Burg matrix divergence-based hierarchical distance metric learning for binary classification

WANG, Yan; LI, Hanxiong

Published in:
IEEE Access

Published: 01/02/2017

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
Unspecified

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2017.2669203

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.
Burg Matrix Divergence-Based Hierarchical Distance Metric Learning for Binary Classification

YAN WANG AND HAN-XIONG LI, (Fellow, IEEE)
Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong
State Key Laboratory of High Performance Complex Manufacturing, Central South University, Changsha 410012, China
Corresponding author: H. Xiong Li (mehxli@cityu.edu.hk)

The work was supported in part by the GRF Project from RGC of Hong Kong City University under Grant 11205615, and in part by the project from City University of Hong Kong under Grant 7004666.

ABSTRACT Distance metric learning is the foundation of many learning algorithms, and it has been widely used in many real-world applications. The basic idea of most distance metric learning methods is to find a space that can optimally separate data points that belong to different categories. However, current methods are mostly based on the single space that only learns one Mahalanobis distance for each data set, which actually fails to perfectly separate different categories in most real-world applications. To improve the accuracy of binary classification, a hierarchical method is proposed in this paper to completely separate different categories by sequentially learning subspace distance metrics. In the proposed method, a base-space distance metric is learned based on a similarity constraint first. Then, for binary classification problems, we formulate the subspace learning problem as a particular Burg Matrix optimization problem that minimizes the Burg Matrix divergence with distance constraints. Moreover, a cyclic projection algorithm is presented to solve the subspace learning problems. The experiments on five UCI data sets using different performance indices demonstrate the improved performance of the proposed method when compared with the state-of-the-art methods.

INDEX TERMS Hierarchical distance metric learning, sub-space distance metric, K-L divergence, LogDet optimization.

I. INTRODUCTION
Choosing a good distance metric in feature space is crucial in many supervised and unsupervised learning algorithms, such as nearest neighbors classifier [1] and K-means clustering [2]. Taking the k-nearest neighbors (kNN) classifier for instance, the key is to identify the set of labeled samples that are closest to a given test sample in the feature space, which involves the estimation of distance metric. However, the choice of distance metric is highly problem-specified and it ultimately determines the result of learning algorithms. Due to the high dimensions of data set in current problems, choosing distance metric manually is no longer an achievable goal for humans.

Over the past decade, many distance metric learning (DML) algorithms have been proposed to learn a Mahalanobis distance in feature space, and some of them have been successfully applied to real world applications. In order to learn a distance metric that can well separate the dissimilar data pairs, an earlier work [2] uses a semidefinite programming formulation under similarity and dissimilarity constraints. In [3], a similar method is presented as DLM-Eig which is further shown to be an eigenvalue optimization problem. Other eigenvalue methods are also popular, including Linear Discriminant Analysis (LDA) [4], Principle Component Analysis (PCA) [5] and Relevant Component Analysis (RCA) [6], [7].

In [8]–[11], the Large Margin Nearest Neighbor (LMNN) is suggested to learn a Mahalanobis distance metric for kNN Classification. By exploiting the nearest neighbor samples as side information, it maintains consistency in data’s neighborhood and keeps a large margin at the boundaries of different categories. In [12]–[14], the proposed algorithms, BoostMetric and MetricBoost, improve the performance of LMNN method by using the loss function to derive an AdaBoost like optimization procedure.

Another way to regularize metric is minimizing the divergence between the objective metric and a given metric.
In [15], [16], Information-Theoretic Metric Learning (ITML) method is presented as a particular Bregman optimization, and the distance function is chosen as Burg matrix divergence. It is more efficient and scalable for high-dimensional data because it does not require any eigenvalue computations or semi-definite programming. In addition, some other studies have also achieved encouraging performance in real world applications, including logistic Discriminant based DML (LDML) [17], keep it simple and straightforward (KISS) metric [18], multi-view neighborhood repulsed DML (MNRML) [19], evolutionary DML (EDML) [20], decomposition-based method for transfer DML (DTDML) [21], aggregated DML (ADML) [22], domain adaptation metric learning (DAML) [23] and semi-supervised multi-view DML (SSM-DML) [24].

Learning a Mahalanobis distance parameterized by a Positive Semi Definite (PSD) matrix \( A \) is exactly equal to learning a linear transformation \( L \) in feature space, where \( A = LL^T \).

