN KITML requires much less time, taking 2–519 seconds to run each dataset, while for LMNN this typically exceeded 24 hours to finish calculation. KNN ITML* was served as an execution time baseline here and it took 96–9,416 seconds to run each dataset even when it only used 100 features. These execution time differences illustrate that our KNN KITML greatly reduced the computation load comparing to both LMNN and KNN ITML*. All experiments were executed in MATLAB 2012a software (The Mathworks, Natick, MA) on a Quad core Intel 3.5GHz PC.

5.2. Estimating Severity of Sample SubGoups Results

The estimated severity levels as determined by KITML on the three micro-array datasets from bladder, prostate, and ovarian multi-stage cancer patient studies

Table IV. Classification Algorithm Comparison with Multi-Class AUC

Dataset	KNN KITML	KNN ITML*	KNN PCA	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	**0.9524±0.0211**	0.8571±0.0321	0.8333±0.0150	0.7919±0.0102	0.9048±0.0239	0.8992±0.0192	0.7380±0.0067	0.7142±0.0014	0.9523± 0.0223
Bittner	**0.9211±0.0065**	0.8947±0.0307	0.8421±0.0165	0.7105±0.0089	0.8684±0.0023	0.8571±0.0186	0.6988±0.0105	0.6077±0.0089	0.8421±0.0012
Bredel	0.8129±0.0121	0.8000±0.0002	0.7516±0.0247	0.6871±0.0042	0.7839±0.0129	0.7512±0.0100	0.5035±0.0095	0.5763±0.0128	**0.8193±0.0078**
Garber	0.4625±0.0094	0.5612±0.0035	0.4875±0.0283	0.4625±0.0133	0.4625±0.0000	0.4875±0.0002	0.7556±0.0237	0.6391±0.0015	**0.5687±0.0016**
Golub-v1	**0.9865±0.0122**	0.9800±0.0246	0.9200±0.0026	0.9387±0.0045	0.9600±0.0000	0.9520±0.0112	0.8455±0.0068	0.8548±0.0042	0.9493±0.0052
Golub-v2	**0.9444±0.0102**	**0.9444±0.0045**	0.8889±0.0017	0.8889±0.0223	0.7778±0.0263	0.7922±0.0220	0.6562±0.0223	0.6641±0.0189	**0.9444±0.0155**
Gordon	**0.9838±0.0052**	0.9805±0.0037	0.9822±0.0002	0.9221±0.0315	0.9677±0.0082	0.9502±0.0196	0.9026±0.0127	0.9026±0.0049	0.9677±0.0089
Nutt	0.8928±0.0012	0.9285±0.0085	0.6785±0.0304	0.6421±0.0088	0.8928±0.0122	0.8815±0.0060	0.8099±0.0241	0.8918±0.0011	**0.9642±0.0047**
Pomeroy	0.9000±0.0000	0.8650±0.0207	0.8750±0.0263	0.8500±0.0000	0.9100±0.0312	0.9100±0.0000	0.5433±0.0082	0.5971±0.0175	**0.9500±0.0002**
Su	0.8774±0.0137	0.8512±0.0123	0.7355±0.0010	0.6514±0.0012	0.6322±0.0015	0.6552±0.0450	**1.0000±0.0000**	0.9381±0.0201	0.7692±0.0083
Tomlins-v1	0.9962±0.0060	0.9629±0.0034	0.8425±0.0032	0.8185±0.0038	0.9259±0.0164	0.8915±0.0109	0.8162±0.0104	0.7758±0.0023	**1.0000±0.0000**
Tomlins-v2	0.9988±0.0154	0.9027±0.0075	0.8222±0.0369	0.6925±0.0247	0.9148±0.0101	0.9022±0.0133	0.6814±0.0067	0.7943±0.0108	**1.0000±0.0000**
Yeoh-v1	**0.9883±0.0043**	0.9764±0.0003	0.9186±0.0103	0.9186±0.0112	0.9419±0.0187	0.9231±0.0020	0.9951±0.0062	0.9951±0.0037	0.9767±0.0012
Yeoh-v2	**0.9691±0.0017**	0.8098±0.0329	0.9518±0.0006	0.8024±0.0150	0.9333±0.0213	0.8981±0.0060	0.6050±0.0208	0.5617±0.0046	0.8259±0.0203
Average	**0.9061**	0.8796	0.8235	0.7698	0.8293	0.8414	0.7537	0.7509	0.8950

