Semi-Supervised Sparse Metric Learning Using Alternating Linearization Optimization

Wei Liu
Columbia University
New York, NY, USA
wl2223@columbia.edu

Shiqian Ma
Columbia University
New York, NY, USA
sm2756@columbia.edu

Dacheng Tao
Nanyang Technological University
Singapore
dctao@ntu.edu.sg

Jianzhuang Liu
The Chinese University of Hong Kong
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences, China
jzliu@ie.cuhk.edu.hk

Peng Liu
Barclays Capital
New York, NY, USA
liup1024@gmail.com

ABSTRACT

In plenty of scenarios, data can be represented as vectors and then mathematically abstracted as points in a Euclidean space. Because a great number of machine learning and data mining applications need proximity measures over data, a simple and universal distance metric is desirable, and metric learning methods have been explored to produce sensible distance measures consistent with data relationship. However, most existing methods suffer from limited labeled data and expensive training. In this paper, we address these two issues through employing abundant unlabeled data and pursuing sparsity of metrics, resulting in a novel metric learning approach called *semi-supervised sparse metric learning*. Two important contributions of our approach are: 1) it propagates scarce prior affinities between data to the global scope and incorporates the full affinities into the metric learning; and 2) it uses an efficient alternating linearization method to directly optimize the sparse metric. Compared with conventional methods, ours can effectively take advantage of semi-supervision and automatically discover the sparse metric structure underlying input data patterns. We demonstrate the efficacy of the proposed approach with extensive experiments carried out on six datasets, obtaining clear performance gains over the state-of-the-arts.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

KDD'10, July 25–28, 2010, Washington, DC, USA.

General Terms

Algorithms

Keywords

Metric learning, semi-supervised sparse metric learning, sparse inverse covariance estimation, alternating linearization

1. INTRODUCTION

Vectored data frequently occur in a variety of fields, which are easy to handle since they can be mathematically abstracted as points residing in a Euclidean space. An appropriate distance metric in this space spanned by input data vectors is quite demanding for a great number of applications including classification, clustering and retrieval. Two most commonly used distance metrics are Euclidean distance and Mahalanobis distance, of which the former is independent of the input data while the latter is related to second-order statistics of the input data. In practice, we need to seek distance metrics suitable for the requirements of different tasks.

In the context of classification, distance metrics are frequently applied in concert with kNN classifiers. As such, the goal of metric learning towards kNN classification is to keep the distances among nearby points as small as possible and push the differently labeled neighbors out of the neighborhood of any of these points. Neighbourhood Components Analysis (NCA) [11] and its seminal work Maximally Collapsing Metric Learning (MCML) [10] emphasize that the target metric should support tight neighborhoods and even reach zero distances within all neighborhoods. Like the notion of SVMs, Large Margin Nearest Neighbor classification (LMNN) [28] not only narrows the distance gap within all neighborhoods, but also maximizes the soft margin of distances over each neighborhood.

As for clustering, metric learning usually cooperates with constrained clustering, namely, semi-supervised clustering [26][3][15] where some background knowledge about data proximities is given beforehand. Specifically, two kinds of

pairwise links, i.e., *must-links* and *cannot-links*, are given. The must-links indicate that two data points must be in the same cluster, while the cannot-links require that two data points not be grouped into the same cluster. Therefore, the purpose of metric learning applied to semi-supervised clustering is to minimize the distances associated with must-links and simultaneously maximize the distances associated with cannot-links. There have been some works [29][4][8][25] which are engaged in learning metrics towards better clusterings.

In the field of content-based image retrieval (CBIR), choosing appropriate distance metrics plays a key role in establishing effective systems. Regular CBIR systems usually adopt the Euclidean distance measure for images represented in a vector form. Unfortunately, Euclidean distance is generally not effective enough in retrieving relevant images. A main reason stems from the well-known semantic gap between low-level visual features and high-level semantic concepts [24]. The commonly used relevance feedback scheme [23] may remedy the semantic gap issue, which produces, aided by users, a set of pairwise constraints about relevance (similarity) or irrelevance (dissimilarity) between two images. These constraints along with involved image examples are called *log data*. Then the key to CBIR is to find an effective way of utilizing the log data in relevance feedback so that the semantic gap can be successfully reduced. A lot of ways have been studied to utilize the log data to boost the performance of CBIR. In particular, one can use a metric learning technique devoted to semi-supervised clustering for tackling CBIR since these relevance constraints are essentially must-links and cannot-links. The recent works [2][14][18] have recommended learning proper distance metrics for image retrieval tasks.

In this paper, we pose metric learning under the semi-supervised setting where only a few pairwise constraints including similar and dissimilar exist and most data instances are not involved in such constraints. We propose a novel metric learning technique called *semi-supervised sparse metric learning* (S^3ML). The major features of S^3ML include: 1) it is capable of propagating scarce pairwise constraints to all data pairs; 2) it generates a sparse metric matrix which coincides with the sparse spirit of feature correlations in the high-dimensional feature space; and 3) it is quite efficient by using the *alternating linearization method* in contrast to existing metric learning approaches using expensive optimizations such as semidefinite programming. The proposed S^3ML technique has widespread applicability without being limited to particular backgrounds. Quantitative experiments are performed for classification and retrieval tasks, uncovering the effectiveness and efficiency of S^3ML.

The remainder of this paper is arranged as follows. Section 2 reviews the related work on recent metric learning. Section 3 describes and addresses the semi-supervised metric learning problem. Section 4 presents the S^3ML algorithm by using the alternating linearization optimization method. Section 5 validates the efficacy of the proposed S^3ML through extensive experiments. Section 6 includes conclusions.

2. RELATED WORK

In recent years, there are some emerging research interests in learning data representations in some intrinsic low-dimensional space embedded in the ambient high-dimensional space such that regular Euclidean distance is more meaningful in the low-dimensional space. The early efforts are learning linear representations by Principal Component Analysis (PCA) and learning nonlinear representations by manifold learning. However, these methods are unsupervised and loosely related to the distance outcome. This paper investigates distance metric learning which is vital to a lot of machine learning and data mining applications. The recent metric learning research can be classified into three main categories as follows.