So, current DML methods are actually learning one feature representation space that properly separate different categories. Unfortunately, due to the complex uncertainty, a space that can perfectly separate different categories may not exist in most real world applications.

In order to remedy the disadvantages of learning single-space distance metric, we propose a hierarchical distance metric learning (HDML) method, which can completely separates training data, as shown in figure 1. This method is fundamentally different with the Boostmetric method [12]–[14] which trains many trace-one metrics as weak learners. Our method actually trains a base-space metric as a strong learner, along with many sub-spaces to overcome the weakness of base-space. The relationship between base-space and subspaces is just similar to that of the international corporations and their local subsidiaries. For those markets with special demands and customs, the international corporations always build a local subsidiary to help them better adapt to the local market and thus get good achievement.

In section 2, inseparable data and inseparable set will be defined according to the idea of naive Bayes. In section 3, a base-space distance metric will be learnt first under similarity constraints. Then, for data points that fail to be correctly classified in base-space, a Burg matrix optimization problem which can better classify them is formulated to learn sub-space distance metrics. As indicated in figure 2, uncertain set will gradually shrink to empty, with the quantity increase of sub-spaces. Then the final classification result is obtained from the weighted mean of classification results in hierarchical spaces. In case study, we conduct experiments on five public data sets to demonstrate the effectiveness and universality of the proposed method.

II. PROBLEMS AND DEFINITIONS

This study will mainly address four problems: 1) How to define the inseparable of data? 2) How to find a proper base-space distance metric? (in section 3 A) 3) How to learn sub-space distance metrics that can effectively shrink the inseparable set? (in section 3 B) 4) How can we determine the final classification result from the hierarchical spaces (in section 3 D)? In this section, the criterion of inseparable will be defined as the foundation of the HDML method.

Given a data set \( x_i \), where \( x_i \in \mathbb{R}^d, i = 1, 2, \ldots, n \), the Mahalanobis distance parameterized by positive semi-definite (PSD) matrix \( A \) is expressed as:

\[
d_A(x_i, x_j) = \sqrt{(x_i - x_j)^T A^{-1} (x_i - x_j)}
\]

Following the idea of the naive Bayes method, a probability distribution is assigned to each of the possible class:

\[
P(x|\mu_k, A) = \frac{1}{(2\pi)^{d/2}|A|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^T A^{-1} (x - \mu_k)\right)
\]

where \( \mu_k \) is the center of data belonging to class \( k \). One thing should be noted is that all probability distributions are parameterized by the same covariance matrix \( A \). Then, the conditional probability of point \( x \) can be computed as:

\[
P(C_k|x, A) = \frac{P(C_k) P(x|\mu_k, A)}{P(x)}
\]

where \( C_k \) denotes the label of class \( k \), \( P(C_k|x, A) \) is the probability of a point \( x \) belonging to class \( k \) parameterized by PSD matrix \( A \). Then the conditional probability can be expressed as:

\[
P(C_k|x, A) = \frac{P(C_k) P(x|\mu_k, A)}{\sum_k P(C_k) P(x|\mu_k, A)}
\]

Based on the probability distributions, the inseparable data and the inseparable set can be defined as follows.
Definition 1 (Inseparable Data): For data \( x_i \in \mathbb{R}^d \), \( i = 1, 2, \ldots, n \), if
\[
P(y_i|x_i, A) < \max(P(y|x_i, A))
\] (5)
then data \( x_i \) is inseparable in space \( A \). In this formula, \( y_i \) is the actual class label of data \( x_i \), and \( y \) denotes all the possible class labels in data sets. In other words, if the actual label \( y_i \) is not the most probable class label of \( x_i \), then data \( x_i \) is inseparable in space \( A \).

Definition 2 (Inseparable Set): For data \( x_i \in \mathbb{R}^d \), \( i = 1, 2, \ldots, n \), if \( x_i \) is inseparable in all the existing spaces, then \( x_i \) will belong to inseparable set \( U \). As shown in figure 1, the goal of proposed method is gradually shrinking the inseparable data and inseparable set \( U \) to empty: \( U \to \emptyset \).

