AdaBoost on Low-Rank PSD Matrices for Metric Learning

Jinbo Bi[1]*, Dijia Wu[2], Le Lu[3] Meizhu Liu[4], Yimo Tao[5], Matthias Wolf[3]
[1]University of Connecticut, Storrs, CT 06250, jinbo@engr.uconn.edu
[2]Siemens Corporate Research, Princeton, NJ 08540, dijia.wu@siemens.com
[3]Siemens Medical Solutions, Malvern, PA 19355, le-lu, mwolf@siemens.com
[4]University of Florida, Gainesville, FL 32611, mliu@cise.ufl.edu
[5]Microsoft Corporation, Redmond, WA 98052, yimot@microsoft.com

Abstract

The problem of learning a proper distance or similarity metric arises in many applications such as content-based image retrieval. In this work, we propose a boosting algorithm, MetricBoost, to learn the distance metric that preserves the proximity relationships among object triplets: object i is more similar to object j than to object k. MetricBoost constructs a positive semi-definite (PSD) matrix that parameterizes the distance metric by combining rank-one PSD matrices. Different options of weak models and combination coefficients are derived. Unlike existing proximity preserving metric learning which is generally not scalable, MetricBoost employs a bipartite strategy to dramatically reduce computation cost by decomposing proximity relationships over triplets into pair-wise constraints. MetricBoost outperforms the state-of-the-art on two real-world medical problems: 1. identifying and quantifying diffuse lung diseases; 2. colorectal polyp matching between different views, as well as on other benchmark datasets.

1. Introduction

The choice of a distance or similarity metric over the input space is critical to the performance of many learning algorithms such as the simplest k-Nearest-Neighbor (k-NN) classifier and K-means clustering. Clearly, a good metric is task dependent and previous work [8, 11, 9] has shown k-NN classification accuracy significantly benefits from properly designed distance metric as opposed to the standard Euclidean distance. Metric learning algorithms are often derived from weak labeling of training data. Unlike in traditional classification problems where each training example is associated with a class label, equivalence constraints are provided in metric learning as pairs $(\mathbf{x}_i, \mathbf{x}_j)$ to indicate if the two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ are "similar" or "dissimilar" [20, 3, 2, 4]. An even weaker representation often used in information retrieval [13, 17] is the proximity relationships over triplets (i, j, k): $\mathbf{x}_i$ is closer to $\mathbf{x}_j$ than to $\mathbf{x}_k$. The goal of metric learning is then to learn a distance metric d so that $d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_i, \mathbf{x}_k)$. Proximity relationships are the most natural input for learning a metric, and are of the weakest representation because proximity relation triplets can be derived from equivalence constraints or from traditional format of classification training data $(\mathbf{x}_i, y_i)$ where y is the class label, but not vice versa. The proposed algorithm is a type of proximity preserving approach.

Mahalanobis distance metric which is parameterized by a positive semidefinite (PSD) matrix $\mathbf{M}$ [4, 18, 1] is well-studied and has shown advantages over some other metrics such as multidimensional scaling and locally linear embedding. We design an efficient AdaBoost algorithm which we call, MetricBoost, to learn a Mahalanobis distance that preserves proximity relationships. MetricBoost constructs the matrix $\mathbf{M}$ by additively combining rank-one PSD matrices. Different options of weak models and combination coefficients α are investigated for MetricBoost. As proximity relationship is weak "side information" [4], a large amount of proximity triplets (i, j, k) are often needed in order to learn a proper distance. Existing metric learning algorithms require a computation cost in the order of the number of triplet conditions [17, 16], and thus are vulnerable to scalability issues. In contrast, MetricBoost is computationally efficient due to the decomposition from triplet conditions into pair-wise constraints via a bipartite strategy.

2. Preliminaries

The Mahalanobis metric can be written in terms of $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{M}(\mathbf{x} - \mathbf{y})}$ where $\mathbf{M}$ is a PSD matrix and is often set to the covariance matrix of the training data if no pre-training. The PSD matrix $\mathbf{M}$ is usually nor-

*This work was conducted when the first author was a scientist at Siemens Medical Solutions and the 2^{nd}, 4^{th} and 5^{th} authors worked as summer interns at Siemens Medical Solutions.

malized to prevent the distance to be arbitrarily large by requiring $tr(\mathbf{M}) \leq 1$. To learn a Mahalanobis metric in terms of this representation, it often needs to solve a positive semidefinite program to determine an appropriate PSD matrix $\mathbf{M}$. Current PSD solvers using interior point methods do not scale well to large problems with computation complexity roughly $O(n^{3.5})$ where n is the number of variables. Recently, an effective PSD solver [17] has been proposed for metric learning by column generation techniques. At each iteration, a linear program over all weak models needs to be solved. A later version [16] of this method [17] becomes faster by an iterative process which calculates closed-form updates at each iteration. However, it still requires space and time in an order of the total number of proximity triplets per iteration.

A PSD matrix $\mathbf{M}$ can be eigen-decomposed into $\sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$ where λ_i's are eigenvalues and $\mathbf{u}_i$'s are orthonormal eigenvectors. Since $\mathbf{M} \succeq 0$, all $\lambda_i \geq 0$, and $\mathbf{M} = \sum_i \lambda_i \mathbf{U}_i$ where matrices $\mathbf{U}_i = \mathbf{u}_i \mathbf{u}_i^\top$ are rank-one matrices. Given that $\mathbf{u}_i$ is a vector whose norm equals to 1, $tr(\mathbf{U}_i) = 1$. We give the following definitions. Let $\Omega = \{\mathbf{M} | \mathbf{M} \succeq 0, tr(\mathbf{M}) = 1\}$ be the space of $d \times d$ PSD matrices with trace equal to 1. Let $\Omega_1 = \{\mathbf{U} | \mathbf{U} \succeq 0, tr(\mathbf{U}) = 1, rank(\mathbf{U}) = 1\}$ be the space of $d \times d$ rank-one PSD matrices with trace equal to 1. It has been proved [17] that Ω is the convex hull of Ω_1 and matrices in Ω_1 form the set of extreme points of Ω.