Table V. p-Values of Right-Sided Wilcoxon Signed-Rank Test between KNN KITML and the Other Eight Classification Algorithms

	KNN ITML*	KNN PCA	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
***p*-values for macro-averaged F1**	6.1e-04	6.1e-05	6.1e-05	6.1e-05	6.1e-05	6.1e-05	1.8e-04	0.02
***p*-values for multi-class AUC**	0.0471	3.6e-4	1.2e-04	4.8e-04	4.2e-04	0.0083	0.0034	0.2939

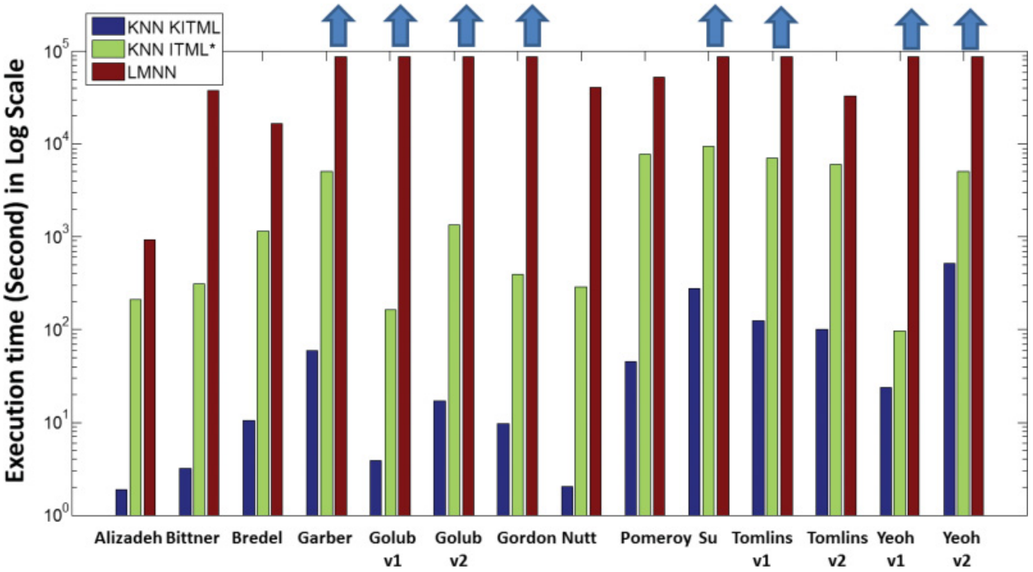

Fig. 4. Comparing execution time between KNN KITML, KNN ITML*, and LMNN for all 14 datasets. KNN KITML used the least execution time in every dataset and LMNN used the most time. The execution time of KNN ITML* fell in between KNN KITML and LMNN in every dataset. Since Garber, Golub-v1, Golub-v2, Gordon, Su, Tomlins-v1, Yeoh-v1, and Yeoh-v2 need more than 24 hours execution time for LMNN, we draw their bars using the same longest length in the figure.

(Table II) are shown in Figure 5. Each dataset contains samples from three different cancer stages, indicating three different severity levels. Each y_i is the mean of sample severity levels within $\mathbf{E}_i^?$. The standard deviation is shown as the error bar in the figure. We consider the relative order of $y_1 < y_2 < y_3$ as the correct severity estimation since it matches the true group labels and our KITML approach correctly estimated the relative severity levels of the three test groups. Notably, without any prior information about $\mathbf{E}_2^?$ in each dataset, our approach still can estimate the severity level y_2 of $\mathbf{E}_2^?$ in the right order—between y_1 and y_3.

