Kernel-Based Metric Adaptation with Pairwise Constraints

Hong Chang and Dit-Yan Yeung

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{hongch,dyyeung}@cs.ust.hk

Abstract. Many supervised and unsupervised learning algorithms depend on the choice of an appropriate distance metric. While metric learning for supervised learning tasks has a long history, extending it to learning tasks with weaker supervisory information has only been studied very recently. In particular, several methods have been proposed for semi-supervised metric learning based on pairwise (dis)similarity information. In this paper, we propose a kernel-based approach for nonlinear metric learning, which performs locally linear translation in the kernel-induced feature space. We formulate the metric learning problem as a kernel learning problem and solve it efficiently by kernel matrix adaptation. Experimental results based on synthetic and real-world data sets show that our approach is promising for semi-supervised metric learning.

1 Introduction

Many machine learning and pattern recognition methods, such as nearest neighbor classifiers, radial basis function networks, support vector machines for classification and the k-means algorithm for clustering involve the use of a distance metric. The performance of these methods often depends very much on the metric of choice. Instead of determining a metric manually, a promising approach is to learn an appropriate metric from data.

For supervised learning applications such as classification and regression tasks, one can easily formulate the distance function learning problem as a well-defined optimization problem based on the supervisory information available in the training data. This approach can be dated back to early work on optimizing the metric for k-nearest neighbor density estimation [1]. More recent research continued to develop locally adaptive metrics for nearest neighbor classifiers [2–5]. Besides, there are other methods that also perform metric learning based on nearest neighbors, e.g., radial basis function networks and variants [6].

While class label information is available for metric learning in classification (or supervised learning) tasks, such information is not available in standard clustering (or unsupervised learning) tasks. Under the unsupervised learning setting, the distance function learning problem is ill-posed with no well-defined optimization criteria. In order to adapt the metric to improve the clustering

results, some additional background knowledge or supervisory information is required. The supervisory information may be in the form of labeled data, which are typically limited in quantity. For such problems, the classification accuracy can usually be improved with the aid of additional unlabeled data. Some methods that adopt this approach include [7–9].

Another type of supervisory information is in the form of pairwise similarity or dissimilarity constraints. This type of supervisory information is weaker than the first type, in that pairwise constraints can be derived from labeled data but not vice versa. Wagstaff and Cardie [10] and Wagstaff et al. [11] first used such pairwise constraints to improve clustering results. Extensions have also been made to model-based clustering based on the expectation-maximization (EM) algorithm for Gaussian mixture models [12, 13]. However, these methods do not incorporate metric learning into the clustering algorithms. Recently, some methods have been proposed for learning global Mahalanobis metrics and related distance functions from pairwise information [14–17]. However, the distance functions learned are either nonmetric or globally linear metrics. Chang and Yeung [18] generalized the globally linear metrics to a new metric that is linear locally but nonlinear globally. However, the criterion function of the optimization problem defined in that paper has local optima, and the algorithm is not efficient enough.

In this paper, we propose a new kernel-based metric learning method along the same direction we pursued before [18]. Instead of applying metric adaptation in the input space, we define locally linear translation in the kernel-induced feature space, where data points have higher separability. Instead of formulate the metric learning problem as an optimization problem, we formulate it as a kernel learning problem, and solve it by iterative kernel matrix adaptation. As a nonparametric approach, the new method has higher computational efficiency than that proposed in [18].

The rest of this paper is organized as follows. In Section 2, we present our metric learning method, where we perform metric learning in kernel-induced feature space and formulate the metric learning problem as a kernel learning problem. Experimental results on both synthetic and real data are presented in Section 3, comparing our method with some previous metric learning methods. Finally, some concluding remarks are given in the last section.

2 Kernel-Based Metric Adaption

2.1 Basic Ideas

Let us denote a set of n data points in a d-dimensional input space by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the set of pairwise similarity constraints by $\mathcal{S}$, and the set of pairwise dissimilarity constraints by $\mathcal{D}$. $\mathcal{S}$ and $\mathcal{D}$ are both represented as sets of point pairs, where each pair $(\mathbf{x}_i, \mathbf{x}_j)$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar or dissimilar to each other, respectively. Intuitively, we want to adjust the locations of the data points, such that similar points in similarity constraints tend to get

closer while dissimilar points in dissimilarity constraints tend to move away from each other. (1) For computational efficiency, we resort to locally linear translation. (2) Instead of applying locally linear translation in the input space, we define the translation in the kernel-induced feature space, where data points have higher separability. (3) To preserve the topological relationships between data points, we move not only the points involved in the similarity and dissimilarity constraints but also other points in their neighborhoods. Locally linear translation is equivalent to changing the metric of the feature space implicitly. In this section, we will propose a nonparametric metric learning algorithm in kernel space and establish the relationship between metric learning with kernel matrix adaptation.

