Semi-Supervised Distance Metric Learning by Quadratic Programming

Hakan Cevikalp
Eskisehir Osmangazi University, Meselik 26480 Eskisehir Turkey.
hakan.cevikalp@gmail.com

Abstract

This paper introduces a semi-supervised distance metric learning algorithm which uses pair-wise equivalence (similarity and dissimilarity) constraints to improve the original distance metric in lower-dimensional input spaces. We restrict ourselves to pseudo-metrics that are in quadratic forms parameterized by positive semi-definite matrices. The proposed method works in both the input space and kernel induced feature space, and learning distance metric is formulated as a quadratic optimization problem which returns a global optimal solution. Experimental results on several databases show that the learned distance metric improves the performances of the subsequent classification and clustering algorithms.

1. Introduction

Learning distance metrics is very important for various applications such as classification, image and video retrieval, and image segmentation [1-4], and this task is much easier when class labels associated to the data samples are available. However, in many applications, there is a lack of labeled data since obtaining labels is a costly procedure as it often requires human labor. On the other hand, in some applications, side information – given in the form of pair-wise equivalence (similarity and dissimilarity) constraints between points – is available without or with less extra cost. For example, faces extracted from successive video frames in roughly the same location can be assumed to represent same person, whereas faces extracted in different locations in the same frame cannot be the same person. In some applications, side information is the natural form of supervision, e.g., in image retrieval, there is only the notion of similarities between the query and retrieved images. Side information may also come from human feedback in interactive environments, often at a substantially lower cost than explicit labeled data as in semi-supervised image segmentation applications [2].

Recently, learning distance metrics has been actively studied in machine learning. Some of the distance metric learning algorithms use class labels [5,6] and we will not consider them here. We will focus only semi-supervised distance metric learning algorithms which use equivalence constraints. Existing semi-supervised distance metric learning methods [4,7-13] revise the original distance metric (commonly chosen as the Euclidean distance) to accommodate the pair-wise equivalence constraints, and then a clustering algorithm with the learned distance metric is used to partition data to discover the desired groups within data. In [9], a full-rank pseudo distance metric is learned by means of convex programming using equivalence constraints. Relevant Component Analysis [11] is introduced as an alternative to this method, but it can exploit only similarity constraints. Shalev-Shwartz et al. [12] proposed a sophisticated online distance metric learning algorithm that uses side information. Davis et al. [10] proposed an information-theoretic approach to learn a Mahalanobis distance function. Kwok and Tsang [7,8] formulated a metric learning problem that uses side information in a quadratic optimization scheme. Note that all semi-supervised distance metric learning algorithms mentioned above attempt to learn full-rank distance metrics, and thus they are suitable for lower-dimensional input spaces. In high-dimensional spaces it is better to learn low-rank distance metrics as in [2]. A comprehensive survey of semi-supervised distance metric learning techniques can be found in [13].

In this paper we also focus on lower-dimensional spaces and try to learn a pseudo distance metric parameterized by positive semi-definite matrices. To this end, we formulate the problem as a quadratic optimization problem as in [7,8], but we incorporate the large margin concept in the procedure and reduce the number of user-chosen parameters.

2. Method

2.1 Problem Setting

Let $\mathbf{x}_i \in \Re^d$, $i = 1,...,n$, denote the samples in the training set. We are given a set of equivalence con-

straints in the form of similar and dissimilar pairs. Let *S* be the set of similar pairs

$$S = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\} \quad (1)$$

and let *D* be the set of dissimilar pairs

$$D = \{(\mathbf{x}_k, \mathbf{x}_l) \mid \mathbf{x}_k \text{ and } \mathbf{x}_l \text{ belong to different classes}\} \quad (2)$$

Our objective is to find a pseudo-metric that satisfies the equivalence constraints and at the same time reflects the true underlying relationships imposed by such constraints. We focus on the pseudo-metrics of the form

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \|_{\mathbf{A}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)}, \quad (3)$$

where $\mathbf{A} \geq \mathbf{0}$ is a symmetric positive semi-definite matrix. In this case there exists a rectangular projection matrix $\mathbf{W}$ of size $d \times q$ ($q \leq d$) satisfying $\mathbf{A} = \mathbf{W}\mathbf{W}^T$ such that

$$\| \mathbf{x}_i - \mathbf{x}_j \|_{\mathbf{A}}^2 = \| \mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j \|_2^2. \quad (4)$$