6. CONCLUSION

Recent advancements in genome-wide molecular profiling technologies have eased the measurement of molecular signatures of cancer cells. In this article, we have two novel applications of the KITML approach for cancer diagnosis using distance metric learning. Our analyses have shown that KITML achieves favorable performance over other state-of-the-art methods. In the future, we plan to extend KITML approach to more biomedical applications and at the same time, to design strategies to reduce computational complexity.

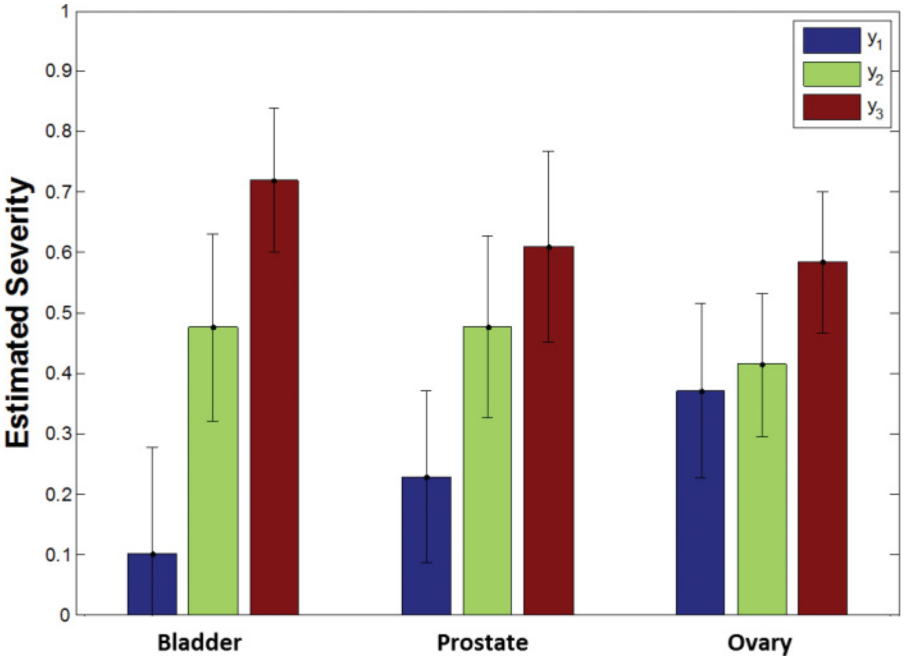

Fig. 5. Severity estimation results of three datasets.

A. APPENDIX

Proof. Assuming the means of the Gaussian distributions are 0, we have

$$
\begin{aligned}
d(A_0||A) &= KL(Pr(x|A_0)||Pr(x|A))\\
&= \int Pr(x|A_0)\log\frac{Pr(x|A_0)}{Pr(x|A)}d\mathbf{x}\\
&= \int [\log(Pr(x|A_0)) - log(Pr(x|A))]Pr(x|A_0)dx\\
&= \int \left[\frac{1}{2}\log\frac{|A|}{|A_0|} + \frac{1}{2}x^T A_0^{-1}x + \frac{1}{2}x^T A^{-1}x\right]Pr(x|A_0)dx\\
&= \frac{1}{2}\log\frac{|A|}{|A_0|} - \frac{1}{2}tr\left\{E(x^T x)A_0^{-1} + \frac{1}{2}E(x^T x)A^{-1}\right\}\\
&= \frac{1}{2}\log\frac{|A|}{|A_0|} - \frac{1}{2}tr\{I_m\} + \frac{1}{2}tr\{A^{-1}A_0\}\\
&= \frac{1}{2}(tr(A^{-1}A_0) + log|A| - log|A_0| - m)
\end{aligned}
$$

Supplement website: MATLAB implementation of the proposed framework is available at github.com/DataFusion4NetBio/Paper15-MetricLearn4CancerExpression.