In the MetricBoost setting, the metric $d(\cdot, \cdot)$ is assumed to take the form of $\sqrt{H(\cdot, \cdot)}$ where H is constructed as a linear combination of multiple weak models $H(\cdot, \cdot) = \sum_t \alpha_t h_t(\cdot, \cdot)$, and h_t is a weak hypothesis and α_t is the combination coefficient. The weak model $h_t(\mathbf{x}, \mathbf{y})$ is parameterized by a rank-one PSD matrix $\mathbf{U}_t$ as $(\mathbf{x} - \mathbf{y})^\top \mathbf{U}_t (\mathbf{x} - \mathbf{y})$. The training data $\mathbf{X}$ which comprises training vectors $\mathbf{x}$ as rows is given together with triplet index set I_{tr} containing triplets of example indices such as (i, j, k). Each triplet (i, j, k) imposes a triplet condition: $d(\mathbf{x}_i, \mathbf{x}_j) < d(\mathbf{x}_i, \mathbf{x}_k)$. The performance of a distance metric can be evaluated in different ways depending on the specific target. In this paper, for a given triplet $(i, j, k) \in I_{tr}$, the error is 1 if $d(\mathbf{x}_i, \mathbf{x}_j) \geq d(\mathbf{x}_i, \mathbf{x}_k)$ or 0 otherwise. Therefore the overall error rate ϵ can be characterized as

$$\epsilon = \sum_{(i,j,k) \in I_{tr}} D((i,j,k)) \mathbf{1}_{(H(\mathbf{x}_i, \mathbf{x}_j) \geq H(\mathbf{x}_i, \mathbf{x}_k))} \quad (1)$$

where D is a given distribution of triplets over I_{tr} and $\mathbf{1}_{(a \geq b)}$ equals to +1 if $a \geq b$ and 0 otherwise.

3. MetricBoost Algorithm

Boosting is a machine learning approach to generation of highly accurate predictive models by combining many "weak" models which may be only moderately accurate.

Algorithm 1 Algorithm MetricBoost($\mathbf{X}$, I_{tr}, T)

```
Input: $\mathbf{X} = [\mathbf{x}_1 \, \mathbf{x}_2 \, \cdots \, \mathbf{x}_n]^\top$,
       $I_{tr} = \{(i,j,k) \mid \mathbf{x}_i$ is closer to $\mathbf{x}_j$ than to $\mathbf{x}_k\}$.
Initialize $D_1((i,j,k))$ (usually set to $1/m$ where $m$ is the total number of triplets).
for $t = 1$ to $T$ do
    Train a weak learner using distribution $D_t$.
    Get weak hypothesis $h_t(\mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$
    Choose $\alpha_t \in \mathcal{R}$.
    Update
        $D_{t+1}((i,j,k)) = \frac{D_t((i,j,k)) \exp(\alpha_t(h_t(\mathbf{x}_i, \mathbf{x}_j) - h_t(\mathbf{x}_i, \mathbf{x}_k)))}{Z_t}$    (2)
    where $Z_t$ is a normalization factor (chosen so that $D_{t+1}$ is a distribution).
end for
Output the final model $H(\cdot, \cdot) = \sum_t \alpha_t h_t(\cdot, \cdot)$.
```

In the MetricBoost setting, each weak hypothesis h_t is the square of a distance metric, is solely determined by a rank-one matrix $\mathbf{U}_t$. The goal is to learn h_t which satisfies a moderate amount of triplet conditions as defined in I_{tr}, and combine h_t in an effective way.

We outline the procedure of MetricBoost in Algorithm 1 which largely follows the existing AdaBoost.

In the algorithm, the weak model is $h_t(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top \mathbf{U}_t (\mathbf{x} - \mathbf{y})$ where $\mathbf{U}_t = \mathbf{u}_t \mathbf{u}_t^\top$, and the final hypothesis is $H(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top \mathbf{M} (\mathbf{x} - \mathbf{y})$ where $\mathbf{M} = \sum_t \alpha_t \mathbf{U}_t$. If $\alpha_t \geq 0 \quad \forall t \in [1, \cdots, T]$, $\mathbf{M}$ is a PSD matrix, and we define the distance function $d(\cdot, \cdot) = \sqrt{H(\cdot, \cdot)}$. A PSD matrix $\mathbf{M}$ does not guarantee the resulting d to satisfy the identity of indiscernibles, i.e. $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$ for a metric, but with a bit misuse of the notation, we still call d a metric. Notice that rescaling the matrix $\mathbf{M}$ preserves the satisfaction of triplet constraints. In other words, if $\mathbf{M}$ forms a metric that satisfies the triplet conditions, so does its multiplier. Meanwhile, α_t and $\mathbf{u}_t$ may not be the eigenvalue and eigenvector of the final matrix $\mathbf{M}$ as orthonormality is not required for weak models $\mathbf{u}_t$.

The training error of the final hypothesis H as defined in Eq.(1) is upper bounded by $\prod_{t=1}^T Z_t$ as shown below.

Theorem 1 *Using the notation in Algorithm 1, the following bound holds on the training error of H with respect to the initial distribution D:*

$$\sum D((i,j,k)) \mathbf{1}_{(H(\mathbf{x}_i, \mathbf{x}_j) \geq H(\mathbf{x}_i, \mathbf{x}_k))} \leq \prod_{t=1}^T Z_t.$$

Proof. Unraveling the update rule (2), we obtain

$$D_{T+1}((i,j,k)) = \frac{D((i,j,k))\exp(H(\mathbf{x}_i,\mathbf{x}_j) - H(\mathbf{x}_i,\mathbf{x}_k))}{\prod_{t=1}^{T} Z_t}$$

According to the fact that $\mathrm{sgn}(x) \leq e^x$ for all real x, the training error of H with respect to the initial distribution D satisfies

$$\begin{aligned} & \sum D((i,j,k))\mathbf{1}_{(H(\mathbf{x}_i,\mathbf{x}_j)\geq H(\mathbf{x}_i,\mathbf{x}_k))} \\ \leq \;& \sum D((i,j,k))\exp(H(\mathbf{x}_i,\mathbf{x}_j) - H(\mathbf{x}_i,\mathbf{x}_k)) \\ = \;& \sum D_{T+1}((i,j,k))\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} Z_t. \end{aligned}$$

This proves the upper bound. ∎

Now what remains important is how to choose combination coefficients α_t and construct the weak hypothesis h_t. Multiple choices exist depending on the output range of weak hypothesis, similar to those in the derivation of AdaBoost. For any given weak model h_t, α_t can be chosen by minimizing Z_t at each iteration t.