Definition 3 (Base-Space Distance Metric): The base-space distance metric will be a Mahalanobis distance parameterized by a PSD matrix \( A^0 \). It will be regarded as a global optimal metric which is learned in accordance with the target that similar pairs should be nearer and dissimilar pairs should be further.

Definition 4 (Sub-Space Distance Metrics): The sub-space distance metrics will be a group of Mahalanobis distances parameterized by different PSD matrix \( A^1, \ldots, A^k \). Under different probability constraints that force the data in inseparable set to be correctly classified, these distance metrics will be learnt by minimizing their K-L divergence to the base-space.

III. OPTIMIZATION FRAMEWORK

The optimization framework consists of two parts: base-space distance metric learning problem and sub-space distance metric learning problem. The base-space distance metric is exactly same with the Mahalanobis distance learned by most DML methods reviewed in section 1. In this paper, we will follow the method in [2] to find a distance metric under similar and dissimilar constraints, which is quite simple and practical. Then, by minimizing the Kullback-Leibler (K-L) divergence to the base-space, sub-space distance metrics will be learned under different probability constraints. In classification, an entropy based weighted mean method will be adopted to compute the final prediction results.

A. BASE-SPACE DISTANCE METRIC LEARNING

For data set \( x_i \), where \( x_i \in \mathbb{R}^d \), \( i = 1, 2, \ldots, n \), similar and dissimilar pairs are defined as follows.

Similar pairs \((x_i, x_j) \in S : x_i, x_j \) belong to the same class;

Dissimilar pairs \((x_i, x_j) \in D : x_i, x_j \) belong to different classes;

The target of base-space distance metric is finding a Mahalanobis distance that can best separate dissimilar points. So, a simple way of defining a criterion for the desirable metric is to demand that pairs of points \((x_i, x_j) \in S \) have small squared distance \( \min \sum_{x_i, x_j \in S} d_{A^0}(x_i, x_j) \) subject to the constraint \( \sum_{x_i, x_j \in D} d_{A^0}(x_i, x_j) \geq 1 \). This gives the optimization problem for base-space distance metric \( A^0 \).

\[
\begin{align*}
\min_{A^0} & \sum_{x_i, x_j \in S} d_{A^0}(x_i, x_j) \\
\text{s.t.} \hspace{0.5cm} & 1) \sum_{x_i, x_j \in D} d_{A^0}(x_i, x_j) \geq 1 \\
 & 2) A^0 \succeq 0;
\end{align*}
\] (6)

After learning, \( A^0 \) will be the base space metric. Then we define:
\[
g(A^0) = \sum_{x_i, x_j \in S} (x_i - x_j)^T (A^0)^{-1}(x_i - x_j)
\]
\[
- \log \sum_{x_i, x_j \in D} (x_i - x_j)^T (A^0)^{-1}(x_i - x_j)
\] (7)

It is straightforward to show that minimizing \( g(A^0) \) (subject to \( A^0 \succeq 0 \)) is equivalent. Then, we can use Newton-Raphson to efficiently optimize \( g \). With base-space distance metric \( A^0 \), the inseparable data and inseparable set \( U \) can be computed via (1)-(4). In following sub-sections, the task of sub-space learning is to shrink the inseparable set \( U \) by sequentially learning different distance metrics.

B. SUB-SPACE LEARNING FOR BINARY PROBLEMS

1) PROBLEM FORMULATION

As discussed above, once the base space \( A^0 \) is determined, the task of sub-space learning is shrinking the uncertain set \( U \). The most straightforward method is re-learning a distance metric to properly classify the data in uncertain set \( U \). However, this method usually leads to over-fitting problem when uncertain set \( U \) shrinks to a very small size. On this point, our solution is minimizing the distance between sub-spaces and the base-space under different probability constraints. The probability constraints we adopt here will force the data in uncertain set \( U \) to be correctly classified in subsequent sub-space. Therefore, our method learning sub-spaces as the base-space with minor modifications to avoid the over-fitting problem. To achieve this target, we first measure the distance between sub-space and the base space by using the K-L divergence of corresponding probability distributions:

\[
KL(P(x|\mu, A^0); P(x|\mu, A^{new}))
\]
\[
= \int P(x|\mu, A^0) \log \frac{P(x|\mu, A^0)}{P(x|\mu, A^{new})}
\] (8)