2.2 Learning Distance Metric by Quadratic Programming

Assume that the learned distance matrix is $\mathbf{A}$. Intuitively, the learned distance metric must pull similar pairs closer and push the dissimilar pairs apart. Additionally, it should generalize well to unseen data. To this end, we define the margin *b*, which is defined to be the minimum separation between all pairs of similar and dissimilar samples. That is

$$d_{\mathbf{A}}^2(\mathbf{x}_k, \mathbf{x}_l) - d_{\mathbf{A}}^2(\mathbf{x}_i, \mathbf{x}_j) \geq b, \ (\mathbf{x}_k, \mathbf{x}_l) \in D \text{ and } (\mathbf{x}_i, \mathbf{x}_j) \in S$$

Without loss of generality, we can scale **A** and *b* by any positive constant. We therefore set *b* to be 2 and search for a distance matrix **A** which has small norm. However, if we have *m* similar and *n* dissimilar pairs, the number of total constraints will be *mn* which may be a large number. Therefore we introduce a threshold $\gamma' \geq 1$ and replace the constraints with

$$\begin{aligned} & d_{\mathbf{A}}{}^2(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma' - 1, \quad (\mathbf{x}_i, \mathbf{x}_j) \in S, \\ & d_{\mathbf{A}}{}^2(\mathbf{x}_k, \mathbf{x}_l) \geq \gamma' + 1, \quad (\mathbf{x}_k, \mathbf{x}_l) \in D, \end{aligned} \quad (5)$$

If we let $\gamma = \gamma' - 1$ and introduce slack variables for the sample pairs violating margin constraints, we obtain the following quadratic programming problem

$$\begin{aligned} \min \ & \frac{1}{2} \| \mathbf{A} \|_2^2 + \frac{C_S}{n_S} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \xi_{ij} + \frac{C_D}{n_D} \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \xi_{kl} \\ \text{s.t.} \ & d_{\mathbf{A}}{}^2(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma + \xi_{ij}, \quad (\mathbf{x}_i, \mathbf{x}_j) \in S, \\ & d_{\mathbf{A}}{}^2(\mathbf{x}_k, \mathbf{x}_l) \geq \gamma + 2 - \xi_{kl}, \quad (\mathbf{x}_k, \mathbf{x}_l) \in D, \\ & \gamma, \xi_{ij}, \xi_{kl} \geq 0 \end{aligned} \quad (6)$$

where n_S and n_D are the numbers of pairs in *S* and *D* respectively, C_S, C_D are non-negative user-chosen adjustable parameters, and ξ_{ij}, ξ_{kl} are slack variables. Note that the similar sample pairs which are far from each other contribute more to the loss function than the ones which are closer. In a similar manner, the dissimilar pairs which are closer to each other contribute more to the loss function than the ones which are further from each other. In fact if the square of distances between the dissimilar pairs are larger than the threshold ($\gamma' + 1$), those dissimilar sample pairs do not contribute to the loss function at all. Therefore, just as in the Support Vector Machine's hinge loss, our objective function is triggered by the dissimilar pairs in the vicinity of decision boundaries that participate in the construction of the inter-class decision boundaries. In contrast, there is not such a systematical selection mechanism that respects the margin concept in the method of Kwok and Tsang [7,8]. They just aim to pull all similar pairs together and to maximize the distance differences between the learned and original distance metrics for dissimilar sample pairs.

To derive the dual, we consider the Lagrangian

$$\begin{aligned} L(\mathbf{A}, \xi, \gamma, \alpha, \eta, \mu) \ = \ & \frac{1}{2} \| \mathbf{A} \|_2^2 + \frac{C_S}{n_S} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \xi_{ij} \\ & + \frac{C_D}{n_D} \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \xi_{kl} + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \alpha_{ij} \left\{ (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j) - \gamma - \xi_{ij} \right\} \\ & + \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \alpha_{kl} \left\{ -(\mathbf{x}_k - \mathbf{x}_l)^T \mathbf{A} (\mathbf{x}_k - \mathbf{x}_l) + \gamma + 2 - \xi_{kl} \right\} \\ & - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \eta_{ij} \xi_{ij} - \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \eta_{kl} \xi_{kl} - \mu\gamma \end{aligned}$$

where $\alpha_{ij}, \alpha_{kl}, \eta_{ij}, \eta_{kl}, \mu \geq 0$. The Lagrangian *L* has to be maximized with respect to α, η, μ and minimized with respect to $\mathbf{A}, \xi, \gamma$. The optimality conditions yield