2.2 Centering in the Feature Space

Suppose we use a kernel function $\hat{k}$ which induces a nonlinear mapping $\hat{\phi}$ from $\mathcal{X}$ to some feature space $\mathcal{F}$.[1] The images of the n points in $\mathcal{F}$ are $\hat{\phi}(\mathbf{x}_i)\,(i = 1,\ldots,n)$, which in general are not centered (i.e., their sample mean is not zero). The corresponding kernel matrix $\hat{\mathbf{K}} = [\hat{k}(\mathbf{x}_i,\mathbf{x}_j)]_{n\times n} = [\langle\hat{\phi}(\mathbf{x}_i),\hat{\phi}(\mathbf{x}_j)\rangle]_{n\times n}$.

We want to transform (simply by translating) the coordinate system of $\mathcal{F}$ such that the new origin is at the sample mean of the n points. As a result, we also convert the kernel matrix $\hat{\mathbf{K}}$ to $\mathbf{K} = [k(\mathbf{x}_i,\mathbf{x}_j)]_{n\times n} = [\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle]_{n\times n}$.

Let $\mathbf{\Phi} = [\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_n)]^T$, $\hat{\mathbf{\Phi}} = [\hat{\phi}(\mathbf{x}_1),\ldots,\hat{\phi}(\mathbf{x}_n)]^T$ and $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is a column vector of ones. We can express $\mathbf{\Phi} = \mathbf{H}\hat{\mathbf{\Phi}}$. Hence,

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{H}\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^T\mathbf{H} = \mathbf{H}\hat{\mathbf{K}}\mathbf{H}. \tag{1}$$

2.3 Locally Linear Translation in the Feature Space

For each similarity constraint pair $(\mathbf{x}_k,\mathbf{x}_l) \in \mathcal{S}$, we define a translation vector

$$\mathbf{a}_k = [\phi(\mathbf{x}_l) - \phi(\mathbf{x}_k)]\,/2,$$

pointing from $\phi(\mathbf{x}_k)$ to the midpoint of $\phi(\mathbf{x}_k)$ and $\phi(\mathbf{x}_l)$. $\phi(\mathbf{x}_k)$ and $\phi(\mathbf{x}_l)$ are translated towards their midpoint, indicated by vector $\mathbf{a}_k$ and $-\mathbf{a}_k$, respectively. For each dissimilarity constraint pair $(\mathbf{x}_u,\mathbf{x}_v) \in \mathcal{D}$, we define a translation vector

$$\mathbf{b}_u = [\phi(\mathbf{x}_u) - \phi(\mathbf{x}_v)]\,/2,$$

pointing from the midpoint of $\phi(\mathbf{x}_u)$ and $\phi(\mathbf{x}_v)$ to $\phi(\mathbf{x}_u)$. $\phi(\mathbf{x}_u)$ and $\phi(\mathbf{x}_v)$ are moved away from their midpoint, indicated by vector $\mathbf{b}_k$ and $-\mathbf{b}_k$, respectively. If a data point is involved in more than one point pair, we consider the linear translation for each pair separately.

To preserve the topological relationships between data points, we apply the above linear translations to other data points in the neighborhood sets of those points involved in the similarity or dissimilarity constraints. Therefore, the new

[1] We use RBF kernel in this paper.

location $\psi(\mathbf{x}_i)$ of $\phi(\mathbf{x}_i)$ in the feature space is the overall translation effected by possibly all similarity and dissimilarity constraints (and hence neighborhood sets):

$$\psi_i = \phi_i + \alpha \sum_{(k,l)\in\mathbf{S}} \pi_{ki}\mathbf{a}_k + \beta \sum_{(u,v)\in\mathbf{D}} \pi_{ui}\mathbf{b}_u, \tag{2}$$

where π_{ki} and π_{ui} are Gaussian functions. If $\phi(\mathbf{x}_i)$ is closer to $\phi(\mathbf{x}_k)$ than $\phi(\mathbf{x}_l)$,

$$\pi_{ki} = \exp\left[-\frac{1}{2}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i))^T \mathbf{\Sigma}_k^{-1}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i))\right],$$