The K-L divergence in (7) provides a well-founded measure of similarity between two metrics. Then, in the term of binary problems, the condition of inseparable data:
\[
P(y_i|x_i, A) < \max(P(y|x_i, A))
\] can be rewritten as:
\[
P(y = y_i|x_i, A) < P(y = \bar{y}_i|x_i, A)
\] (9)

where \( y_i \) is the actual class label of data \( a \), and \( \bar{y}_i \) is the other. To force the inseparable data to be separable, a constraint is
proposed as:
\[
\sum_j P(y = y_j|x_j, A^{\text{new}}) \geq \sum_j P(y = \overline{y}_j|x_j, A^{\text{new}}), \quad x_j \in U
\] (10)

This constraint can force approximately half of the data in inseparable set to be inseparable. Thus, the sub-space distance metric learning problem is formulated:
\[
\min_{A^{\text{new}}} \sum_i KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}}))
\]
subject to 1) \(\sum_j P(y = y_j|x_j, A^{\text{new}}) \geq \sum_j P(y = \overline{y}_j|x_j, A^{\text{new}}), \quad x_j \in U;\)
2) \(A^{\text{new}} \succeq 0;\) (11)

2) K-L DIVERGENCE
The K-L divergence in (7) can be presented as:
\[
KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}})) = \int P(x_i|\mu, A^0) \log P(x_i|\mu, A^0)
- \int P(x_i|\mu, A^{\text{new}}) \log P(x_i|\mu, A^{\text{new}})
\] (12)

The first term is the differential entropy of \(P(x_i|\mu, A^0),\) which can be expressed as:
\[
KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}})) = \int P(x_i|\mu, A^0) \left[ -\log(2\pi)^\frac{1}{2} |A^0|^\frac{1}{2} \\
- \frac{1}{2} (x - \mu)^T (A^0)^{-1} (x - \mu) \right]
- \frac{1}{2} \log(2\pi)^d |A^{\text{new}}| - \frac{1}{2} \int P(x_i|\mu, A^{\text{new}}) \log((A^{\text{new}})^{-1}(x - \mu) (x - \mu)^T)
- \frac{1}{2} \log(2\pi)^d |A^{\text{new}}| - \frac{1}{2} \log((A^{\text{new}})^{-1} A^0)
= \frac{1}{2} \log(2\pi)^d |A^0| - \frac{1}{2} \log((A^{\text{new}})^{-1} A^0)
\] (13)

For the second term:
\[
KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}})) = \int P(x_i|\mu, A^0) \left[ -\log(2\pi)^\frac{1}{2} |A^{\text{new}}|^\frac{1}{2} \\
- \frac{1}{2} (x - \mu)^T (A^{\text{new}})^{-1} (x - \mu) \right]
- \frac{1}{2} \log(2\pi)^d |A^{\text{new}}| - \frac{1}{2} \log((A^{\text{new}})^{-1} A^0)
\] (14)

Consequently, the loss function in (10) will be
\[
\sum_i KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}})) \propto KL(P(x_i|\mu, A^0); P(x_i|\mu, A^{\text{new}}))
= \frac{1}{2} \log((A^{\text{new}})^{-1} A^0) - \log|(A^{\text{new}})^{-1} A^0|
\] (15)

The above formulas follow the definition of the covariance \(A = E((x - \mu)(x - \mu)^T),\) and also follow the fact that \((x - \mu)^T A^{-1} (x - \mu) = tr(A^{-1}(x - \mu)(x - \mu)^T).\) Then problem (10) is re-formulated as:
\[
\min_{A^{\text{new}}} tr((A^{\text{new}})^{-1} A^0) - \log|(A^{\text{new}})^{-1} A^0|
\]
subject to 1) \(\sum_j P(y = y_j|x_j, A^{\text{new}}) \geq \sum_j P(y = \overline{y}_j|x_j, A^{\text{new}}), \quad x_j \in U;\)
2) \(A^{\text{new}} \succeq 0;\) (16)

3) SIMPLIFIED CONSTRAINTS IN BINARY PROBLEMS
In problem (15), the constraint \(\sum_j P(y = y_j|x_j, A^{\text{new}}) \geq \sum_j P(y \neq y_j|x_j, A^{\text{new}}), \quad x_j \in U\) is nonlinear, which is too complex to optimize. In this paper, we consider binary classification problem only, and re-formulating problem (15) as a Burg Matrix optimizing problem [25].