$$\begin{aligned} & \frac{\partial L}{\partial \mathbf{A}} = \mathbf{0} \rightarrow \mathbf{A} = \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \alpha_{kl} (\mathbf{x}_k - \mathbf{x}_l)(\mathbf{x}_k - \mathbf{x}_l)^T \\ & \qquad\qquad - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \alpha_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T, \\ & \frac{\partial L}{\partial \xi_{ij}} = 0 \rightarrow \alpha_{ij} = \frac{C_S}{n_S} - \eta_{ij} \rightarrow 0 \leq \alpha_{ij} \leq \frac{C_S}{n_S}, \quad (\mathbf{x}_i, \mathbf{x}_j) \in S, \\ & \frac{\partial L}{\partial \xi_{kl}} = 0 \rightarrow \alpha_{kl} = \frac{C_D}{n_D} - \eta_{kl} \rightarrow 0 \leq \alpha_{kl} \leq \frac{C_D}{n_D}, \quad (\mathbf{x}_k, \mathbf{x}_l) \in D, \\ & \frac{\partial L}{\partial \gamma} = 0 \rightarrow \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \alpha_{kl} - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \alpha_{ij} = \mu \\ & \qquad \rightarrow \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \alpha_{kl} - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \alpha_{ij} \geq 0 \end{aligned}$$

Thus, the dual of the optimization problem becomes

$$\min \ \frac{1}{2} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \sum_{(\mathbf{x}_m, \mathbf{x}_n) \in S} \alpha_{ij} \alpha_{mn} \left[(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_m - \mathbf{x}_n) \right]^2$$

$$+\frac{1}{2}\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\sum_{(\mathbf{x}_p,\mathbf{x}_q)\in D}\alpha_{kl}\alpha_{pq}\left[(\mathbf{x}_k-\mathbf{x}_l)^T(\mathbf{x}_p-\mathbf{x}_q)\right]^2$$
$$-\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{ij}\alpha_{kl}\left[(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_k-\mathbf{x}_l)\right]^2-2\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl} \quad (7)$$

subject to

$$\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij}-\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl}\le 0,\ 0\le\alpha_{ij}\le\frac{C_S}{n_S},\text{ and }\ 0\le\alpha_{kl}\le\frac{C_D}{n_d}.$$

This is a quadratic programming problem with $n_s + n_d$ variables (which is independent of input dimensionality *d*) as in [7]. From the Karush-Kuhn-Tucker conditions, we get

$$d_{\mathbf{A}}{}^2(\mathbf{x}_i,\mathbf{x}_j)\begin{cases}=\gamma & 0<\alpha_{ij}<C_S/n_S,\\ \le\gamma & \alpha_{ij}=0,\\ \ge\gamma & \alpha_{ij}=C_S/n_S.\end{cases}\qquad(\mathbf{x}_i,\mathbf{x}_j)\in S$$

$$d_{\mathbf{A}}{}^2(\mathbf{x}_k,\mathbf{x}_l)\begin{cases}=\gamma+2 & 0<\alpha_{kl}<C_D/n_D,\\ \ge\gamma+2 & \alpha_{kl}=0,\\ \le\gamma+2 & \alpha_{kl}=C_D/n_D.\end{cases}\qquad(\mathbf{x}_k,\mathbf{x}_l)\in D$$

Thus to find the value of γ, we take all the sample pairs with $0<\alpha_{ij}<C_S/n_S$ and $0<\alpha_{kl}<C_D/n_D$, compute corresponding $d_{\mathbf{A}}{}^2(\mathbf{x}_i,\mathbf{x}_j)$ and $d_{\mathbf{A}}{}^2(\mathbf{x}_k,\mathbf{x}_l)-2$ and average them.

Note that the resulting distance matrix **A** is not necessarily a positive semi-definite matrix. To make sure that **A** is a positive semi-definite matrix, we apply eigen-decomposition to **A** and construct it using positive eigenvalues and corresponding eigenvectors, $\mathbf{A}=\sum_k\lambda_k\mathbf{u}_k\mathbf{u}_k^T$ where λ_k's are the positive eigenvalues and $\mathbf{u}_k$'s are the corresponding eigenvectors.