otherwise

$$\pi_{ki} = -\exp\left[-\frac{1}{2}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i))^T \mathbf{\Sigma}_k^{-1}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i))\right],$$

with $\mathbf{\Sigma}_k$ being the covariance matrix. For simplicity, we use a hyperspherical Gaussian function, meaning that the covariance matrix is diagonal with all diagonal entries being σ^2. Thus π_{ki} can be rewritten as

$$\pi_{ki} = \exp[-\|\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i)\|^2/(2\sigma^2)],$$

if $\phi(\mathbf{x}_i)$ is closer to $\phi(\mathbf{x}_k)$ than $\phi(\mathbf{x}_l)$, and

$$\pi_{ki} = -\exp[-\|\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i)\|^2/(2\sigma^2)],$$

otherwise. For dissimilar constraints, π_{ui} is defined in the same way.

Let $\mathbf{\Pi}^S = [\pi_{ki}]$ and $\mathbf{\Pi}^D = [\pi_{ui}]$, with $k = 1, \ldots, |\mathcal{S}|, u = 1, \ldots, |\mathcal{D}|$, and $i = 1, \ldots, n$. $|\mathcal{S}|$ and $|\mathcal{D}|$ denote the numbers of similarity and dissimilarity constraints in $\mathcal{S}$ and $\mathcal{D}$, respectively. Let $\mathbf{A}$ and $\mathbf{B}$ denote the translation matrices decided by the similarity and dissimilarity constraints, respectively, with each of the $|\mathcal{S}|$ or $|\mathcal{D}|$ columns representing a different translation vector. From (2), the adaptation of data set $\mathbf{\Phi}$ can be expressed as

$$\mathbf{\Psi} = \mathbf{\Phi} + \alpha\mathbf{A}\mathbf{\Pi}^S + \beta\mathbf{B}\mathbf{\Pi}^D. \tag{3}$$

2.4 Iterative Kernel Matrix Adaptation

We first apply the centering transform as described in Section 2.2 to give the kernel matrix $\mathbf{K}$. Then, we compute the new kernel matrix $\tilde{\mathbf{K}}$ after performing the locally linear translation defined in Section 2.3. It is worthy to note that we can use kernel trick to avoid explicit embedding of data points in the feature space. Metric learning with pairwise constraints is actually formulated as a kernel learning problem. Let us omit the derivation details due to page limit. The kernel

matrix will be adapted as follows:

$$\begin{aligned}\tilde{\mathbf{K}} = \mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{K} &+ \alpha[(\mathbf{K}^{\Phi S_2} - \mathbf{K}^{\Phi S_1})\mathbf{\Pi}^S + \mathbf{\Pi}^{S^T}(\mathbf{K}^{S_2\Phi} - \mathbf{K}^{S_1\Phi})]/2 \\ &+ \beta[(\mathbf{K}^{\Phi D_1} - \mathbf{K}^{\Phi D_2})\mathbf{\Pi}^D + \mathbf{\Pi}^{D^T}(\mathbf{K}^{D_1\Phi} - \mathbf{K}^{D_2\Phi})]/2 \\ &+ \alpha^2\mathbf{\Pi}^{S^T}(\mathbf{K}^{S_2S_2} - 2\mathbf{K}^{S_2S_1} + \mathbf{K}^{S_1S_1})\mathbf{\Pi}^S/4 \\ &+ \beta^2\mathbf{\Pi}^{D^T}(\mathbf{K}^{D_1D_1} - 2\mathbf{K}^{D_1D_2} + \mathbf{K}^{D_2D_2})\mathbf{\Pi}^D/4 \\ &+ \alpha\beta\mathbf{\Pi}^{S^T}(\mathbf{K}^{S_2D_1} - \mathbf{K}^{S_2D_2} - \mathbf{K}^{S_1D_1} + \mathbf{K}^{S_1D_2})\mathbf{\Pi}^D/4 \\ &+ \alpha\beta\mathbf{\Pi}^{D^T}(\mathbf{K}^{D_1S_2} - \mathbf{K}^{D_2S_2} - \mathbf{K}^{D_1S_2} + \mathbf{K}^{D_2S_1})\mathbf{\Pi}^S/4.\end{aligned} \tag{4}$$

We define $\mathbf{K}^{PQ}$ as a submatrix of $\mathbf{K}$, with P and Q specifying the indices of data points, corresponding to the rows and columns extracted from $\mathbf{K}$. Φ represents the indices of all data points in $\mathcal{X}$. $S_1 = \{k|(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{S}\}$, and $S_2 = \{l|(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{S}\}$. D_1 and D_2 are defined in a similar way.