2.1 Supervised Metric Learning

The first category is supervised metric learning approaches for classification where distance metrics are usually learned from training data associated with explicit class labels. The representative techniques include Neighbourhood Components Analysis (NCA) [11], Maximally Collapsing Metric Learning (MCML) [10], and metric learning for Large Margin Nearest Neighbor classification (LMNN) [28]. Nevertheless, the performance of these supervised approaches rests highly on the amount of labeled data that are often practically difficult and expensive to gather. Moreover, all of them request nontrivial optimizations such as semidefinite programming [5], which is inefficient for real-world datasets.

2.2 Weakly Supervised Metric Learning

Our work is closer to the second category of weakly supervised metric learning which learns distance metrics from pairwise constraints present in input data, or known as *side information* [29]. It is manifest that such side information is weaker than exact label information. In detail, each constraint indicates whether two data points are relevant (similar) or irrelevant (dissimilar) in a particular learning task. A well-known metric learning method with these constraints was proposed by Xing et al. [29], which casts the learning task into a convex optimization problem and applies the generated solution to data clustering. Following this work, there are several emerging metric learning techniques in the "weakly supervised" direction. For instance, Relevance Component Analysis (RCA) learns a global linear transformation by exploiting only the equivalent (relevant) constraints [2]. Recently, Information-Theoretic Metric Learning (ITML) [8][7] expresses the weakly supervised metric learning problem as a Bregman optimization problem where the pairwise constraints are treated as inequality constraints.

2.3 Sparse Metric Learning

In [21], to speedup the training time of metric learning, ℓ_1 regularization is incorporated into the original non-sparse metric learning objective, resulting in a much faster learning procedure: Sparse Distance Metric Learning (SDML). Besides, the sparsity of desirable metric matrices makes sense since the Mahalanobis matrix is nearly sparse under the high-dimensional data space. This sparsity spirit stems from the weak correlations among different feature dimensions in the high-dimensional feature space because most distinct features are measured by distinct mechanisms and relatively independent of each other. In another perspective, [22][30] attempt to learn a low-rank sparse metric matrix by inducing sparsity to a low-rank linear mapping whose outer product constitutes the sparse metric. Nonetheless, learning low-rank sparse metrics often implicates complex optimization

which has a large computational cost and is sensitive to the choice of the low rank as well.

3. SEMI-SUPERVISED METRIC LEARNING

In this section, we investigate the *semi-supervised metric learning* problem [14] which takes scarce pairwise constraints as input and, importantly, exploits abundant unlabeled data that are not involved in these constraints. Consequently, semi-supervised metric learning supplements the aforementioned weakly supervised metric learning due to the merit of accessing unlabeled data. Different from standard *semi-supervised learning* [31], semi-supervised metric learning does not need any class labels, so it can be applied to a broad spectrum of applications such as semi-supervised classification, semi-supervised clustering, relevance feedback based information retrieval, etc.

3.1 Problem Description

Assume that we are given a set of n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subseteq \mathbb{R}^m$ and two sets of pairwise constraints among these data points:

$$\begin{aligned}\mathcal{S} &= \{(i,j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are judged to be similar}\}\\ \mathcal{D} &= \{(i,j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are judged to be dissimilar}\},\end{aligned} \quad (1)$$

where $\mathcal{S}$ is the set of similar pairwise constraints, and $\mathcal{D}$ is the set of dissimilar pairwise constraints. Each pairwise constraint (i,j) indicates if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are relevant or irrelevant judged by users under some application context. Note that it is not necessary for all the points in $\mathcal{X}$ to be involved in $\mathcal{S}$ or $\mathcal{D}$.

For any pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$, let $d(\mathbf{x}_i, \mathbf{x}_j)$ denote the distance between them. To compute the distance, let $M \in \mathbb{R}^{m\times m}$ be a symmetric metric matrix. We can then express the distance measure as follows:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_M = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)}. \quad (2)$$

In practice, the symmetric matrix M is a valid metric if and only if it satisfies the non-negativity and the triangle inequality conditions. In other words, M must be positive semidefinite, i.e., $M \succeq 0$. Generally, the matrix M parameterizes a family of Mahalanobis distances on the vector space $\mathbb{R}^m$. As an extreme case, when setting M to be identity matrix $I \in \mathbb{R}^{m\times m}$, the distance in eq. (2) becomes the common Euclidean distance.

Definition 1. *The goal of semi-supervised metric learning is to learn an optimal symmetric matrix $M \in \mathbb{R}^{m\times m}$ from a collection of data points $\mathcal{X}$ on a vector space $\mathbb{R}^m$ together with a set of similar pairwise constraints $\mathcal{S}$ and a set of dissimilar pairwise constraints $\mathcal{D}$, which can be formulated as the following optimization prototype:*

$$\min_{M\succeq 0} \quad g(M, \mathcal{X}, \mathcal{S}, \mathcal{D}) \quad (3)$$

where M is maintained to be positive semidefinite and $g(\cdot)$ is a proper objective function defined over the given data and constraints.

Given the above definition, the strategy to attack metric learning is to first design an appropriate objective function g and then seek an efficient algorithm to minimize it. In the following, we discuss some principles for formulating reasonable optimization models. Importantly, we emphasize that it is critical for solving real-world metric learning problems to avoid overfitting.

Min-Max Principle. One common principle for metric learning is to minimize the distances among the data points with similar constraints and meanwhile to maximize the distances among the data points with dissimilar constraints. We refer to it as a *min-max* principle. Many existing metric learning works such as [29][28][14] can be interpreted via this min-max principle. Immediately, we can define g simply based on this principle:

$$g(M) = \sum_{(i,j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_M^2 - \gamma \sum_{(i,j)\in\mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_M^2, \quad (4)$$

where $\gamma > 0$ is the trade-off parameter. Although the above objective function has been shown effective for some clustering tasks, it may be unsuitable for the applications in which the pairwise constraints are scarce and most data instances are not involved in any constraint. As a matter of fact, optimizing the above g is likely to overfit the two limited constraint sets $\mathcal{S}$ and $\mathcal{D}$. Hence, we should incorporate all data in $\mathcal{X}$ into the design of g.

To facilitate the following derivations, we assume that there exits a linear mapping $U^\top : \mathbb{R}^m \to \mathbb{R}^r$ ($U = [\mathbf{u}_1, \ldots, \mathbf{u}_r] \in \mathbb{R}^{m\times r}$) such that $M = UU^\top$. We require that $\mathbf{u}_1, \ldots, \mathbf{u}_r$ be linearly independent so that r is the rank of the metric matrix M. Then the distance under M between two inputs can be computed as:

$$\begin{aligned}\|\mathbf{x}_i - \mathbf{x}_j\|_M &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)}\\ &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top UU^\top(\mathbf{x}_i - \mathbf{x}_j)}\\ &= \left\|U^\top(\mathbf{x}_i - \mathbf{x}_j)\right\|.\end{aligned}$$

Actually, the target metric M is usually low-rank in high-dimensional data spaces [7]. Thus, we may pursue the subspace U equivalently.