2.3 Extension to the Nonlinear Case

Here we consider the case where the data samples are mapped into a higher-dimensional feature space and the distance metric is sought in this feature space. This is accomplished by using the kernel trick. Notice that the objective function of (7) can be written in terms of the dot products of the sample pairs. Thus, we replace all $\langle\mathbf{x}_i,\mathbf{x}_j\rangle=\mathbf{x}_i^T\mathbf{x}_j$ with the kernel function $k(\mathbf{x}_i,\mathbf{x}_j)=\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ where $\phi:\Re^d\to\Im$ is the mapping function from the input space to the feature space $\Im$. Once we compute the optimal α coefficients, the distance between two samples $\phi(\mathbf{x}_a)$ and $\phi(\mathbf{x}_b)$ in the mapped space under the metric **A** can be computed as

$$\begin{aligned}d_{\mathbf{A}}(\phi(\mathbf{x}_a),\phi(\mathbf{x}_b))&=(\phi(\mathbf{x}_a)-\phi(\mathbf{x}_b))^T\mathbf{A}(\phi(\mathbf{x}_a)-\phi(\mathbf{x}_b))\\&=\phi(\mathbf{x}_a)^T\mathbf{A}\phi(\mathbf{x}_a)-2\phi(\mathbf{x}_a)^T\mathbf{A}\phi(\mathbf{x}_b)+\phi(\mathbf{x}_b)^T\mathbf{A}\phi(\mathbf{x}_b)\\&=\tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_a)-2\tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b)+\tilde{k}_{\mathbf{A}}(\mathbf{x}_b,\mathbf{x}_b)\end{aligned}$$

where

$$\tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b)=\phi(\mathbf{x}_a)^T\mathbf{A}\phi(\mathbf{x}_b)=\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl}\{[k(\mathbf{x}_a,\mathbf{x}_k)-k(\mathbf{x}_a,\mathbf{x}_l)]$$
$$[k(\mathbf{x}_k,\mathbf{x}_b)-k(\mathbf{x}_l,\mathbf{x}_b)]\}-\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij}\{[k(\mathbf{x}_a,\mathbf{x}_i)-k(\mathbf{x}_a,\mathbf{x}_j]$$
$$[k(\mathbf{x}_i,\mathbf{x}_b)-k(\mathbf{x}_j,\mathbf{x}_b]\}.$$

3. Experiments

We performed experiments on two synthetic databases and three real-world databases chosen from UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). We compared the distance metric obtained by the proposed method to the Euclidean distance metric and the distance metrics learned by the Relevant Component Analysis (RCA) [11] and the method of Tsang and Kwok [8]. We used only positive semi-definite distance matrices in the experiments. For the nonlinear case, we used the Gaussian kernels. In order to assess the performance of the distance metrics, we evaluated the clustering and classification performances. The k-means and spectral clustering are used as clustering algorithms (we report the one yielding the best result), and the pair-wise F measure (harmonic mean of the pair-wise precision and recall measures [2]) is used to evaluate the clustering results based on the underlying classes. For classification, we used 1-nearest neighbor classification rule with the learned distance metrics.

3.1 Experiments on Synthetic Databases

The first synthetic database includes 10-dimensional data samples belonging two classes. The first dimension is distinctive feature, where the first class is normally distributed as *N*(3,1) and the second class as *N*(-3,1). The remaining dimensions are irrelevant features distributed as *N*(0,16). Since the data are linearly separable, we only tested linear methods for this database. We created 100 samples for each class and used 50 samples per class for choosing equivalence constraints and the remaining samples are used for testing. We used only 100 (60 similarity and 40 dissimilarity) equivalence constraints. For the second synthetic database, we used 2-dimensional samples drawn from two-component mixture models which are typically used in XOR problem. Classes are not linearly separable, thus we tested kernel methods with the Gaussian kernel. We used only 40 (20 similarity and 20 dissimilarity) constraints. Clustering and classification accuracies are given in Table I and Table II respectively. Results are averages over 50 runs.

Table I. Classification Accuracy (%)

Data	Kernel	Euclidean Metric	RCA	Tsang & Kwok [8]	Proposed Method
1st Synth.	Linear	81.1	**98.9**	93.8	94.1
2nd Synth.	Gaussian	**99.95**	-	99.89	**99.95**

Table II. Clustering Accuracy (%)

Data	Kernel	Euclidean Metric	RCA	Tsang & Kwok [8]	Proposed Method
1st Synth.	Linear	58.2	**96.4**	87.4	88.6
2nd Synth.	Gaussian	66.22	-	99.18	**99.94**

Since the first synthetic data has identical covariance distribution for both classes, RCA performs the best as expected. Our proposed method comes the second outperforming method of Tsang and Kwok [8] with a slight edge. For the second database RCA does not work well since the data has nonlinear distribution. The best classification accuracy is obtained by both the proposed method and Euclidean metric, whereas our proposed method is the best performer in terms of clustering accuracy. Note that the clustering performance of the Euclidean metric is very low. In general, all metric learning methods show an improvement over the Euclidean metric.