3.1. Choosing α_t

According to Theorem 1, low training error can be obtained if Z_t is minimized at each round to achieve $Z_t \leq 1$. Relying on the output range of the weak model h_t, three options are typically discussed.

1. Generally, we expect a distance function to output any proper nonnegative real number, which means that the weak model h_t ranges in $[0, \infty]$. However, for this general range of h_t, there has not been any analytical formula for calculation of α_t. In stead, Z_t can be viewed as a function of α_t and a binary search procedure can be introduced to numerically search for a proper value of α_t [15].

2. An analytical solution of α_t can be calculated to minimize Z_t in the special case that h_t has only binary outputs $\{0, 1\}$. For example, if we construct the weak model as follows:

$$h_t(\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \text{if } (\mathbf{x}-\mathbf{y})^\top \mathbf{U}_t(\mathbf{x}-\mathbf{y}) \geq \beta_t, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where β_t is the threshold so that the resulting weak distance metric only differentiates each pair of objects as similar or dissimilar without further quantitative information about how similar or dissimilar they are. This choice of h_t may not be appropriate if there are no equivalence constraints, and only triplet conditions are available and required.

For such a h_t, the difference between any two distance measures reported by h_t, i.e., $h_t(\mathbf{x}_i,\mathbf{x}_j) - h_t(\mathbf{x}_i,\mathbf{x}_k)$, takes three possible values $\{-1, 0, 1\}$. Let I_+, I_0 and I_- denote, respectively, the sets of triplets satisfying $h_t(\mathbf{x}_i,\mathbf{x}_j) - h_t(\mathbf{x}_i,\mathbf{x}_k)$ equal to $+1$, 0, and -1. Then

$$\begin{aligned} Z_t &= \sum D_t((i,j,k))\exp(\alpha_t(h_t(\mathbf{x}_i,\mathbf{x}_j) - h_t(\mathbf{x}_i,\mathbf{x}_k))) \\ &= \sum\nolimits_{I_+} D_t((i,j,k))e^{\alpha_t} + \sum\nolimits_{I_0} D_t((i,j,k)) + \\ &\quad \sum\nolimits_{I_-} D_t((i,j,k))e^{-\alpha_t} \\ &= \epsilon_+ e^{\alpha_t} + \epsilon_0 + \epsilon_- e^{-\alpha_t} \end{aligned}$$

where ϵ_+, ϵ_0 and ϵ_- correspond to the specific splits of the summation over $D_t((i,j,k))$ according to I_+, I_0 and I_-, respectively. Hence $\epsilon_+ + \epsilon_0 + \epsilon_- = 1$. The function Z_t with respect to α_t has a unique minimizer $\alpha_t = \ln(\epsilon_-/\epsilon_+)/2$ which yields the value of $Z_t = \epsilon_0 + 2\sqrt{\epsilon_+\epsilon_-}$. Since $2\sqrt{\epsilon_+\epsilon_-} \leq \epsilon_+ + \epsilon_-$, $Z_t \leq 1$. If the weak model h_t at least outperforms a random guess, then $\epsilon_- > \epsilon_+$, and therefore $\alpha_t > 0$.

3. Analogous to the derivation of the original AdaBoost [6], we can also derive an analytical solution for α_t when the weak model h_t outputs values in $[0, 1]$, for instance, a probability output. In terms of learning a metric using PSD matrices, it may require a calibration or normalization of h_t to confine its range within $[0, 1]$. For such a h_t, the difference of the distance values, $h_t(\mathbf{x}_i,\mathbf{x}_j) - h_t(\mathbf{x}_i,\mathbf{x}_k)$ between any two pairs of objects ranges in $[-1, +1]$.

Due to the convexity of the function $e^{\alpha x}$ in terms of x for any constant $\alpha \in \Re$, the inequality $e^{\alpha x} \leq e^{\alpha}(1 + x)/2 + e^{-\alpha}(1 - x)/2$ holds when $x \in [-1, 1]$. For any real value of α_t, we can approximate Z_t by the upper bound

$$Z_t \leq e^{\alpha_t}\frac{1-r}{2} + e^{-\alpha_t}\frac{1+r}{2} \tag{4}$$

where

$$r = \sum_{(i,j,k)\in I_{tr}} D_t((i,j,k))\left(h_t(\mathbf{x}_i,\mathbf{x}_k) - h_t(\mathbf{x}_i,\mathbf{x}_j)\right). \tag{5}$$

The formula in Eq.(4) can be minimized when $\alpha_t = \ln((1+r)/(1-r))/2$ which corresponds to $Z_t \leq \sqrt{1-r^2}$. Obviously, $Z_t \leq 1$ and if $r > 0$, $\alpha_t > 0$. Furthermore, the inequality implies that we can achieve smaller Z_t by minimizing its upper bound $\sqrt{1-r^2}$. Hence, a weak learner can be designed to maximize $|r|$ for a sensible model h_t.