As for the Gaussian window parameter, we make σ^2 depend on the average of squared Euclidean distances between all point pairs in the feature space:

$$\sigma^2 = \frac{\theta}{n^2}\sum_{i,j=1}^{n}\|\phi_i - \phi_j\|^2 = \frac{2\theta}{n}\left[\mathrm{Tr}(\mathbf{K}) - n\bar{\mathbf{K}}\right],$$

where $\bar{\mathbf{K}}$ represents the mean value of the elements in matrix $\mathbf{K}$. The parameter θ is set to be the same ($= 0.5$) for all data sets in our experiments. Note that $\|\phi(\mathbf{x}_k) - \phi(\mathbf{x}_i)\|^2 = k(\mathbf{x}_k, \mathbf{x}_k) - 2\ k(\mathbf{x}_k, \mathbf{x}_i) + k(\mathbf{x}_i, \mathbf{x}_i)$, so the Gaussian functions defined in our method can be computed directly using the kernel matrix $\mathbf{K}$. The parameters α and β in Equation (2) and (3) determine the learning rate and the relative effects of the similarity and dissimilarity constraints. We set them to $1/|\mathcal{S}|$ and $1/|\mathcal{D}|$, respectively.

The locations of the data points in the feature space are translated iteratively, with the kernel matrix being adapted accordingly. As in [18], the Gaussian window parameter and learning rate should decrease over time to increase the local specificity gradually. The iterative adaptation procedure will stop when either there is no significant change in the kernel matrix or the maximum number of iterations T has been reached. We summarize the metric learning algorithm below:

3 Experiments

To assess the efficacy of LLMA, we perform extensive experiments on toy data as well as real data from the UCI Machine Learning Repository.[2]

[2] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Input: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathcal{S}$, $\mathcal{D}$
Begin
  $t = 0$
  Centering transform to get initial kernel $\mathbf{K}^0$ (Equation (1))
  Repeat {
    Update kernel matrix $\mathbf{K}^t$ to $\mathbf{K}^{t+1}$ (Equation (4))
    Decrease parameters σ^2, α, β
    $t = t + 1$
    $\lambda = \|\mathbf{K}^{t+1} - \mathbf{K}^t\|$
  } Until ($\lambda <$threshold or $t = T$)
End

Fig. 1. Iterative metric learning by kernel matrix adaptation

3.1 Experimental Setting

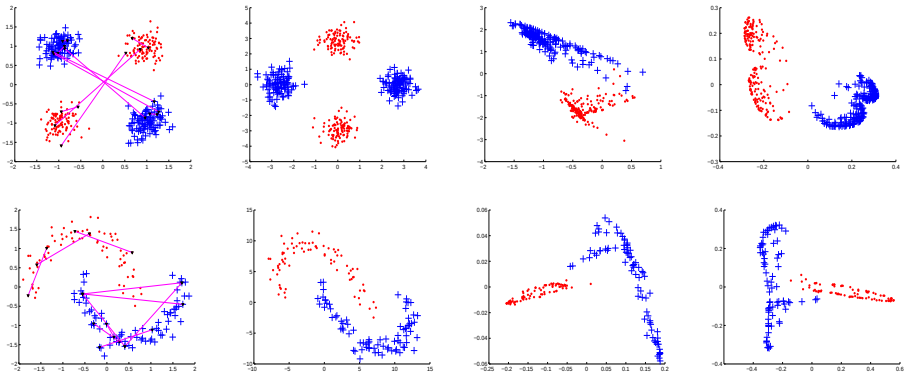
We compare our proposed method described in Section 2 with two previous methods. The first method is relevant component analysis (RCA) [14]. As a metric learning method, RCA changes the input space by a global linear transformation which assigns large weights to relevant dimensions and low weights to irrelevant dimensions. Another method is locally linear metric adaptation (LLMA) [18], which is more general in that it is linear locally but nonlinear globally. We also use the Euclidean distance without metric learning for baseline comparison. Since both RCA and LLMA make use of pairwise similarity constraints only, we also use such supervisory information only for our method. In summary, the following four distance measures for the k-means clustering algorithm are included in our comparative study (the short forms inside brackets will be used subsequently for convenience):

1. k-means without metric learning (Euclidean)
2. k-means with RCA for metric learning (RCA)
3. k-means with LLMA for metric learning (LLMA)
4. k-means with our kernel-based method for metric learning (Kernel-based)

There exist many performance measures for clustering tasks. As in [18, 17, 14], we use the Rand index [19] to quantify the agreement of the clustering result with the ground truth. For each data set, we randomly generate 20 different $\mathcal{S}$ sets to provide pairwise similarity constraints to the clustering task. In addition, for each $\mathcal{S}$ set, we perform 20 runs of k-means with different random initializations and report the average Rand index over the 20 runs.