For a binary classification problem, since the covariance matrix of two multi-Gaussian distributions are same, inequality \(P(y = y_j|x_j, A^{\text{new}}) - P(y = \overline{y}_j|x_j, A^{\text{new}}) > 0\) can be expressed as:
\[
P(y = y_j|x, A^{\text{new}}) - P(y = \overline{y}_j|x, A^{\text{new}}) \geq 0
\rightarrow P(y = y_j) * P(x_j|y_j, A^{\text{new}}) - P(y = \overline{y}_j)P(x_j|y = \overline{y}_j, A^{\text{new}}) \geq 0
\rightarrow P(y = y_j) \exp(-\frac{1}{2} d^{\text{new}}(x, \mu_{y_j}))
- P(y = \overline{y}_j) \exp(-\frac{1}{2} d^{\text{new}}(x, \mu_{\overline{y}_j})) \geq 0
\rightarrow \log \frac{P(y = y_j)}{P(y = \overline{y}_j)} - \frac{1}{2} d^{\text{new}}(x, \mu_{y_j}) + \frac{1}{2} d^{\text{new}}(x, \mu_{\overline{y}_j}) \geq 0
\rightarrow d^{\text{new}}(x_j, \mu_{y_j}) - d^{\text{new}}(x_j, \mu_{\overline{y}_j}) - 2 \log \frac{P(y = y_j)}{P(y = \overline{y}_j)} \geq 0
\] (17)

Then, a linear approximate constraint is written as:
\[
\sum_j (d^{\text{new}}(x_j, \mu_{y_j}) - d^{\text{new}}(x_j, \mu_{\overline{y}_j}) - 2 \log \frac{P(y = y_j)}{P(y = \overline{y}_j)}) \geq 0
\quad \text{for } x_j \in U
\] (18)

which can be further expressed as:
\[
tr(A^{\text{new}}) \sum_j ((x_j - \mu_{y_j})(x_j - \mu_{y_j})^T
- (x_j - \mu_{\overline{y}_j})(x_j - \mu_{\overline{y}_j})^T) \leq 2NU \log \frac{P(y = y_j)}{P(y = \overline{y}_j)}
\quad \text{for } x_j \in U
\] (19)
Algorithm 1 Sub-Space Distance Metric Learning

Input: X: n*d training set; A^0: Base-space distance metric; U: Current inseparable set; μ_0, μ_1: centers of input Gaussians; N_U: Number of data points in U

Output: A^{new}: new distance metric; U^{new}: shrink inseparable set

1: A^{new} ← A^0; λ ← 0; b = 2 N_U \log \frac{P(y = y_j)}{P(y = \overline{y}_j)};
2: repeat
3: For data x_j in U: p ← tr(A^{new}^{-1} \sum_j ((x_j - μ_j)(x_j - μ_j)^T - (x_j - μ_j^*)(x_j - μ_j^*)^T))
4: α ← min (λ, \frac{1}{p} - \frac{1}{b})
5: λ ← λ - α
6: β ← α - αp
7: A^{new^{-1}} = A^{new^{-1}} + β A^{new^{-1}} \sum_i ((x_i - μ_j)(x_i - μ_j)^T - (x_i - μ_j^*)(x_i - μ_j^*)^T) A^{new^{-1}}
8: until convergence
9: A^{new} ← (A^{new^{-1}})^{-1}
10: Computing P(y|x, A^{new}), P(y|x, A^{new^*}) via (1)-(4)
11: for x_j ∈ U do
12: if P(y = y_j | x_i, A^{new}) < P(y = \overline{y}_j | x_i, A^{new}) then
13: U^{new} ← U^{new} ∪ x_i
14: end if
15: end for

where N_U is the number of data points in inseparable set U.