3.2. Weak Learners

The ultimate goal of metric learning is to construct a metric that preserves the overall proximity relations among the objects. For example, a metric d is desired if $d(\mathbf{x}_i,\mathbf{x}_k) > d(\mathbf{x}_i,\mathbf{x}_j)$ for any triplet (i,j,k) where the $\mathbf{x}_i$ is more similar to $\mathbf{x}_j$ than to $\mathbf{x}_k$, or if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "similar" while $\mathbf{x}_i$ and $\mathbf{x}_k$ are "dissimilar". The variable r defined in Eq.(5) provides a quantitative measure, in the same spirit to *margin*, about the difference between magnitude distance of pairs from different classes and that of pairs from same classes. The traditional error measure $\sum D_t((i,j,k))\mathbf{1}_{(h_t(\mathbf{x}_i,\mathbf{x}_k)>h_t(\mathbf{x}_i,\mathbf{x}_j))}$ determines a relatively qualitative measure about how many of the required triplet conditions are satisfied by h_t. Both measures shed a light on how to optimize a weak model h_t.

We focus on the maximization of the quantitative measure $|r|$ for h_t, which can be equivalently formulated as the following optimization problem at each iteration:

$$\begin{aligned}\max_{\substack{\mathbf{U}_t = \mathbf{u}_t\mathbf{u}_t^\top \\ ||\mathbf{u}_t|| = 1}} \quad & |\textstyle\sum_{(i,j,k)\in I_{tr}} D_t((i,j,k))(h_t(\mathbf{x}_i,\mathbf{x}_k) - \\ & h_t(\mathbf{x}_i,\mathbf{x}_j))| \\ \text{subject to} \quad & h_t(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^\top \mathbf{U}_t(\mathbf{x}-\mathbf{y}) \end{aligned} \tag{6}$$

Using simple matrix algebraic operations, the objective function of problem (6) can be rewritten as

$$\begin{aligned}|\mathbf{u}_t^\top[\textstyle\sum_{(i,j,k)\in I_{tr}} D_t((i,j,k))((\mathbf{x}_i-\mathbf{x}_k)(\mathbf{x}_i-\mathbf{x}_k)^\top - \\ (\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top)]\mathbf{u}_t|\end{aligned} \tag{7}$$

The problem of maximizing the objective (7) subject to a normalization constraint $||\mathbf{u}_t|| = 1$ has a closed-form solution: the optimal $\mathbf{u}_t$ is the eigenvector corresponding to the eigenvalue λ, which has the largest absolute value, of the matrix

$$\begin{aligned}\textstyle\sum_{(i,j,k)\in I_{tr}} D_t((i,j,k))((\mathbf{x}_i-\mathbf{x}_k)(\mathbf{x}_i-\mathbf{x}_k)^\top - \\ (\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top)\end{aligned} \tag{8}$$

and r corresponds to λ. Notice that if $r < 0$, the resulting weak model h_t serves better as a similarity measure rather than a distance measure and a negative α_t is obtained. In this case, the combined matrix $\mathbf{M} = \sum_t \alpha_t \mathbf{u}_t\mathbf{u}_t^\top$ is not necessarily PSD and thus the function $H(x,y) = (x-y)^\top\mathbf{M}(x-y)$ will be a non-metric distance function. If the task is to compare the relative distance between object pairs instead of forming a metric, it is reasonable to combines all distance measures with positive α_t and similarity measures with negative α_t for the final hypothesis.

The weak model $h_t(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^\top\mathbf{u}_t\mathbf{u}_t^\top(\mathbf{x}-\mathbf{y})$ can output any real value. To restrict the weak model h_t to predict only binary outputs $\{0,1\}$ so that we can use analytical α_t as discussed in Option 2 of Section 4, we determine a threshold β_t to cut the real value $(\mathbf{x}-\mathbf{y})^\top\mathbf{U}_t(\mathbf{x}-\mathbf{y})$ into two sets. Here, we assume h_t follows univariate Gaussian distributions over the set of all pairs of objects that are "similar", and the set of pairs of objects that are "dissimilar". Then the variable β_t can be analytically evaluated through the means and standard deviations of the two sets. Similar techniques used in previous works such as [19] can be adapted to our setting in determining β_t. The final hypothesis H can be evaluated as $\sum_t \alpha_t \mathrm{sgn}\left((\mathbf{x}-\mathbf{y})^\top\mathbf{U}_t(\mathbf{x}-\mathbf{y}) \geq \beta_t\right)$ for any pair of examples $(\mathbf{x},\mathbf{y})$.

To confine the range of h_t within $[0,1]$, we can normalize or calibrate the predictions of h_t on the pair-wise data. There exist general ways to calibrate the range of h_t. In our implementation, we used a simple normalization scheme by assuming that the training examples $\mathbf{X}$ cover the span of the sample population space. It is different from the assumption that training data represent the entire population and just assumes that the maximum distance $||\mathbf{x}-\mathbf{y}||$ among all possible pairs of training examples $(\mathbf{x},\mathbf{y})$ is a reasonable factor to be used in normalizing the h_t outputs. Let $C = \max\{||\mathbf{x}-\mathbf{y}|| \mid \mathbf{x} \neq \mathbf{y}, \mathbf{x}, \mathbf{y} \in \mathbf{X}\}$ which is a constant for a given set of data.

$$\begin{aligned} h_t(\mathbf{x},\mathbf{y}) &= \frac{(\mathbf{x}-\mathbf{y})^\top\mathbf{u}_t\mathbf{u}_t^\top(\mathbf{x}-\mathbf{y})}{C^2} \\ &\leq \frac{||\mathbf{x}-\mathbf{y}||^2||\mathbf{u}_t||^2}{C^2} \leq 1 \end{aligned} \tag{9}$$

Notice that the proximity relations are invariant to rescaling. Hence, the final hypothesis H can still be represented as $(\mathbf{x}-\mathbf{y})^\top\left(\sum \alpha_t\mathbf{U}_t\right)(\mathbf{x}-\mathbf{y})$.

4. Speed-up MetricBoost

In general, MetricBoost requires $O(|I_{tr}|)$ space and time per iteration, which might be computationally prohibitive or intractable in large scale applications. We employ the bipartite strategy similarly used in RankBoost [5] to develop a more efficient implementation of MetricBoost for proximity preserving over triplets. Especially when underlying problem is a classification problem, and triplets are derived based on the class labels, our bipartite MetricBoost can be of an immediate benefit to reduce computational cost.