3.2 Experiments on Synthetic Data

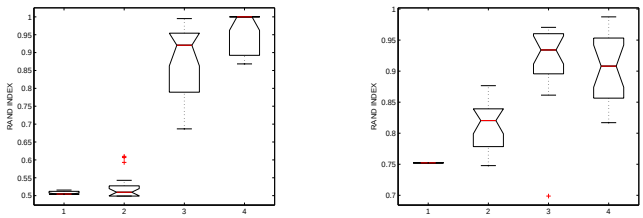
Figure 2 demonstrates the power of our proposed metric learning method on two synthetic data sets. One is the XOR data set and the other is the 2-moon data set which is commonly used in some recent semi-supervised learning work. However,



(a) data (b) RCA (c) LLMA (d) Kernel-based

Fig. 2. Comparison of different metric learning methods on two synthetic data sets. (a) original synthetic data sets; and the data sets after applying (b) RCA; (c) LLMA; (d) our kernel-based method.

the difference is that we do not exploit the underlying manifold structure here. Instead, only some limited pairwise similarity constraints are provided. The two data sets are shown in the first column of Figure 2. Data points with the same mark and color belong to the same cluster. Point pairs corresponding to the similarity constraints are connected by solid lines. The second, third and fourth columns show the data sets after applying RCA, LLMA and our kernel-based method. Obviously, RCA, which performs globally linear metric learning, cannot give satisfactory results. The performance of LLMA is significantly better, although some points from the two classes are quite close to each other. On the other hand, our kernel-based approach can group the data points well according to their class.



(a) (b)

Fig. 3. Clustering results for (a) XOR data set and (b) 2-moon data set.

We further perform some semi-supervised clustering experiments on the XOR and 2-moon data sets. The results are shown in Figure 3. For each trial, 10 point pairs are randomly selected to form $\mathcal{S}$.

3.3 Experiments on UCI Data

Table 1. Nine UCI data sets used in our experiments

Data set	n	c	d	$\lvert\mathcal{S}\rvert$
Soybean	47	4	35	10
Protein	116	6	20	15
Iris	150	3	4	20
Wine	178	3	13	20
Ionosphere	351	2	34	30
Boston housing	506	3	13	40
Breast cancer	569	2	31	50
Balance	625	3	4	40
Diabetes	768	2	8	50

To access the efficacy of our metric learning methods for real-world data sets, we further perform experiments for semi-supervised clustering tasks on some data sets from the UCI Machine Learning Repository. Table 1 summarizes the characteristics of nine UCI data sets used in our experiments. The number of data points n, the number of clusters c, the number of features d, and the number of randomly selected similarity constraints $\mathcal{S}$ are shown in each row of Table 1.