Then, the sub-space metric learning problem (15) can be formulated as a Burg Matrix optimization problem [25]:

min \min_{A^{new}} tr((A^{new^{-1}})^{-1} A^0) - \log |A^{new^{-1}}|^{-1} \]
s.t. 1) tr(A^{new^{-1}} \sum_j ((x_j - μ_j)(x_j - μ_j)^T - (x_j - μ_j^*)(x_j - μ_j^*)^T) ≤ 2 N_U \log \frac{P(y = y_j)}{P(y = \overline{y}_j)} \quad \text{for } x_i \in U
2) A^{new} ≥ 0 \quad (20)

4) CYCLIC PROJECTIONS

To solve the optimization problem (20), we extend the cyclic projections method in [15, 16, 25]. The optimization method forms the basis on iteratively computing the projection of current solution onto a single constraint. With a constraint like \( tr(A^{new^{-1}} K) ≤ b \), this projection is updated as:

A_{t+1} = A_t + β A_t K A_t \quad (21)

where β is a Lagrange multiplier computed by the algorithm. Each iteration in this algorithm costs O(d^2) computation time. The resulting algorithm with a closed form solution is given as algorithm 1. The inputs to the algorithm are the Mahalanobis matrix A_0 in base-space metric, training set X, inseparable set U, and centers of input Gaussian.

Note: Rank in Burg matrix optimization. In algorithm 1, the rank of A_{new} will be equal to the rank of A_0 during all iterations. The illustration can be shown as follows:

Let the eigendecompositions of A_0 and A_{new} be A_0 = V_0 A_0 M and A_{new} = U_0 A_{new} M, the rank of A will be equal to the rank of M at each iteration.

Algorithm 2 HDML

Input: X: n*d training set; U: Current inseparable set; μ_0, μ_1: centers of input Gaussians; α: Similar pairs; D: Set of dissimilar pairs

Output: A^0, ..., A^k: distance metrics,

1: U ← φ
2: Base-space distance metric learning:
3: A^0 ← arg min \sum_{x_i, x_j ∈ S} d^0(x_i, x_j);
4: s.t. \sum_{S} ≥ 1; \quad A^0 ≥ 0;
5: Computing P(x|μ_i, A^0), P(y|x, A^0) via (1)-(4)
6: for i = 1:n do
7: if P(y = y_i | x_i, A_{new}) < P(y = \overline{y}_j | x_i, A_{new}) then
8: U^{old} ← U^{old} ∪ x_i
9: end if
10: end for
11: Sub-space distance metric learning:
12: Computing A^{new} and U^{new} via algorithm 1;
13: A^i ← A^{new}, j ← j + 1
14: until size of U < tol

C. HIERARCHICAL DISTANCE METRIC LEARNING ALGORITHM

The HDML algorithm is shown as algorithm 2. Specifically, the base-space distance metric will be learned in problem (6). Then, for the inseparable set in the base space, sub-spaces are sequentially learned by solving problem (20) until the inseparable set shrinks to empty.

D. CLASSIFICATION

In classification, we construct classification models \( f^0, ..., f^m \) in each space, where \( f^j(x) \) will be the
classification result in space $A^i$. The final classification will be obtained from the weighted mean of $f_0(x), f_1(x), \ldots, f_m(x)$, as follows.

For input $x$, its entropy in space $A^i$ is defined as:

$$I(x|A^i) = -\sum_k P(C_k|x, A^i) \log P(C_k|x, A^i)$$ (23)

Then, the exponent of its entropy in each space will be regarded as the confidence weights of prediction result: $\omega_i = \exp (-I(x|A^i))$. For binary classification such as 0-1 classification, the probability of $y = 1$ can be computed as:

$$p(y = 1|x, A^0, \ldots, A^m) = \frac{\sum_{i=0}^m \omega_i f_i(x)}{\sum_{i=0}^m \omega_i}$$ (24)

And consequently the final classification result will be:

$$y = \arg \max_y p(y|x, A^0, \ldots, A^m)$$ (25)

### IV. EXPERIMENTS

In this section, we conduct experiments on five public data sets from UCI [26] to demonstrate the effectiveness of the proposed method. In experiments I, the classification accuracies and computation time of k-Nearest Neighbor (kNN) associated with different distance metrics are compared. In experiment II, the HDML-kNN is compared with support vector machine (SVM) and logistic regression.

#### A. EXPERIMENT I: COMPARISON WITH OTHER DML METHODS

In experiment I, the HDML is compared with many state-of-the-art DML methods, including Euclidean distance, Mahalanobis distance, lda, ITML [15], [16] and LMNN [10], [11]. In the first experiment, for all data sets we have set $k = 1$ for nearest neighbor classification. The trade-off parameters in ITML and LMNN are tuned via three-fold cross validation.