Without loss of generality, let us assume that there are two underlying classes of objects, sample data $\mathbf{X}_1$ and $\mathbf{X}_2$ are collected for each of the classes. Denote $|\mathbf{X}|$ the cardinality of the set $\mathbf{X}$ (without notational ambiguity, $\mathbf{X}$ also denotes the data matrix collected for the set.) Each triplet (i,j,k) can be decomposed into two pairs (i,j) and (i,k). Then only two types of pairs of examples exist: S_1, the set of (i,j) pairs, and S_2, the set of (i,k) pairs.

For the set $\mathbf{X}_1$, totally $|\mathbf{X}_1|(|\mathbf{X}_1|-1)|\mathbf{X}_2|$ triplets can be formed. Overall, the number of the triplets over the two classes is $|I_{tr}| = |\mathbf{X}_1|(|\mathbf{X}_1|-1)|\mathbf{X}_2| + |\mathbf{X}_2|(|\mathbf{X}_2|-1)|\mathbf{X}_1|$. However, there are only $C^2_{|X_1|} + C^2_{|X_2|}$ pairs of examples that are in the same class, and totally $|\mathbf{X}_1||\mathbf{X}_2|$ possible pairs of examples that have different class labels. Naive implementation of MetricBoost requires, at each iteration, space and time costs in the order of $|\mathbf{X}_1|^2|\mathbf{X}_2| + |\mathbf{X}_2|^2|\mathbf{X}_1|$. The bipartite MetricBoost requires only $O(C^2_{|X_1|} + C^2_{|X_2|} + |\mathbf{X}_1||\mathbf{X}_2|)$ space and time per iteration. When $|\mathbf{X}| \geq 3$, the bipartite implementation saves costs dramatically.

Instead of maintaining a distribution $D_t((i,j,k))$over the triplets at each round, we assume that the distribution at round t can be decomposed into two separate parts.

$$D_t((i,j,k)) = \mu_t((i,j))\mu_t((i,k)) \tag{10}$$

This assumption is easily met for the first round by setting $\mu_1((i,j)) = \mu_1((i,k)) = 1/\sqrt{m}$ so that $D_1((i,j,k)) = 1/m$ over all triplets. Based on distribution update rule (2), the decomposition also holds for round $t+1$:

$$\begin{aligned} & D_{t+1}((i,j,k)) \\ = \; & D_t((i,j,k)) \exp(\alpha_t(h_t(\mathbf{x}_i,\mathbf{x}_j) - h_t(\mathbf{x}_i,\mathbf{x}_k)))/Z_t \\ = \; & (\mu_t((i,j)) \exp(\alpha_t h_t(\mathbf{x}_i,\mathbf{x}_j))/\sqrt{Z_t}) \cdot \\ & (\mu_t((i,k)) \exp(-\alpha_t h_t(\mathbf{x}_i,\mathbf{x}_k))/\sqrt{Z_t}) \\ = \; & \mu_{t+1}((i,j))\mu_{t+1}((i,k)) \end{aligned}$$

Hence the update rule in (2) can be revised accordingly. First, we calculate the non-normalized $\tilde{\mu}_{t+1}((i,j))$ and $\tilde{\mu}_{t+1}((i,k))$:

$$\begin{aligned} \tilde{\mu}_{t+1}((i,j)) &= \mu_t((i,j)) \exp(\alpha_t h_t(\mathbf{x}_i,\mathbf{x}_j)) \\ & \quad (i,j) \in S_1 \\ \tilde{\mu}_{t+1}((i,k)) &= \mu_t((i,k)) \exp(-\alpha_t h_t(\mathbf{x}_i,\mathbf{x}_k)) \\ & \quad (i,k) \in S_2 \end{aligned}$$

Then we calculate the normalization factor $Z_t = \sum_{(i,j,k)\in I_{tr}} \mu_{t+1}(\mathbf{x}_i,\mathbf{x}_j)\mu_{t+1}(\mathbf{x}_i,\mathbf{x}_k)$ and obtain the normalized $\mu_{t+1} = \tilde{\mu}_{t+1}/\sqrt{Z_t}$. As described above, by decomposing the triplet distribution $D_t((i,j,k))$ into pairs $\mu_t((i,j))$ and $\mu_t((i,k))$, the computational and memory cost can be significantly reduced from $O(|\mathbf{X}_1|^2|\mathbf{X}_2| + |\mathbf{X}_2|^2|\mathbf{X}_1|)$ to $O(C^2_{|X_1|} + C^2_{|X_2|} + |\mathbf{X}_1||\mathbf{X}_2|)$.

In order to construct weak models, the matrix (8) needs to be evaluated, and as it can be decomposed into the following four components, the computational cost is only $O(C^2_{|X_1|} + C^2_{|X_2|} + |\mathbf{X}_1||\mathbf{X}_2|)$:

$$\begin{aligned} \textstyle\sum_{(i,j,k)\in I_{tr}} \quad & \mu_t((i,j))\mu_t((i,k))((\mathbf{x}_i - \mathbf{x}_k) \\ & (\mathbf{x}_i - \mathbf{x}_k)^\top - (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top) \\ = \textstyle\sum_{i\in X_1,(i,k)\in S_2} \quad & \mu_t((i,k))s_1(i)(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^\top \\ - \textstyle\sum_{i,j\in X_1} \quad & \mu_t((i,j))w_1(i)(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \\ + \textstyle\sum_{i\in X_2,(i,k)\in S_2} \quad & \mu_t((i,k))s_2(i)(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^\top \\ - \textstyle\sum_{i,j\in X_2} \quad & \mu_t((i,j))w_2(i)(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \end{aligned}$$

where the parameters $s_1(i) = \sum_{j\in X_1, j\neq i} \mu_t((i,j))$, $s_2(i) = \sum_{j\in X_2, j\neq i} \mu_t((i,j))$, $w_1(i) = \sum_{k\in X_2} \mu_t((i,k))$, and $w_2(i) = \sum_{k\in X_1} \mu_t((i,k))$.