Affinity-Preserving Principle. To remedy overfitting, we aim at taking full advantage of unlabeled data that are demonstrated to be quite beneficial to the semi-supervised learning problem. Due to this consideration, we define g based on the notion of affinity preserving [18]. Given the collection of n data points $\mathcal{X}$ including the data involved in the given constraints and the unlabeled data, we need an affinity matrix $W \in \mathbb{R}^{n\times n}$ on $\mathcal{X}$ such that each entry W_{ij} measures the strength of affinity between data pair $\mathbf{x}_i$ and $\mathbf{x}_j$. The larger W_{ij}, the closer $\mathbf{x}_i$ and $\mathbf{x}_j$. Through absorbing all data information $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ guided by the affinity matrix W, we formulate g as follows:

$$\begin{aligned}&g(M, \mathcal{X}, \mathcal{S}, \mathcal{D})\\ &= \frac{1}{2}\sum_{i,j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|_M^2 W_{ij} = \frac{1}{2}\sum_{i,j=1}^n \left\|U^\top(\mathbf{x}_i - \mathbf{x}_j)\right\|^2 W_{ij}\\ &= \sum_{d=1}^r \mathbf{u}_d^\top X(D-W)X^\top \mathbf{u}_d = \sum_{d=1}^r \mathbf{u}_d^\top XLX^\top \mathbf{u}_d\\ &= tr\left(U^\top XLX^\top U\right) = tr\left(XLX^\top UU^\top\right) = tr(XLX^\top M),\end{aligned} \quad (5)$$

where $tr(\cdot)$ stands for the *trace* operator, and D is a diagonal matrix whose diagonal elements equal the sums of the row entries of W, i.e., $D_{ii} = \sum_{j=1}^n W_{ij}$. The matrix $L = D - W$ is known as the *graph Laplacian* [31].

Laplacian Regularized Metric Learning (LRML) [14] and SDML [21] follow the same affinity-preserving principle. Under supervised settings, it is quite simple to obtain the affinity matrix W by setting $W_{ij} = 1$ if $(i, j) \in \mathcal{S}$ and $W_{ij} = -1$ if $(i, j) \in \mathcal{D}$. However, under semi-supervised settings, there is no intuitive way to acquire W since $\mathcal{S}$ and $\mathcal{D}$ often contain very few data instances. As a weak measure of affinities, LRML defines W in an unsupervised fashion, i.e., $W_{ij} = 1$ if $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$ or vice versa. Such a definition for W fails to absorb the pairwise constraints and thus does not take full advantage of semi-supervision.

3.2 Affinity Propagation

We aim at designing better affinity matrices for semi-supervised settings through integrating the min-max principle and the affinity-preserving principle. Based on the weak affinities revealed by k-NN search, we intend to propagate the strong (definitely correct) affinities revealed by the given pairwise constraints to the global scope of the data. Let us define a neighborhood indicator matrix $P \in \mathbb{R}^{n\times n}$ on $\mathcal{X}$:

$$P_{ij} = \begin{cases} \frac{1}{k}, & j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathcal{N}_i$ denotes the index list composed of k nearest neighbors of data point $\mathbf{x}_i$ using the Euclidean distance. Note that P is asymmetric and provides weak affinities.

To enable metric learning techniques to work for practical applications, we should shrink the distances between as many similar pairs as possible and enlarge the distances between as many dissimilar pairs as possible. Although the affinity-preserving function g has incorporated all unlabeled data, it does not emphasize the distances between "real" similar or dissimilar data pairs. Since the two real constraint sets $\mathcal{S}$ and $\mathcal{D}$ are available, we desire to propagate the limited real constraints to all data pairs via the found neighborhoods P using the Euclidean distance. Specifically, we learn an affinity matrix W such that W_{ij} reflects the extent of propagated affinity between data pair $(\mathbf{x}_i, \mathbf{x}_j)$.

Let us begin with an initial affinity matrix $W^0 \in \mathbb{R}^{n\times n}$ where we set $W^0_{ii} = 1$ for any i, $W^0_{ij} = 1$ for any $(i, j) \in \mathcal{S}$, $W^0_{ij} = -1$ for any $(i, j) \in \mathcal{D}$, and $W^0_{ij} = 0$ otherwise. Clearly, W^0 represents the strong affinities. If we consider ± 1-entries in W^0 as signed energies, our intent is to propagate the energies in W^0 to its 0-entries. The propagation path coincides with the neighborhood structure at each data point, so we pose the affinity propagation criterion as the locally linear energy mixture, i.e.,

$$W^{(t+1)}_{i.} = (1-\alpha)W^{(0)}_{i.} + \alpha \sum_{j=1}^{n} P_{ij} W^{(t)}_{j.}, \tag{7}$$

where $W^{(t)}_{i.}$ denotes the ith row of $W^{(t)}$ and $t = 0, 1, 2, \cdots$ is the time stamp. We write the matrix form of eq. (7) as

$$W^{(t+1)} = (1-\alpha)W^{(0)} + \alpha P W^{(t)}, \tag{8}$$

where $0 < \alpha < 1$ is the trade-off parameter and P is like the transition probability matrix widely used in Markov random walk models. Because $0 < \alpha < 1$ and the eigenvalues of P are in $[-1, 1]$, the limit $W^* = \lim_{t\to\infty} W^{(t)}$ exists. It suffices to solve the limit as

$$W^* = (1-\alpha)(I - \alpha P)^{-1} W^{(0)}. \tag{9}$$

Then, we build a new reliable affinity matrix by symmetrizing the converged affinity matrix W^* and removing unreliable affinities, that is

$$W = \left\lfloor \frac{W^* + W^{*\top}}{2} \right\rfloor_{||\geq\theta}, \tag{10}$$

in which the operator $\lfloor S \rfloor_{||\geq\theta}$ zeros out the entries of S whose absolute values are smaller than $0 < \theta < 1$.