We run experiments on 5 UCI public data sets, which are: Pima Indian Diabetes (768 samples and 8 features), Breast Cancer Wisconsin Diagnostic (WDBC, 569 samples and 30 features), Heart (270 samples and 13 features), Liver Disorders (BUPA, 345 samples and 6 features) and Robot execution failures (540 samples and 20 features). All experimental results are obtained by averaging 20 runs. For each run, we randomly split the data sets 80% for training and 20% for testing.

As the accuracy of kNN classification is always affected by the different scales of features, we show the testing result before normalization in table 1, and the testing result after normalization in table 2. The proposed method achieves the best performance on most data sets. We can conclude that through the cooperation of hierarchical spaces, the proposed method can efficiently improve the classification accuracy of kNN. Since no hyper-parameter exists, the performance of proposed method is robust and easily achievable.

| TABLE 1. KNN (k=1) classification error via different metrics (before normalization). |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Diabetes        | WDBC            | Heart           | BUPA            | Failures        |
| HDML            | 0.307(0.032)    | 0.064(0.012)    | 0.242(0.039)    | 0.393(0.039)    | 0.121(0.014)    |
| LMNN            | 0.310(0.023)    | 0.074(0.019)    | 0.260(0.041)    | 0.402(0.049)    | 0.137(0.016)    |
| LDA             | 0.322(0.021)    | 0.073(0.014)    | 0.245(0.044)    | 0.433(0.044)    | 0.118(0.019)    |
| ITML            | 0.319(0.029)    | 0.088(0.033)    | 0.272(0.054)    | 0.390(0.029)    | 0.150(0.018)    |
| Euclidean       | 0.320(0.021)    | 0.085(0.035)    | 0.410(0.081)    | 0.393(0.054)    | 0.202(0.014)    |
| Mahalanobis     | 0.329(0.020)    | 0.105(0.027)    | 0.419(0.052)    | 0.449(0.069)    | 0.215(0.019)    |

| TABLE 2. KNN (k=1) classification error via different metrics (after normalization). |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Diabetes        | WDBC            | Heart           | BUPA            | Failures        |
| HDML            | 0.275(0.022)    | 0.037(0.021)    | 0.268(0.031)    | 0.357(0.045)    | 0.112(0.018)    |
| LMNN            | 0.277(0.018)    | 0.047(0.019)    | 0.371(0.061)    | 0.307(0.027)    | 0.130(0.010)    |
| LDA             | 0.310(0.024)    | 0.060(0.014)    | 0.261(0.034)    | 0.402(0.055)    | 0.133(0.013)    |
| ITML            | 0.288(0.029)    | 0.045(0.030)    | 0.249(0.059)    | 0.406(0.031)    | 0.130(0.022)    |
| Euclidean       | 0.301(0.016)    | 0.048(0.032)    | 0.403(0.070)    | 0.412(0.044)    | 0.144(0.021)    |
| Mahalanobis     | 0.308(0.019)    | 0.048(0.017)    | 0.433(0.048)    | 0.408(0.063)    | 0.148(0.013)    |

| TABLE 3. Number of sub-spaces on different data sets. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Diabetes        | WDBC            | Heart           | BUPA            | Failures        |
|                 | 4              | 2              | 3              | 6              | 4              |

| TABLE 4. Computational time of metric learning (s). |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Diabetes        | WDBC            | Heart           | BUPA            | Failures        |
| HDML            | 4.77            | 1.03            | 2.21            | 0.97            | 4.41            |
| LMNN            | 3.65            | 1.55            | 0.33            | 0.70            | 1.34            |
| ITML            | 5.34            | 6.16            | 6.37            | 4.79            | 7.13            |

To understand the complexity of HDML, the number of sub-space metrics learned for each data set after normalization is listed in table 3, which is less than 10 on each data set. The computation time of LMNN, ITML and proposed method are shown in table 4.