5. Experiments

We validate MetricBoost on publicly available benchmark data sets together with two real-world computer-aided-diagnosis (CAD) problems of detecting abnormal structures from medical images. Effectively solving CAD problems has been the driving force for developing the proposed method. We compare MetricBoost with other state-of-the-art metric learning methods: including information theoretic metric learning (ITML) [3], BoostMetric method [17, 16], convex optimization (COP) metric learning method [4]. Standard distance metrics, such as Mahalanobis, ℓ_1 and ℓ_2 norm distances are also used as baseline in some experiments. Two measures have been used to evaluate the performance in the test phase: the classification accuracy of k-Nearest Neighbor ($k = 1$), and percentage of triplets that preserves the class-implied proximity relationship, that is, whether two points in the same class are closer than any one of them to a third point from a different class.

5.1. Benchmark data

The first set of experiments was conducted on six benchmark data sets from the Machine Learning Repository at University of California, Irvine (UCI)[1]. In this experiment, the metric learning methods were evaluated via five-fold cross validation and results were averaged over 40 runs. For MetricBoost, both binary and normalized weak models h_t were implemented and compared. The number of iterations T was set to 20 which was the same in PSDBoost for fair comparison. The slack variable γ in ITML and the weight parameter C in PSDBoost were tuned over the values $\{0.01, 0.1, 1, 10\}$ using a separate cross validation. To test the accuracy of these algorithms on training triplets or pairs, we used all triplets or pairs formed from the training data, and a small subset of them (5% pairs and 0.5% triplets) randomly sampled from the full set to train distance metrics.

Testing results on triplet relationships and k-NN classification for various data sets are summarized in Tables 1 and 2. MetricBoost is the only algorithm that achieves the top accuracy across all datasets in both measurements. In general, the performance improves for all metric learning methods with more triplet or pair constraints in training available. However, both MetricBoost and ITML are less sensitive to changes in the number of triplet or pair constraints than COP. For MetricBoost, the binary weak model seems to provide better performance than the normalized weak model, especially in k-NN ($k = 3$) classification as shown in Table 2. It suggests that binary models are probably more appropriate on qualitative labels of data because MetricBoost is based on proximity comparison which does not demand quantitative measure on distance of pairs.

5.2. Diffuse lung disease

MetricBoost has been deployed to a computer-aided-diagnosis (CAD) system which detects diffuse parenchymal lung diseases from CT images. In the related experiment, we have $22,923$ samples in the lung dataset annotated by

[1] $http://www.ics.uci.edu/\_mlearn/MLRepository.html.$

Table 1. Percentage of correctly identified triplet distance relationship via different learned metrics. ‘A’ stands for training with all triplets or pairs if ITML and ‘S’ for subset triplets or pairs. Both the average and stand deviation of the testing results are listed.

	WINE	SPECT	SONAR	IONOSPHERE	BRST CANCER	DIABETES
Euclidean (ℓ_2)	71.0 (3.7)	58.7 (4.0)	51.5 (1.5)	62.4 (1.4)	82.2 (1.7)	53.5 (1.0)
Mahalanobis	60.2 (2.6)	35.2 (2.8)	51.9 (3.0)	65.2 (1.1)	58.9 (1.9)	54.0 (1.2)
Binary MetricBoost (A)	91.4 (2.6)	67.0 (6.8)	58.9 (4.1)	74.3 (5.1)	87.7 (1.7)	55.8 (1.9)
Binary MetricBoost (S)	90.5 (3.0)	65.4 (6.2)	59.7 (4.7)	73.7 (5.2)	87.6 (2.4)	55.4 (2.1)
Normalized MetricBoost (A)	82.7 (3.6)	71.9 (3.9)	58.9 (4.3)	71.2 (4.2)	85.9 (2.2)	53.7 (2.2)
Normalized MetricBoost (S)	85.0 (4.0)	69.5 (6.5)	60.3 (4.0)	72.8 (5.5)	85.7 (2.6)	54.8 (2.0)
ITML (A)	80.0 (4.5)	49.0 (6.0)	54.4 (2.9)	72.5 (3.5)	82.6 (2.1)	56.6 (1.4)
ITML (S)	71.1 (3.4)	54.1 (8.1)	54.9 (3.3)	72.1 (3.5)	82.5 (1.7)	55.9 (1.5)
COP (A)	88.6 (2.8)	48.7 (6.0)	59.1 (3.6)	67.9 (1.5)	80.6 (2.8)	54.1 (1.5)
COP (S)	70.3 (7.1)	44.9 (3.7)	53.7 (2.9)	57.0 (3.6)	67.9 (4.8)	51.5 (1.1)
BoostMetric (S)	91.4 (3.1)	70.8 (3.9)	58.0 (3.3)	73.2 (5.6)	85.7 (2.8)	54.2 (2.4)

Table 2. k-NN ($k = 1$) classification accuracy rate via different learned metrics.

	WINE	SPECT	SONAR	IONOSPHERE	BRST CANCER	DIABETES
Euclidean (ℓ_2)	72.0 (3.7)	72.0 (3.3)	70.3 (3.6)	82.4 (2.3)	90.8 (1.3)	66.4 (2.0)
Mahalanobis	89.0 (3.3)	62.4 (5.0)	62.4 (4.4)	73.4 (2.7)	81.2 (2.4)	67.4 (1.6)
Binary MetricBoost (A)	96.8 (1.7)	75.6 (3.2)	74.4 (3.3)	85.1 (7.5)	95.1 (1.2)	68.6 (1.8)
Binary MetricBoost (S)	96.6 (2.2)	74.8 (2.9)	71.2 (4.2)	83.4 (6.4)	94.9 (1.4)	67.1 (2.4)
Normalized MetricBoost (A)	89.7 (5.6)	75.3 (3.1)	72.1 (4.2)	83.0 (3.2)	92.4 (1.9)	67.4 (2.5)
Normalized MetricBoost (S)	92.0 (5.3)	75.3 (3.6)	69.2 (4.7)	82.6 (3.0)	93.0 (1.9)	66.3 (2.7)
ITML (A)	92.0 (3.0)	70.6 (3.4)	73.4 (4.8)	83.3 (2.4)	92.9 (1.3)	68.8 (2.0)
ITML (S)	76.0 (6.0)	71.9 (3.7)	72.9 (4.2)	83.5 (2.8)	92.4 (1.3)	68.3 (1.7)
COP (All)	94.7 (1.8)	71.0 (3.5)	72.3 (4.2)	80.5 (3.8)	91.7 (1.7)	66.6 (2.2)
COP (Subset)	78.4 (7.0)	69.2 (3.8)	70.6 (4.8)	80.8 (2.6)	84.0 (4.2)	63.1 (2.0)
BoostMetric	96.4 (1.9)	75.9 (3.2)	68.7 (7.3)	83.0 (5.9)	91.5 (1.9)	63.9 (3.1)