In summary, we are capable of learning a better affinity matrix provided with strong affinities, i.e., real constraints. The effect of learning affinity matrices subject to the known pairwise constraints is enhancing the generalization and robustness capabilities of the affinity-preserving function g formulated in eq. (5). The learned affinities between all data pairs lead to better generalization characteristics of g than using only strong affinities or only weak affinities. Compared with our parallel work [18] which employs similarity (positive affinity) propagation to learn non-sparse metrics, the proposed affinity propagation in this paper discloses more hidden relevances and irrelevances among data.

3.3 Log-Determinant Divergence

So far, we can give a general formula for semi-supervised metric learning as the minimization of the log-determinant Bregman divergence $D_{\ell d}(M, M_0) = tr(MM_0^{-1}) - \log\det(MM_0^{-1}) - m$ [8] between a given metric matrix $M_0 \in \mathbb{R}^{m\times m}$ and the desirable metric matrix M regularized by the affinity-preserving function $g(M, \mathcal{X}, \mathcal{S}, \mathcal{D}) = tr(XLX^\top M)$ prescribed in subsection 3.1, that is

$$\begin{aligned} \min_{M} \quad & tr(M_0^{-1}M) - \log\det M + \beta tr(XLX^\top M) \\ s.t. \quad & M \succeq 0 \end{aligned} \tag{11}$$

where $\beta > 0$ is the regularization parameter and the graph Laplacian L is computed based on the learned affinity matrix in eq. (10). This optimization problem is convex, but it is not easy to solve efficiently. In the next section, we propose a fast algorithm to solve it with ℓ_1 regularization.

4. SPARSE METRIC OPTIMIZATION

We adopt the following notations throughout this section. We use S^m_+ to denote the set of positive semidefinite matrices in dimensions $m \times m$. For matrix M, $\|M\|_0$ represents the number of components which are nonzeros in M; $\|M\|_1$ is the sum of absolute values of all components in M; and the matrix inner product operator is $\langle M, A\rangle = tr(M^\top A)$.

4.1 Sparse Inverse Covariance Estimation

Let us start with the hot statistical problem *Sparse Inverse Covariance Estimation* (SICE) [1][9]. To estimate the sparse inverse covariance matrix, one common approach is to penalize its maximum likelihood function by a cardinality term. Thus the maximum likelihood estimate of the sparse inverse covariance matrix reduces to the following optimization problem:

$$\max_{M\in S^m_+} \quad \log\det M - \langle \Sigma, M\rangle - \rho\|M\|_0, \tag{12}$$

where Σ is the empirical covariance matrix, $\|M\|_0$ is a penalty term to enforce the solution to be sparse, and $\rho > 0$ is the trade-off parameter to balance the maximum likelihood and the sparsity. This problem is combinatorial in essence and

thus numerically intractable. A common approach to overcome this difficulty is to replace the cardinality norm $\|M\|_0$ by its tightest convex relaxation, ℓ_1 norm $\|M\|_1$ (see [13]), which results in the following convex optimization problem:

$$\max_{M \in S_+^m} \quad \log\det M - \langle \Sigma, M\rangle - \rho\|M\|_1,$$

or equivalently, the following minimization problem:

$$\min_{M \in S_+^m} \quad -\log\det M + \langle \Sigma, M\rangle + \rho\|M\|_1. \tag{13}$$

If we regarded $\Sigma = M_0^{-1} + \beta X L X^\top$, the semi-supervised metric learning problem formulated in eq. (11) could be converted to the SICE problem by introducing the ℓ_1 norm. Therefore, we call the process of optimizing eq. (13) as *semi-supervised sparse metric learning* (S^3ML).

Note that eq. (13) can be rewritten as

$$\min_{M \in S_+^m} \max_{\|U\|_\infty \le \rho} -\log\det M + \langle \Sigma + U, M\rangle. \tag{14}$$

Thus the dual problem of eq. (13) is given by exchanging the order of max and min in eq. (14), i.e.,

$$\max_{\|U\|_\infty \le \rho} \min_{M \in S_+^m} -\log\det M + \langle \Sigma + U, M\rangle,$$

which is equivalent to

$$\begin{aligned} \max \quad & \log\det Z + m \\ s.t. \quad & \|Z - \Sigma\|_\infty \le \rho. \end{aligned} \tag{15}$$

We will see in the next subsection that our algorithm easily constructs a feasible solution to the primal problem eq. (13) and a feasible solution to the dual problem eq. (15). Thus we can easily compute the duality gap and use it to determine when to terminate the algorithm and claim optimality.

4.2 Alternating Linearization Method

If we define

$$f(M) := -\log\det M + \langle \Sigma, M\rangle \tag{16}$$

and $h(M) := \rho\|M\|_1$, then eq. (13) can be viewed as minimizing the sum of two convex functions f and h:

$$\min \quad f(M) + h(M). \tag{17}$$

We can leverage the *alternating linearization method* (ALM) proposed in [12] to solve eq. (17). ALM requires the two functions f and h to be both in the class $C^{1,1}$ ($C^{1,1}$ contains differentiable functions whose gradients are Lipschitz continuous), which means their gradients are Lipschitz continuous. So we need to smooth the ℓ_1 term $h(M)$ first. One way to smooth the ℓ_1 function $h(M)$ is to apply *Nesterov*'s smoothing technique [20]. We use $h_\sigma(M)$ to denote a smoothed approximation to $h(M)$ with a smoothness parameter σ. According to the Nesterov's technique, $h_\sigma(M)$ is given by

$$h_\sigma(M) := \max_U \{\langle U, M\rangle - \frac{\sigma}{2}\|U\|_{\mathrm{F}}^2 : \|U\|_\infty \le \rho\}, \tag{18}$$

in which $\|.\|_{\mathrm{F}}$ denotes the Frobenius norm. It is easy to check that $U^* := \min\{\rho, \max\{M/\sigma, -\rho\}\}$ is the optimal solution to eq. (18). According to Theorem 1 in [20], the gradient of $h_\sigma(M)$ is given by $\nabla h_\sigma(M) := U^*$ and is Lipschitz continuous with constant $L(h_\sigma) = 1/\sigma$. ALM which can solve the smoothed SICE problem

$$\min \quad f(M) + h_\sigma(M) \tag{19}$$

Algorithm 1 ALM for SICE

Input: $M^0 = Y^0$, $i = 0$.
repeat
  solve $M^{i+1} := \arg\min_M f(M) + h_\sigma(Y^i) + \langle \nabla h_\sigma(Y^i), M - Y^i\rangle + \frac{1}{2\mu}\|M - Y^i\|_{\mathrm{F}}^2$;
  solve $Y^{i+1} := \arg\min_Y f(M^{i+1}) + \langle \nabla f(M^{i+1}), Y - M^{i+1}\rangle + \frac{1}{2\mu}\|Y - M^{i+1}\|_{\mathrm{F}}^2 + h_\sigma(Y)$;
  $i = i + 1$;
until M^i converges
Output: $M = M^i$.