Then, we compare the performance of HDML with other 5 methods under different number of neighbors. In figure 3 and figure 4, we plot the test error versus number.
TABLE 5. Classification error and standard deviation of different classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Diabetes</th>
<th>WDBC</th>
<th>Heart</th>
<th>BUPA</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDML-kNN</td>
<td>0.245(0.032)</td>
<td>0.047(0.012)</td>
<td>0.212(0.039)</td>
<td>0.301(0.039)</td>
<td>0.071(0.014)</td>
</tr>
<tr>
<td>SVM</td>
<td>0.231(0.019)</td>
<td>0.049(0.013)</td>
<td>0.208(0.042)</td>
<td>0.290(0.043)</td>
<td>0.080(0.028)</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.241(0.021)</td>
<td>0.072(0.014)</td>
<td>0.245(0.044)</td>
<td>0.331(0.044)</td>
<td>0.113(0.019)</td>
</tr>
<tr>
<td>Euclidean knn</td>
<td>0.273(0.021)</td>
<td>0.048(0.031)</td>
<td>0.312(0.054)</td>
<td>0.352(0.041)</td>
<td>0.124(0.025)</td>
</tr>
</tbody>
</table>

B. EXPERIMENT II: COMPARISON WITH OTHER CLASSIFIERS

In Experiment II, we compare our method, HDML-kNN, with SVM, logistic regression and Euclidean knn on the same data sets. The SVM and logistic regression were implemented through Libsvm [27] and Liblinear [28], respectively. For each data set, we try our best to manually adjust the parameters to the optimum.

We conduct experiments on five data sets. For Pima Indian diabetes, we set ‘k=3’ in HDML-kNN and Euclidean knn, ‘cost=5’ and ‘radial basis kernel’ in SVM. For WDBC, we set ‘k=1’ in HDML-kNN and Euclidean knn, ‘cost=0.1’ and ‘linear kernel’ in svm. For Heart, we set ‘k=11’ in HDML-kNN and Euclidean knn, ‘cost=5’ and ‘radial basis kernel’ in SVM. For BUPA, we set in SVM. For Popfailures, we set ‘k=5’ in HDML-kNN and Euclidean knn, ‘cost=1’ and ‘radial basis kernel’ in SVM. All experimental results are obtained by averaging 20 runs. For each run, we randomly split the data sets 80% for training and 20% for testing.

As we can see from Table 5, HDML-kNN out-performs SVM and logistic regression on 2 of the 5 data sets. Although HDML-kNN does not prevail in all data sets, the experiment result powerfully demonstrates that the proposed method is an effective classifier which reaches, and can even exceed, the performance of some state-of-the-art classifiers.

V. CONCLUSION

Instead of classification in single-space, a hierarchical model is proposed in this paper to improve the accuracy of binary classification. By proposing sub-space DML problem as a Burg Matrix optimization problem, the proposed model enables us to derive an efficient close-form algorithm. The experiments on five UCI data sets show that this method can effectively improve the classification accuracy of kNN algorithm.

Actually, the proposed method should be regarded more as a general framework than as a completed algorithm. Its essence is shrinking the inseparable data set by sequentially learning new spaces. In this framework, the base-space DML problem can be solved by many other methods. And the sub-space DML problem can also be solved by other kinds of constraints that can effectively eliminate the inseparable set. A flaw of this method is that the Burg Matrix optimization in this paper is designed merely for binary classifications. In further study, this method will be extended to multi-class classification problems as well as regression problems.

REFERENCES

YAN WANG was born in Wuhan, China, in 1988. He received the M.S. degree in communications and information systems from Wuhan University of Technology, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Systems Engineering and Engineering Management, City University of Hong Kong. From 2015 to 2016, he was an Intern with Microsoft Research Asia. His research interests include distance metric learning, natural language processing, text mining, and sentiment analysis.

HAN-XIONG LI (S’94–M’97–SM’00–F’11) received the B.E. degree in aerospace engineering from National University of Defense Technology, China, in 1982, the M.E. degree in electrical engineering from Delft University of Technology, The Netherlands, in 1991, and the Ph.D. degree in electrical engineering from The University of Auckland, New Zealand, in 1997. He is currently a Professor with the Department of Systems Engineering and Engineering Management, City University of Hong Kong. He has broad knowledge and experience in both academia and industry. He has authored two books and published more than 170 SCI journal papers with h-index 31. He holds six patents. His research interests are in system intelligence and control, process design and control integration, and distributed parameter systems with applications to electronics packaging.