two expert radiologists as one of the three classes: healthy, emphysema and fibrosis. The dataset is divided into training dataset and test dataset. The training dataset contains $15,155$ samples, of which $8,067$ healthy samples, $4,988$ emphysema samples and $2,100$ fibrosis samples. The testing dataset contains $7,768$ samples, of which $3,504$, $2,710$ and $1,554$ samples belong to class healthy, emphysema and fibrosis, respectively. Each sample is described by 43 features and thus lies in $\mathbf{R}^{43}$. For all metric learning methods and Mahalnobis distance, we train a metric using a subset of the training samples, and test on the entire test dataset. In the test phase, the label of a test sample is determined by majority voting of the k nearest neighbors in the training set according to the respective metric. The comparison results are shown in Fig. 1 and our methods outperform other ones.

We compare the computational efficiency of MetricBoost and Mathematical Programming based method BoostMetric which both use triplet constraints to learn the distance metric. Table 3 shows the training time of both methods using the same subset of triplet constraints. The tests were run on a dual 2.0GHz Intel pentium processor running Windows XP and averaged over 40 runs. The proposed MetricBoost was implemented following Algorithm 1 and BoostMetric was obtained from its original author[2]. Based on Table 3, MetricBoost constructs metrics faster than BoostMetric because MetricBoost decomposes triplet constraints into pairs and directly work on the pairs. It has no tuning parameter to specify via cross validation.

5.3. Colorectal polyp matching between prone and supine views

When using CT images to diagnose colorectal cancer, each patient is normally imaged twice with different poses (prone or supine), to maximize the chance that a polyp can be found by radiologists or computers. Due to the gravity and deformability of tissues together with some imaging conditions (e.g., liquid/solid tagging), a polyp can appear visually distinct between the prone and supine views (analogical to wide-baseline stereo matching in computer vision). Due to the practical difficulty, most colon cancer CAD systems are not capable of matching the polyps de-

[2] BoostMetric code available at http://code.google.com/p/boosting/.

Table 3. Training time (in seconds) of the proposed method and BoostMetric.

Methods	Breast	Diabetes	Ionosphere	Spect	Wine	Sonar	Lung
BoostMetric	16.2	50.9	22.7	6.4	20.0	8.04	201.82
Proposed	9.6	13.3	4.1	2.5	4.9	3.1	22.1

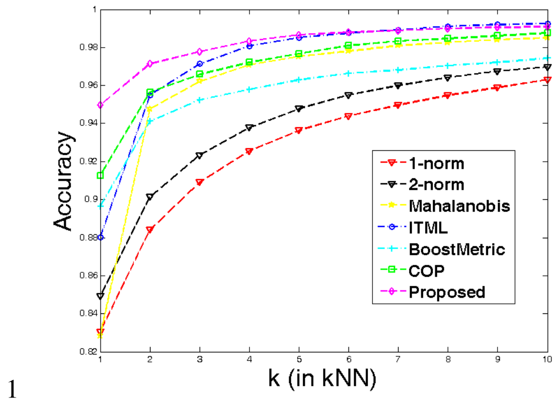


Figure 1. classification accuracy versus different choices of Ks in KNN using ℓ_2, ℓ_1, Mahalanobis [1], ITML [3], BoostMetric [16], COP [4], and our proposed method. Results are evaluated based on testing datsset.

tected from either of the views. We design a system based on MetricBoost to retrieve corresponding polyps from the other view for each polyp detected in one of the views.

When building a computer system to detect polyps, a classifier will be constructed based on a set of numerical image features and used to generate a receiver operating characteristic (ROC) curve, and the regular operating point is around 2 false positive rate per volume. We move the operating point to a larger false positive rate (7.6/volume) to retain high sensitivity (95%) of polyp detection. In training, our computer aided polyp detection algorithm operates at a sensitivity of 94.63% and false positive (FP) rate of 7.586 per volume on average as opposed to normal operating point at FP rate ≈ 2.0. The augmented set of candidates is used to form the triplets for metric learning in the following way. For each positive polyp instance C_i in the prone view of a patient, we find the positive instances[3] $\{C_j\}_{j=1}^n$ of the same polyp and all other candidates (including positives corresponding to other polyps and negatives, or FPs) $\{C_k\}_{k=1}^m$ in the supine view, and (i,j,k) is a triplet, requiring $d(C_i,C_j) < d(C_i,C_k)$. Then we repeat the same process on each positive instance in the supine view to build more triplets. Finally, the PSD distance metric matrix M is learned subject to the constraint of the built triplets.