is described in Algorithm 1 where $\mu := 1/\max\{L(f), L(h_\sigma)\}$. It is easy to see that in the i-th iteration of Algorithm 1, $\nabla h_\sigma(M^i) + \Sigma$ is a feasible solution to the dual problem eq. (15), so we can conveniently compute the duality gap between this dual feasible solution and the primal feasible solution M^i. We terminate Algorithm 1 once the duality gap achieves the desired accuracy.

REMARK 2. *There are four advantages of our ALM algorithm over the block coordinate descent algorithm in [1] and [9]. First, we consider the primal problem eq. (17) where the ℓ_1 term is involved, so ALM will preserve the sparsity of the desired metric matrix M. However, the block coordinate descent algorithm in [1][9] was proposed for the dual problem eq. (15) where the desired matrix M is obtained by inverting the optimal dual solution. Such a matrix inversion usually results in a dense matrix due to the floating-point errors encountered in inverting a matrix. Second, the matrix M in Algorithm 1 is obtained by solving the subproblem where the $\log\det(M)$ term is involved, so M is always positive definite throughout Algorithm 1. Third, both subproblems in Algorithm 1 have closed-form solutions and thereby can be solved substantially efficiently. Finally, there are no iteration complexity results about the block coordinate descent algorithm in [1][9]. In contrast, we can obtain an iteration complexity bound for ALM as we will show below.*

When we apply ALM to the smoothed SICE problem eq. (19), we do get an iteration complexity bound. However, the complexity results in [12] do not apply to Algorithm 1 right away. To apply the complexity results in [12], we need to modify Algorithm 1 a little bit. First, although the function h_σ is in the class $C^{1,1}$, the function f is not in the class $C^{1,1}$ since $\log\det M$ is not in $C^{1,1}$ on S_+^m. However, restricted on the convex set $\{M : M \succeq \hat{\lambda} I\}$ with $\hat{\lambda} > 0$, $f(M)$ is in $C^{1,1}$, i.e., its gradient $\nabla f(M) = -M^{-1} + \Sigma$ is Lipschitz continuous with Lipschitz constant $L(f) = \hat{\lambda}^{-2}$. According to Proposition 3.1 in [19], the optimal solution M^* to eq. (13) satisfies $M^* \succeq \lambda I$ where $\lambda = 1/(\|\Sigma\| + m\rho)$ and $\|\Sigma\|$ denotes the largest eigenvalue of matrix Σ. We denote the domain in which $f(M)$ is in $C^{1,1}$ by $\mathcal{C}$, i.e., $\mathcal{C} := \{M \in S^m : M \succeq \lambda I\}$. Thus we can impose constraint $M \in \mathcal{C}$ to the subproblem with respect to M in Algorithm 1 to guarantee that f is in the class $C^{1,1}$. Second, Y^i might be not positive definite and $f(Y^i)$ is thus not well-defined. To this end, we need to also impose the constraint $Y \in \mathcal{C}$ to the subproblem with respect to Y in Algorithm 1 to guarantee that $f(Y^i)$ is well-defined.

Before addressing our main complexity result, we need to define the terminology *ϵ-optimal solution* first.

DEFINITION 3. *Suppose x^* is an optimal solution to the problem*

$$\min\{f(x) : x \in C\}, \tag{20}$$

where $C \subset \mathbb{R}^n$ is a convex set. $x \in C$ is an ϵ-optimal solution to eq. (20) if $f(x) - f(x^*) \leq \epsilon$ holds.

We have the following result about the relation between an approximate solution to eq. (19) and an approximate solution to eq. (17).

THEOREM 4. *For any $\epsilon > 0$, we let $\sigma := \frac{\epsilon}{m^2\rho^2}$. Suppose M_σ is an $\epsilon/2$-optimal solution to eq. (19), then M_σ is an ϵ-optimal solution to eq. (17).*

PROOF. It is easy to verify that (see equation (2.7) in [20]):

$$h_\sigma(M) \leq h(M) \leq h_\sigma(M) + \sigma D_h, \quad \forall M \in \mathbb{R}^{m\times m}, \tag{21}$$

where $D_h := \max\{\frac{1}{2}\|M\| : \|M\|_\infty \leq \rho\} = \frac{1}{2}m^2\rho^2$. Suppose M^* is an optimal solution to eq. (17) and M_σ^* is an optimal solution to eq. (19), then we exploit the inequalities in eq. (21) to derive

$$\begin{aligned}
&f(M_\sigma) + h(M_\sigma) - f(M^*) - h(M^*)\\
\leq &f(M_\sigma) + h_\sigma(M_\sigma) + \sigma D_h - f(M^*) - h_\sigma(M^*)\\
\leq &f(M_\sigma) + h_\sigma(M_\sigma) + \sigma D_h - f(M_\sigma^*) - h_\sigma(M_\sigma^*)\\
\leq &\frac{\epsilon}{2} + \sigma D_h\\
= &\epsilon,
\end{aligned}$$

where the third inequality is due to the fact that M_σ is an $\epsilon/2$-optimal solution to eq. (19) and the last equality is due to $\sigma = \frac{\epsilon}{m^2\rho^2}$. This completes the proof. □

Now we are ready to give the iteration complexity bound for ALM.