REFERENCES

Alan Agresti. 2010. *Analysis of Ordinal Categorical Data*. Vol. 656. Wiley.

Ash A. Alizadeh, Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore S. Lossos, Andreas Rosenwald, et al. 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403, 6769, 503–511.

Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 59–68.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. 2013. A survey on metric learning for feature vectors and structured data. *arXiv Preprint arXiv:1306.6709* (2013).

M. D. Birkner, S. Kalantri, V. Solao, P. Badam, R. Joshi, A. Goel, M. Pai, and A. E. Hubbard. 2007. Creating diagnostic scores using data-adaptive regression: An application to prediction of 30-day mortality among stroke victims in a rural hospital in India. *Therapeutics and Clinical Risk Management* 3, 3, 475–484.

M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, et al. 2000. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature* 406, 6795, 536–540.

M. Bredel, C. Bredel, D. Juric, G. R. Harsh, H. Vogel, L. D. Recht, and B. I. Sikic. 2005. Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas. *Cancer Res.* 65, 19, 8679–8689.

L. M. Bregman. 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.* 7, 3, 200–217. DOI:http://dx.doi.org/10.1016/0041-5553(67)90040-7

Cancer Genome Atlas Research Network and others. 2011. Integrated genomic analyses of ovarian carcinoma. *Nature* 474, 7353, 609–615.

Cancer Genome Atlas Research Network and others. 2012. Comprehensive genomic characterization of squamous cell lung cancers. *Nature* 489, 7417, 519–525.

Cancer Genome Atlas Research Network and others. 2013. Integrated genomic characterization of endometrial carcinoma. *Nature* 497, 7447, 67–73.

Gavin C. Cawley and Nicola L. C. Talbot. 2006. Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics* 22, 19, 2348–2355.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, 27-1–27-27.

Chi-Kan Chen. 2012. The classification of cancer stage microarray data. *Comput. Methods Programs Biomed.* 108, 3, 1070–1077. DOI:http://dx.doi.org/10.1016/j.cmpb.2012.07.001

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory,* 13, 1 (Jan. 1967), 21–27. DOI:http://dx.doi.org/10.1109/TIT.1967.1053964

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*. ACM, New York, NY, USA, 209–216. DOI:http://dx.doi.org/10.1145/1273496.1273523

Jason V. Davis and Inderjit S. Dhillon. 2008. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 195–203.

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. *Information Theoretic Metric Learning*. University of Texas at Austin, Austin. Retrieved from http://www.cs.utexas.edu/users/pjain/itml/.

Marcilio C. P. de Souto, Ivan G. Costa, Daniel S. A. de Araujo, Teresa B. Ludermir, and Alexander Schliep. 2008. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics* 9, 1 (2008), 1–14. DOI:http://dx.doi.org/10.1186/1471-2105-9-497

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, (Dec. 2006), 1–30.

Sandrine Dudoit, Jane Fridlyand, and Terence P. Speed. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Statistical Assoc.* 97, 457 (2002), 77–87.

Rick Durrett. 2010. *Probability: Theory and Examples*. Cambridge University Press.

Lars Dyrskjot, Thomas Thykjaer, Mogens Kruhoffer, Jens Ledet Jensen, Niels Marcussen, Stephen Hamilton-Dutoit, Hans Wolf, and Torben F. Orntoft. 2003. Identifying distinct classes of bladder carcinoma using microarrays. *Nat. Genet.* 33, 1, 90–96. 10.1038/ng1061.

Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recogn. Lett.* 27, 8 (Jun. 2006), 861–874. DOI:http://dx.doi.org/10.1016/j.patrec.2005.10.010

Mitchell E. Garber, Olga G. Troyanskaya, Karsten Schluens, Simone Petersen, Zsuzsanna Thaesler, Manuela Pacyna-Gengelbach, et al. 2001. Diversity of gene expression in adenocarcinoma of the lung. *Proceedings of the National Academy of Sciences* 98, 24, 13784–13789. DOI:http://dx.doi.org/10.1073/pnas.241500798

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, et al. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 5439, 531–537.

Gavin J. Gordon, Roderick V. Jensen, Li-Li Hsiao, Steven R. Gullans, Joshua E. Blumenstock, Sridhar Ramaswamy, William G. Richards, David J. Sugarbaker, and Raphael Bueno. 2002. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res.* 62, 17, 4963–4967.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, 1735–1742.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. DOI: http://dx.doi.org/10.1145/1656274.1656278

David J. Hand and Robert J. Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* 45, 2, 171–186. DOI: http://dx.doi.org/10.1023/A:1010920819831

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2003. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (corrected ed.). Springer.

Tomer Hertz, Aharon Bar-Hillel, and Daphna Weinshall. 2004. Boosting margin based distance functions for clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML'04)*. ACM, New York, NY, USA, 50. DOI: http://dx.doi.org/10.1145/1015330.1015389

Thomas J. Hudson, Warwick Anderson, Axel Aretz, Anna D. Barker, Cindy Bell, Rosa R. Bernabé, M. K. Bhan, Fabien Calvo, Iiro Eerola, Daniela S. Gerhard et al. 2010. International network of cancer genome projects. *Nature* 464, 7291 (2010), 993–998.

Brian Kulis. 2013. Metric learning: a survey. *Found. Trends Mach. Learn.* 5, 4, 287–364. DOI: http://dx.doi.org/10.1561/2200000019

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Ann. Math. Stat.* 22, 1, pp. 79–86.

Chenlei Leng. 2008. Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. *Comput. Biol. Chem.* 32, 6, 417–425.

Shuya Lu, Jia Li, Chi Song, Kui Shen, and George C. Tseng. 2010. Biomarker detection in the integration of multiple multi-class genomic studies. *Bioinformatics* 26, 3, 333–340.

Martin Renqiang Min, Salim Chowdhury, Yanjun Qi, Alex Stewart, and Rachel Ostroff. 2013. An integrated approach to blood-based cancer diagnosis and biomarker discovery. *WORLD SCIENTIFIC*, Chapter 9, 87–98. DOI: http://dx.doi.org/10.1142/97898145832200009

Richard M. Neve, Koei Chin, Jane Fridlyand, Jennifer Yeh, Frederick L. Baehner, Tea Fevr, Laura Clark, Nora Bayani, Jean-Philippe Coppe, Frances Tong et al. 2006. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer Cell* 10, 6, 515–527.

Nam Nguyen and Yunsong Guo. 2008. Metric learning: a support vector approach. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 125–136.

Catherine L. Nutt, D. R. Mani, Rebecca A. Betensky, Pablo Tamayo, J. Gregory Cairncross, Christine Ladd, et al. 2003. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Res.* 63, 7, 1602–1607.

Charles M. Perou, Therese Sørlie, Michael B. Eisen, Matt van de Rijn, Stefanie S. Jeffrey, Christian A. Rees, Jonathan R. Pollack, Douglas T. Ross, Hilde Johnsen, Lars A. Akslen et al. 2000. Molecular portraits of human breast tumours. *Nature* 406, 6797, 747–752.

Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, et al. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415, 6870, 436–442.

Guo-Jun Qi, Jinhui Tang, Zheng-Jun Zha, Tat-Seng Chua, and Hong-Jiang Zhang. 2009. An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 841–848.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Sridhar Ramaswamy, Ken Ross, Eric Lander, and Todd Golub. 2002. A molecular signature of metastasis in primary solid tumors. *Nat. Genet.* 33, 1, 49–54.

Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. 2011. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* 12, 1 (2011), 1–8. DOI: http://dx.doi.org/10.1186/1471-2105-12-77

Alexander Statnikov, Constantin F. Aliferis, Ioannis Tsamardinos, Douglas Hardin, and Shawn Levy. 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* 21, 5, 631–643. DOI: http://dx.doi.org/10.1093/bioinformatics/bti033

Andrew I. Su, John B. Welsh, Lisa M. Sapinoso, Suzanne G. Kern, Petre Dimitrov, Hilmar Lapp, Peter G. Schultz, Steven M. Powell, Christopher A. Moskaluk, Henry F. Frierson, and Garret M. Hampton. 2001. Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Res.* 61, 20, 7388–7393.

Scott A. Tomlins, Rohit Mehra, Daniel R. Rhodes, Xuhong Cao, Lei Wang, Saravana M. Dhanasekaran, et al. 2007. Integrative molecular concept modeling of prostate cancer progression. *Nat. Genet.* 39, 1, 41–51.

Lawrence True, Ilsa Coleman, others, and Peter S. Nelson. 2006. A molecular correlate to the Gleason grading system for prostate adenocarcinoma. *Proceedings of the National Academy of Sciences* 103, 29, 10991–10996. DOI:http://dx.doi.org/10.1073/pnas.0603678103

Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences* 98, 9, 5116–5121.

Faqiang Wang, Wangmeng Zuo, Lei Zhang, Deyu Meng, and David Zhang. 2013. A kernel classification framework for metric learning. *arXiv Preprint arXiv:1309.5823*.

Hua Wang, Feiping Nie, and Heng Huang. 2014. Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1836–1844.

Jun Wang, Huyen T. Do, Adam Woznica, and Alexandros Kalousis. 2011. Metric learning with multiple kernels. In *Advances in Neural Information Processing Systems*. 1170–1178.

Kilian Q. Weinberger and Lawrence K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* 10, 207–244.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bull.* 1, 6 (Dec. 1945), 80–83. DOI:http://dx.doi.org/10.2307/3001968

Rong Wu, Neali Hendrix-Lucas, others, Eric R. Fearon, and Kathleen R. Cho. 2007. Mouse model of human ovarian endometrioid adenocarcinoma based on somatic defects in the Wnt/-catenin and PI3K/Pten signaling pathways. *Cancer Cell* 11, 4, 321–333. DOI:http://dx.doi.org/10.1016/j.ccr.2007.02.016

Shiming Xiang, Feiping Nie, and Changshui Zhang. 2008. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognit.* 41, 12, 3600–3612.

Eric Xing, Andrew Ng, Michael Jordan, and Stuart Russell. 2003. *Distance Metric Learning with Application to Clustering with Side-Information*. MIT Press, 505–512.

Huilin Xiong and Xue-wen Chen. 2006. Kernel-based distance metric learning for microarray data classification. *BMC Bioinformatics* 7, 1 (2006), 1–11. DOI:http://dx.doi.org/10.1186/1471-2105-7-299

Liu Yang and Rong Jin. 2006. *Distance Metric Learning: A Comprehensive Survey*. Michigan State Universiy (2006), 1–51.

Eng-Juh Yeoh, Mary E. Ross, Sheila A. Shurtleff, W. Kent Williams, Divyen Patel, Rami Mahfouz, et al. 2002. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* 1, 2, 133–143.

Yiming Ying, Kaizhu Huang, and Colin Campbell. 2009. Sparse metric learning via smooth optimization. In *Advances in Neural Information Processing Systems* 22. 2214–2222.

Yiming Ying and Peng Li. 2012. Distance metric learning with eigenvalue optimization. *J. Mach. Learn. Res.* 13, 1, 1–26.

Changshui Zhang, Feiping Nie, and Shiming Xiang. 2010. A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomputing* 73, 4, 959–967.

Received October 2014; revised June 2015; accepted June 2015