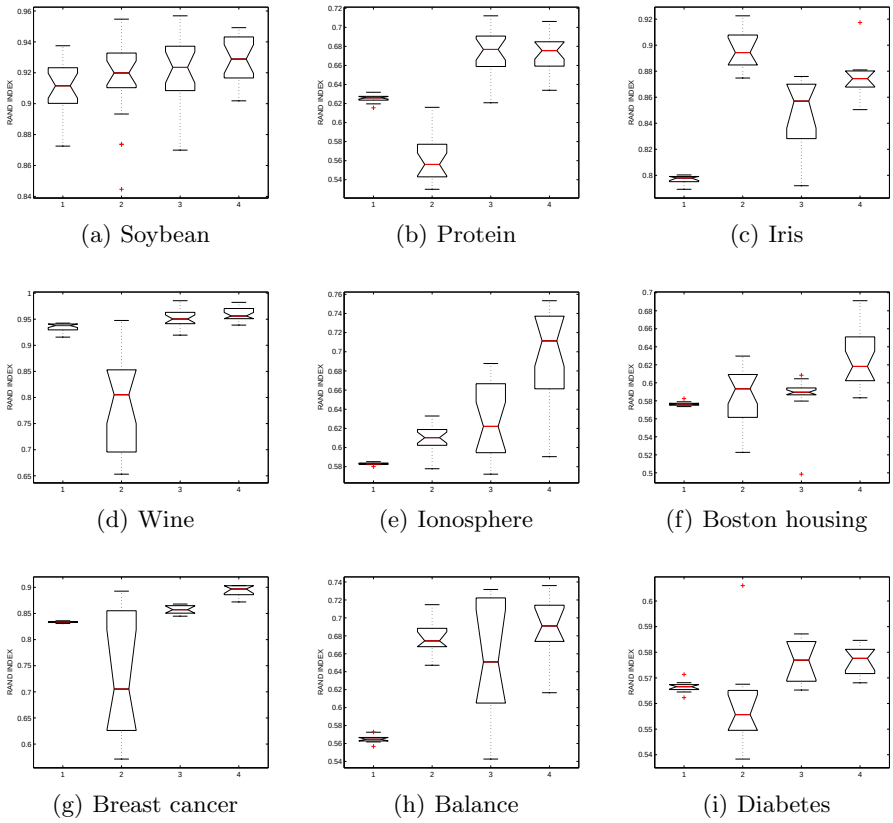
The clustering results using different clustering algorithms numbered as in Section 3.1 are shown in Figure 4. From the clustering results, we can see that our kernel-based method outperforms RCA and is comparable to or even better than LLMA.

4 Concluding Remarks

In this paper, we have proposed a new metric learning method based on semi-supervised learning. Unlike previous methods which perform metric learning in the input space, our method performs metric learning in the kernel-induced feature space by iterative kernel matrix adaptation. We neither have to formulate the metric learning as an optimization problem nor need to compute explicit embedding of the data points in the feature space. These characteristics make our method more powerful for solving some difficult clustering tasks as demonstrated through the synthetic and UCI data sets.

Although our kernel-based metric learning method is simple and yet quite effective, there is still much room for improvement. We have only considered translation in the feature space, which is a restrictive form of locally linear transformation. One possible extension is to generalize it to more general linear transformation. Another limitation is that our method cannot preserve the topology structure during the metric learning procedure.

Except for the illustrative examples shown in Figure 2, we access the performance of our proposed metric learning method by semi-supervised clustering.



(a) Soybean (b) Protein (c) Iris

(d) Wine (e) Ionosphere (f) Boston housing

(g) Breast cancer (h) Balance (i) Diabetes

Fig. 4. Clustering results for nine UCI data sets.

There also exist other machine learning tasks that can be used to evaluate different metric learning methods. One possible task is content-based image retrieval (CBIR), whose performance depends critically on the distance measure between images. We will study this and other applications in the future.

Acknowledgments

The research described in this paper is supported by two grants, CA03/04.EG01 (which is part of HKBU2/03/C) and HKUST6174/04E, from the Research Grants Council of the Hong Kong Special Administrative Region, China.

References

1. K. Fukunaga and L. Hostetler. Optimization of k-nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3):320–326, 1973.
2. J.H. Friedman. Flexible metric nearest neighbor classification. Technical report, Department of Statistics, Stanford University, Stanford, CA, USA, November 1994.

3. T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996.
4. D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.
5. J. Peng, D.R. Heisterkamp, and H.K. Dai. Adaptive kernel metric nearest neighbor classification. In *Proceedings of the Sixteenth International Conference on Pattern Recognition*, volume 3, pages 33–36, Québec City, Québec, Canada, 11–15 August 2002.
6. T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
7. S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 19–26, Sydney, Australia, 8–12 July 2002.
8. J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14(1):217–239, 2002.
9. Z. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1450–1452, Acapulco, Mexico, 9–15 August 2003.
10. K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, Standord, CA, USA, 2000.
11. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with backgroud knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, Williamstown, MA, USA, 2001.
12. Z. Lu and T.K. Lee. Semi-supervised learning with penalized probabilistic clustering. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, USA, 2005.
13. N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA, 2004.
14. A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the Twentieh International Conference on Machine Learning*, pages 11–18, Washington, DC, USA, 21–24 August 2003.
15. T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 393–400, Banff, Alberta, Canada, 4–8 August 2004.
16. M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA, 2004.
17. E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, Cambridge, MA, USA, 2003.
18. H. Chang and D.Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 153–160, Banff, Alberta, Canada, 4–8 August 2004.
19. W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.