THEOREM 5. *By imposing constraints $M \in \mathcal{C}$ and $Y \in \mathcal{C}$ in the two subproblems in Algorithm 1, Algorithm 1 returns an ϵ-optimal solution to eq. (17) in $O(1/\epsilon^2)$ iterations.*

PROOF. From Theorem 4.2 and discussions in section 5.3 in [12], after imposing constraints $M \in \mathcal{C}$ and $Y \in \mathcal{C}$ in the two subproblems in Algorithm 1, Algorithm 1 returns an $\epsilon/2$-optimal solution to eq. (19) in $O(\frac{1}{\sigma\epsilon})$ iterations. If we choose $\sigma = \frac{\epsilon}{m^2\rho^2}$, then incorporating Theorem 4 we conclude that Algorithm 1 returns an ϵ-optimal solution to eq. (17) in $O(\frac{1}{\epsilon^2})$ iterations. □

Now we show how to optimize the two subproblems in Algorithm 1. The first-order optimality condition for the M-subproblem in Algorithm 1 is

$$\nabla f(M) + \nabla h_\sigma(Y^i) + (M - Y^i)/\mu = 0. \tag{22}$$

Since $\nabla f(M) = -M^{-1} + \Sigma$, it is easy to verify that $M^{i+1} := V\mathrm{diag}(\gamma)V^\top$ satisfies eq. (22) and is thus optimal to the M-subproblem in Algorithm 1, where $V\mathrm{diag}(d)V^\top$ is the eigenvalue decomposition of $Y^i - \mu(\Sigma + \nabla h_\sigma(Y^i))$ and

$$\gamma_j = (d_j + \sqrt{d_j^2 + 4\mu})/2, \quad j = 1, \ldots, m. \tag{23}$$

Table 1: Dataset Information: the numbers of features, samples and classes.

DATASET	# FEATURES	# SAMPLES	# CLASSES
IRIS	4	150	3
WINE	13	178	3
BCANCER	30	569	2
CAR	6	1728	4
MSRA-MM	225	10,000	50
NUS-WIDE	225	269,648	81

Note that γ_j is always strictly positive since $\mu > 0$ and matrix M^i is hence always positive definite. If the constraint $M \in \mathcal{C}$ is imposed in the M-subproblem in Algorithm 1, the only change in the optimal solution is changing eq. (23) to

$$\gamma_j = \max\{\lambda, (d_j + \sqrt{d_j^2 + 4\mu})/2\}, \quad j = 1, \ldots, m.$$

The first-order optimality condition for the Y-subproblem is

$$\nabla f(M^{i+1}) + (Y - M^{i+1})/\mu + \nabla h_\sigma(Y) = 0. \tag{24}$$

Since $\nabla h_\sigma(Y) = \min\{\rho, \max\{Y/\sigma, -\rho\}\}$, it is easy to verify that

$$\begin{aligned}
Y^{i+1} = &M^{i+1} - \mu\nabla f(M^{i+1}) -\\
&\mu\min\{\rho, \max\{\frac{M^{i+1} - \mu\nabla f(M^{i+1})}{\sigma + \mu}, -\rho\}\}
\end{aligned} \tag{25}$$

satisfies eq. (24) and is thus optimal to the Y-subproblem in Algorithm 1. In summary, the M-subproblem is corresponding to an eigenvalue decomposition and the Y-subproblem is a trivial projection operation. Consequently, solving M-subproblems dominates the computational complexity since solving Y-subproblems is much cheaper compared with the computational effort on solving M-subproblems.

4.3 The S^{3}ML Algorithm

Given a set of n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subseteq \mathbb{R}^m$ with a similar constraint set $\mathcal{S}$ and a dissimilar constraint set $\mathcal{D}$, an integer k, four real-valued parameters $0 < \alpha < 1$, $\theta > 0$, $\beta > 0$ and $\rho > 0$, and input metric matrix M_0 (identity matrix or inverse covariance matrix), we summarize the proposed semi-supervised sparse metric learning (S^{3}ML) algorithm in Algorithm 2. Interestingly, the proposed S^{3}ML algorithm can also work under supervised settings when simply setting $W = W^0$, so S^{3}ML is appropriate for various metric learning problems.

5. EXPERIMENTS

In this section, we compare the proposed semi-supervised sparse metric learning (S^{3}ML) algorithm with several existing state-of-the-art metric learning algorithms on six datasets including four benchmark UCI datasets and two real-world image datasets. Table 1 describes fundamental information about these datasets.

5.1 Compared Methods

The compared methods include:

Euclidean: the baseline denoted as "EU" in short.

Mahalanobis: a standard Mahalanobis metric denoted as "Mah" in short. Specifically, the metric matrix $A = \mathrm{Cov}^{-1}$ where Cov is the sample covariance matrix.

Algorithm 2 S^3ML

Input: Three sets $\mathcal{X}$, $\mathcal{S}$, $\mathcal{D}$, an integer k, four real-valued parameters $0 < \alpha < 1$, $\theta > 0$, $\beta > 0$ and $\rho > 0$, and input metric matrix M_0.

1. **k-NN Search:** Construct a neighborhood indicator matrix $P \in \mathbb{R}^{n \times n}$ upon all n samples in $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$

$$P_{ij} = \begin{cases} \frac{1}{k}, & \text{if } j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases}$$

in which $\mathcal{N}_i$ denotes the set consisting of the indexes of k nearest neighbors of $\mathbf{x}_i$.

2. **Affinity Propagation:** Set an initial affinity matrix $W^0 \in \mathbb{R}^{n \times n}$ by $W^0_{ii} = 1$ for $\forall i \in \{1, \cdots, n\}$, $W^0_{ij} = 1$ for $\forall (i,j) \in \mathcal{S}$, $W^0_{ij} = -1$ for $\forall (i,j) \in \mathcal{D}$, and $W^0_{ij} = 0$ otherwise. Calculate $W^* = (1-\alpha)(I - \alpha P)^{-1} W^{(0)}$ and the final affinity matrix is

$$W_{ij} = \begin{cases} \frac{W^*_{ij} + W^*_{ji}}{2}, & \text{if } \frac{|W^*_{ij} + W^*_{ji}|}{2} \geq \theta \\ 0, & \text{otherwise.} \end{cases}$$

Calculate the matrix $T = XLX^\top$ where $L = D - W$ and $D = \text{diag}(W\mathbf{1})$.

3. **Alternating Linearization Method:** Set $\Sigma = M_0^{-1} + \beta T$. Solve the following optimization problem using Algorithm 1: $\min_{M \in S_+^m} -\log\det M + \langle \Sigma, M \rangle + \rho \|M\|_1$.

Output: The sparse metric matrix M.

LMNN [28]: Large Margin Nearest Neighbor which works under supervised settings where each sample has an exact class label.

ITML [8]: Information-Theoretic Metric Learning which works under pairwise relevance constraints but does not explicitly engage the unlabeled data.