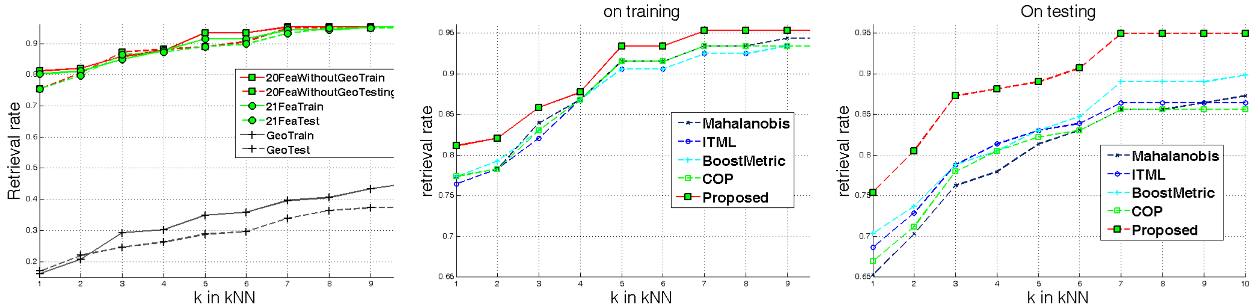
Totally, 96 appearance features, including local geometric features, morphological, shape/intensity and context features, are extracted for each candidate to distinguish between polyps and FPs. A feature selection method is used to choose the most relevant features for the purpose of polyp matching between views. We apply the minimum redundancy maximum relevance (MRMR) feature selection framework [12] and make use of the t-statistic to measure feature relevance and Pearson's correlation to measure feature redundancy, and finally select 20 features.

We treat the matching problem as a polyp retrieval process: for each polyp $\mathbf{x}$ detected in one view, find the polyp $\mathbf{y}$ in the other view that matches $\mathbf{x}$ according to the distance metric $H(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^\top \mathbf{M}(\mathbf{x}-\mathbf{y})$. For quantitative performance evaluation, we list the top k match points $\mathbf{y}$ with shortest distance to $\mathbf{x}$ and evaluate if any true match is within the k points; if the true match is among them, then a "hit" will occur; otherwise, there is no hit. We record the retrieval rate, which is defined as the percentage of polyps being hit within the k match points (closest in feature distances). The results are shown in Fig. 2(a), which demonstrates substantial performance improvement over geodesic distance indexing method. The geodesic distance indexing ($[0,1]$ from rectum to cecum of colon) is the current main approach [21] for polyp matching, but does not estimate accurately when colon is collapsed. Our method uses purely local discriminative features from polyp classification which is more robust. Adding geodesic distance features into MetricBoost does not further improve the performance. The metric-learning-based polyp matching approach is also much more computationally efficient than the state-of-the-art global/local surface registration methods [10, 14], with comparable or better matching accuracy. We also compared the proposed technique against other metric learning methods [1, 17, 3, 16, 4] on this problem. From Fig. 2(b)(c), our method consistently shows superior matching/retrieval accuracies in both training and test, and generalizes better to unseen test cases. The proposed technique may be applicable to wide-baseline stereo matching or object view matching problems.

6. Conclusion

We have sketched a formal framework and an efficient algorithm for the problem of constructing a proper distance metric by combining many weak models each characterized by a rank-one PSD matrix. The proposed boosting algorithm can have several variants depending on the specific

[3] A polyp can be fragmented into several candidates in detection, which is known as multiple instance learning problem [7].



(a) (b) (c)

Figure 2. Performance on polyp matching as retrieval one polyp from candidates in the other view. (Left) geodesic distance comparison; (middle) other ML methods comparison in training; (right) other ML methods comparison in test.

criterion for choosing weak metrics h_t and α_t. Our algorithm can be regarded as an effective solver to specific positive semi-definite programs related to metric learning. The proposed metric learning method can effectively solve two real-world medical problems. Computational results demonstrate the effectiveness of MetricBoost in the k-NN classification and triplet proximity preservation on multiple benchmark datasets and the medical problems of distinguishing several diffuse lung diseases as well as polyp matching between views. We plan to extend the work by further examining the generalization error bounds and selecting the most representative subset of triplets.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *JMLR*, 6:937–965, 2005. 2617, 2623

[2] A. Bar-Hillel and D. Weinshall. Learning distance functions by coding similarity. In *ICML*, 2007. 2617

[3] J. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007. 2617, 2621, 2623

[4] M. J. E.P Xing, A.Y. Ng and S. Russell. Distance metric learning, with applicationto clustering with side-information. In *NIPS*, volume 15, pages 505–512, 2003. 2617, 2621, 2623

[5] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003. 2620

[6] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997. 2619

[7] G. Fung, M. Dundar, B. Krishnapuram, and B. Rao. Multiple instance learning for computer aided diagnosis. In *NIPS*, pages 425–432, 2006. 2623

[8] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996. 2617

[9] X. He, O. King, W. Ma, M. Li, and H.-J. Zhang. Learning a semantic space from users relevance feedback for image retrieval. *IEEE Trans. on CSVT*, 13(1):39–48, 2003. 2617

[10] Z. Lai and et al. Intra-patient supine-prone colon registration in ct colonography using shape spectrum. In *MICCAI*, pages 332–339, 2010. 2623

[11] H. Müller, T. Pun, and D. Squire. Learning from user behavior in image retrieval: application of market basket analysis. *International Journal of Computer Vision*, 56(1-2):65–77, 2004. 2617

[12] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005. 2623

[13] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, 2006. 2617

[14] H. Roth and et al. Establishing spatial correspondence between the inner colon surfaces from prone and supine ct colonography. In *MICCAI*, pages 497–504, 2010. 2623

[15] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999. 2619

[16] C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning with boosting. *Advances in Neural Information Processing Systems*, 22, 2009. 2617, 2618, 2621, 2623

[17] C. Shen, A. Welsh, and L. Wang. PSDBoost: matrix generation linear programming for positive semidefinite matrices learning. *Advances in Neural Information Processing Systems*, 21, 2008. 2617, 2618, 2621, 2623

[18] S. Xiang, F. Nie, and C. Zhang. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600–3612, 2008. 2617

[19] J. Yu, J. Amores, N. Sebe, P. Radeva, and Q. Tian. Distance learning for similarity estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3):451–462, 2008. 2620

[20] S. Zhou, J. Shao, B. Georgescu, and D. Comaniciu. Boostmotion: boosting a discriminative similarity function for motion estimation. In *IEEE Conf. on CVPR*, 2006. 2617

[21] H. Zhu, Y. Fan, H. Lu, and Z. Liang. Improving initial polyp candidate extraction for ct colonography. *Phys Med Biol.*, 55(7):2087C2102, 2010. 2623