3.2 Experiments on Real Databases

Here we tested our proposed method on three databases (Iris, Wine, and Wisconsin Diagnostic Breast Cancer - WDBC) chosen from UCI Repository. For all datasets, we used the half of the samples for choosing 150 pair-wise equivalence constraints, and the remaining data samples are used for testing. Clustering and classification accuracies are given in Table III and Table IV respectively. Results are averages over 20 runs.

Table III. Classification Accuracy (%)

Data	Kernel	Euclidean Metric	RCA	Tsang & Kwok [9]	Proposed Method
Iris	Linear	**95.83**	95.75	93.56	95.00
	Gaussian	95.83	-	96.16	**96.67**
Wine	Linear	93.77	95.00	94.91	**95.51**
	Gaussian	93.77	-	94.40	**96.70**
WDBC	Linear	**95.35**	90.50	94.72	95.28
	Gaussian	95.35	-	95.14	**96.27**

Table IV. Clustering Accuracy (%)

Data	Kernel	Euclidean Metric	RCA	Tsang & Kwok [9]	Proposed Method
Iris	Linear	82.22	**90.52**	86.23	88.67
	Gaussian	82.22	-	**89.74**	88.50
Wine	Linear	84.73	**85.16**	84.40	84.63
	Gaussian	84.73	-	84.30	**85.16**
WDBC	Linear	86.17	87.17	87.23	**88.84**
	Gaussian	86.17	-	87.29	**89.91**

As can be seen from the tables, our proposed method yields the best classification accuracies for all tested methods by using the Gaussian kernel. In terms of clustering accuracy, the proposed kernel method achieves the best accuracy for Wine and WDBC databases. Kernel methods usually give better results compared to their linear counterparts.

4. Conclusion

In this paper we introduced a new distance metric learning algorithm that uses equivalence constraints. The metric learning problem is formulated as a quadratic optimization problem which returns a global optimum solution. The method works both in the input and the kernel induced feature spaces, and it is easier to use compared to the method proposed in [8] since the number of free parameters is reduced. Experimental results show that the learned distance metric improves the clustering and classification performances and generally outperforms the method of Tsang and Kwok [8].

Acknowledgement

This work is supported by the Young Scientists Award Programme (TÜBA-GEBİP/2010) of the Turkish Academy of Sciences.

References

[1] B. Babenko, S. Branson, S. Belongie, "Similarity metrics for categorization: from monolithic to category specific," International Conference on Computer Vision, 2009.

[2] H. Cevikalp and R. Paredes, "Semi-supervised distance metric learning for visual object classification," International Conference on Computer Vision Theory and Applications, 2009.

[3] A. Ghodsi, D. Wilkinson, F. Southey, "Improving embeddings by flexible exploitation of side information," Inter. Joint Conference on Artificial Intelligence, 2007.

[4] T. Hertz, N. Shental, A. Bar-Hillel, D. Weinshall, "Enhancing image and video retrieval: Learning via equivalence constraints," IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 2003.

[5] A. Globerson, S. Roweis, "Metric learning by collapsing classes," Advances in Neural Information Processing Systems (NIPS), 2005.

[6] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinow, "Neighborhood component analysis," NIPS, 2004.

[7] J. Kwok and I. W. Tsang, "Learning with idealized kernels," International Conference on Machine Learning (ICML), 2003.

[8] I. W. Tsang and J. Kwok, "Distance metric learning with kernels," International Conference on Artificial Neural Networks, 2003.

[9] E. P. Xing, A. Y. Ng, M. Jordan, S. Russell, "Distance metric learning with application to clustering with side-information," NIPS, 2003.

[10] J. V. Davis, B. Kullis, P. Jain, I. S. Dhillon, "Information-theoretic metric learning," ICML, 2007.

[11] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, "Learning distance functions using equivalence relations," ICML, 2003.

[12] S. Shalew-Shwartz, Y. Singer, A. Y. Ng, "Online and batch learning of pseudo-metrics," ICML, 2004.

[13] L. Yang, R. Jin, "Distance metric learning: A comprehensive survey," http://www.cse.msu.edu/~yangliu1/framesurveyv2.pdf, 2006.