SDML [21]: Sparse Distance Metric Learning which works under pairwise relevance constraints to produce sparse metrics but does not explicitly engage the unlabeled data.

S^2ML: the supervised counterpart of S^3ML with $W = W^0$, which works under pairwise relevance constraints to produce sparse metrics and does not engage the unlabeled data.

LRML [14]: Laplacian Regularized Metric Learning which works under pairwise relevance constraints and explicitly engages the unlabeled data.

S^3ML: our semi-supervised sparse metric learning method which works under pairwise relevance constraints to produce sparse metrics and explicitly engages the unlabeled data.

To sum up, the compared distance metrics include two standard unsupervised metrics, four (weakly) supervised metrics LMNN, ITML, SDML and S^2ML, as well as two semi-supervised metrics LRML and S^3ML. For the implementation of affinity propagation concerned in S^3ML, we obtain the full affinity matrix in eq. (10) by setting $k = 6$, $\alpha = 0.5$, and $\theta = 0.01$ for all datasets. To run ALM for both of S^2ML and S^3ML, we fix the smoothing parameter $\sigma = 10^{-6}$ for all datasets. We tune two regularization parameters β (for affinity-preserving) and ρ (for sparsity) to the best values on each dataset.

Table 2: Comparisons of classification error rates (%) on UCI datasets.

Compared Methods	Iris	Wine	Breast Cancer	Car
EU	8.45	25.34	15.69	19.80
Mah	10.67	30.25	14.32	22.25
LMNN	7.06	12.67	10.82	15.86
ITML	6.34	22.19	12.61	13.25
SDML	4.55	10.78	7.14	10.74
S^2ML	4.27	9.62	7.56	9.65
LRML	3.26	7.71	6.26	8.46
S^3ML	**2.10**	**7.20**	**2.52**	**6.13**

5.2 Benchmark UCI Datasets

We apply eight distance metrics listed above on four UCI datasets: Iris, Wine, Breast Cancer and Car, where we randomly choose 5% examples from each class as labeled data and treat the other examples as unlabeled data. We evaluate kNN (k=1) classification performance in terms of error rates on unlabeled data. For each dataset, we repeat the evaluation process with the 8 algorithms 50 times, and take the average error rates for comparison. Table 2 reports the average error rates for all compared methods. In this group of experiments, we let the inverse covariance matrix be the initial metric matrix that is fed to ITML, SDML, S^2ML and S^3ML.

In contrast to EU, Mah, LMNN, ITML, SDML, S^2ML and LRML, the proposed S^3ML achieves the lowest average error rates across all these datasets. The significant improvement of S^3ML over S^2ML demonstrates that the "semi-supervised" scenario makes sense and that the proposed affinity propagation trick used in S^3ML effectively utilizes the information of unlabeled data. While the computational efficiency of S^2ML is comparable to the recently proposed sparse metric learning method SDML that applies the block coordinate descent algorithm to solve the related SICE, S^2ML achieves better average classification performance on three datasets, and more importantly, SDML cannot guarantee to produce truly sparse metrics in bounded iterations.

5.3 Image Datasets

We conduct the experiments for image retrieval tasks on two image datasets. One is the MSRA-MM dataset [27] which consists of 10,000 images and labeled with 50 concepts. The other is the NUS-WIDE dataset [6] which consists of 269,648 images with 81 concepts labeled.

In this group of experiments, we set the identity matrix to be the initial metric matrix used for ITML, SDML and S^3ML. Since the supervised metric learning approach LMNN requires explicit class labels, it is unsuitable for image retrieval. Therefore, we only compare six methods (excluding LMNN and S^2ML). We construct a subset for each image concept by selecting 2500 images in which 500 are labeled as relevant and 2000 as irrelevant from the entire datasets. Then, we randomly select 20% samples to form the similar and dissimilar pair sets. We perform 20 random trials and show the averaged performance. The visual features we used in this experiment is the 225-dimentional block-wise color moments extracted over $5 * 5$ fixed grid partitions with each block described by 9-dimensional feature. The performance

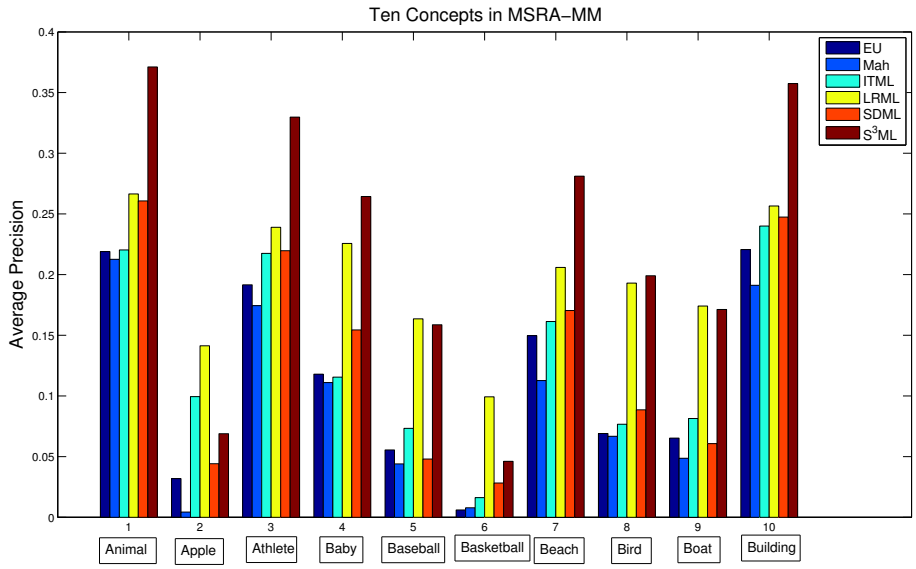

Figure 1: The MSRA-MM dataset. Average precisions for 10 concepts (classes) with the initial 20% labeling rate.

Table 3: Comparisons of Mean APs (MAPs) (%) on image datasets.

Compared Methods	MSRA-MM	NUS-WIDE
EU	14.59	26.57
Mah	12.89	21.63
ITML	16.17	26.98
SDML	16.21	28.82
LRML	22.45	30.19
S^{3}ML	**27.30**	**39.87**

is measured by the widely used non-interpolated Average Precision (AP) which averages the precision values obtained when each relevant image occurs. We average the APs over all concepts in each dataset to get the Mean AP (MAP) for overall performance measurement.

Table 3 lists MAPs of six methods in comparison. We also show APs for ten concepts in each dataset in Fig. 1 and Fig. 2, respectively. We can observe that S^{3}ML consistently achieves the highest accuracy for the most of the concepts. Both MAPs and APs results show that the proposed S^{3}ML is very promising for handling image retrieval tasks.

6. CONCLUSIONS

This paper proposes a semi-supervised sparse metric learning (S^{3}ML) algorithm which works under a few of the pairwise similar and dissimilar constraints and produces favorable sparse metrics. In contrast to previous metric learning techniques, the proposed S^{3}ML can employ unlabeled data in a principled way, i.e., affinity propagation, that assigns reliable affinities to all data pairs through propagating prior strong affinities. Observing that the existing sparse metric learning methods did not optimize the sparse metrics explicitly, we apply the alternating linearization method to optimize the sparse metrics directly. This alternating linearization method has appealing computational and theoretical properties. Extensive experiments have been conducted to evaluate classification and retrieval performance using the sparse metrics offered by S^{3}ML. The promising results show that S^{3}ML is superior to the state-of-the-arts.

Suppose data do not have a vector form and are merely measured via some kernel function, then nonlinear metrics are needed. Our latest work [16] actually learned a nonlinear metric for semi-supervised classification. Another concern is that if the data dimension is substantially large any metric learning method will be computationally expensive and even prohibitive. Motivated by our earlier work [17], it is possible to embed subspace learning into sparse metric learning so that S^{3}ML is able to work on very high-dimensional data such as microarray data and text data.

7. ACKNOWLEDGMENTS

This work was supported by NTU NAP Grant with project number M58020010 and the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1006), Zhejiang University. This work was also supported by grants from Natural Science Foundation of China (No. 60975029) and Shenzhen Bureau of Science Technology&Information, China (No. JC200903180635A).

References

[1] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

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

[3] S. Basu, M. Bilenko, and R. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. KDD*, 2004.

[4] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. ICML*, 2004.

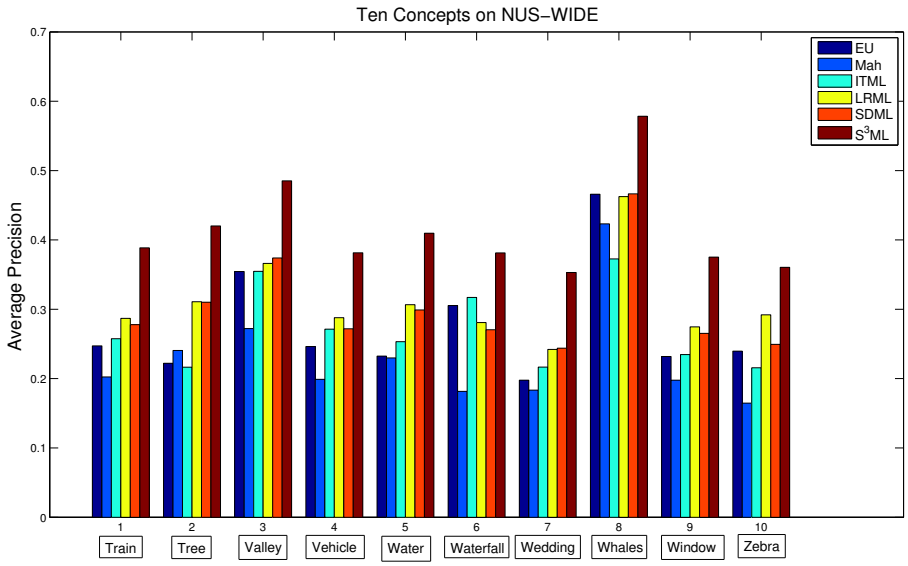

Figure 2: The NUS-WIDE dataset. Average precisions for 10 concepts (classes) with the initial 20% labeling rate.

[5] S. Boyd and L. Vandenberge. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2003.

[6] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proc. CIVR*, 2009.

[7] J. V. Davis and I. S. Dhillon. Structured metric learning for high dimensional problems. In *Proc. KDD*, 2008.

[8] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proc. ICML*, 2007.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 8(1):1–10, 2007.

[10] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS 18*, 2006.

[11] J. Goldberger, S. Roweis, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS 17*, 2005.

[12] D. Goldfarb and S. Ma. Fast alternating linearization methods for minimizing the sum of two convex functions. Technical report, Department of IEOR, Columbia University, 2009. Preprint available at http://arxiv.org/abs/0912.4571.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Springer-Verlag, New York, 1993.

[14] S. C. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Proc. CVPR*, 2008.

[15] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A kernel approach. *Machine Learning*, 74(1):1–22, 2009.

[16] W. Liu, B. Qian, J. Cui, and J. Liu. Spectral kernel learning for semi-supervised classification. In *Proc. IJCAI*, 2009.

[17] W. Liu, D. Tao, and J. Liu. Transductive component analysis. In *Proc. IEEE ICDM*, 2008.

[18] W. Liu, X. Tian, D. Tao, and J. Liu. Constrained metric learning via distance gap maximization. In *Proc. AAAI*, 2010.

[19] Z. Lu. Smooth optimization approach for sparse covariance selection. *SIAM J. Optim.*, 19(4):1807–1827, 2009.

[20] Y. E. Nesterov. Smooth minimization for non-smooth functions. *Math. Program. Ser. A*, 103:127–152, 2005.

[21] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang. An efficient sparse metric learning in high-dimensional space via ℓ_1-penalized log-determinant regularization. In *Proc. ICML*, 2009.

[22] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proc. KDD*, 2006.

[23] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A powerful tool in interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[24] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI*, 22(12):1349–1380, 2000.

[25] W. Tang, H. Xiong, S. Zhong, and J. Wu. Enhancing semi-supervised clustering: A feature projection perspective. In *Proc. KDD*, 2007.

[26] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. ICML*, 2001.

[27] M. Wang, L. Yang, and X.-S. Hua. Msra-mm: Bridging research and industrial societies for multimedia information retrieval. Technical report, Miscosoft Research Asia, 2009. No. MSR-TR-2009-30.

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

[29] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS 15*, 2003.

[30] Y. Ying, K. Huang, and C. Campbell. Sparse metric learning via smooth optimization. In *NIPS 22*, 2010.

[31] X. Zhu. Semi-supervied learning literature survey. Technical report, University of Wisconsin Madison, 2008.