

Learning A Mixture of Sparse Distance Metrics for Classification and Dimensionality Reduction

Yi Hong¹, Quannan Li¹, Jiayan Jiang¹, and Zhuowen Tu^{1,2}

Lab of Neuro Imaging and Department of Computer Science, UCLA¹

yihong@cs.ucla.edu, {quannan.li, jet.jiang, zhuowen.tu}@loni.ucla.edu

Microsoft Research Asia²

Abstract

This paper extends the neighborhood components analysis method (NCA) to learning a mixture of sparse distance metrics for classification and dimensionality reduction. We emphasize two important properties in the recent learning literature, locality and sparsity, and (1) pursue a set of local distance metrics by maximizing a conditional likelihood of observed data; and (2) add ℓ_1 -norm of eigenvalues of the distance metric to favor low rank matrices of fewer parameters. Experimental results on standard UCI machine learning datasets, face recognition datasets, and image categorization datasets demonstrate the feasibility of our approach for both distance metric learning and dimensionality reduction.

1. Introduction

K-nearest neighbor classifier (kNN) [4] is one of the most popular classification schemes in machine learning and it has been widely adopted. The performance of kNN is often sensitive to the adopted distance metric. Commonly used Euclidean distance metric is straightforward to implement but often produces inferior results than a well-tuned distance metric learned from data. In the past, a wealthy body of distance metric learning algorithms [6, 25, 27, 21, 3, 10, 26] have been proposed to improve the performance of kNN. Among them the neighborhood components analysis (NCA) [6] is effective and most related to our approach. Given a set of training data with their corresponding labels, NCA learns a distance metric by maximizing the log-likelihood of observed data in a transformed space. NCA [6] has been successfully applied to data visualization, dimensionality reduction, and classification. However, NCA assumes the same Mahalanobis distance metric on the entire data space, which is sometimes inappropriate; in addition, it requires the specification of the number of dimensions. High-dimensional data often observe sparsity and locality,

which should be taken into account in the metric learning process. Motivated by the above observations, we propose a method, mixture of sparse neighborhood components analysis (msNCA), which (1) adds ℓ_1 -norm of eigenvalues of the distance metric to enforce sparsity, and (2) encourages locally adaptive distance metrics in the data space.

2. Related Work

Distance metric learning is an active research area in machine learning. In this section, we give a brief discussion of some representative approaches such as neighborhood components analysis (NCA) [6], large margin nearest neighbor classifier (LMNN) [25], and information-theoretic metric learning (ITML) [3]. NCA was proposed in [6] for data visualization, dimensionality reduction, and classification. It learns a Mahalanobis distance metric by finding a linear transformation such that a stochastic variant of kNN achieves the minimal leave-one-out classification error in the transformed space. By specifying the dimension of the transformation matrix as e.g. 2 or 3, NCA projects the high-dimensional data onto the corresponding subspace, while preserving its intrinsic structure for classification. LMNN is a large margin induced distance metric learning method [25]. It aims to find a Mahalanobis distance metric such that instances from different classes are well separated by a large margin within the neighborhood. In addition, the authors formulated the above problem as a semidefinite program with linear constraints and solved it globally by standard convex optimization strategies. An information theory based distance metric learning approach, termed as ITML, was proposed in [3]. ITML considers the distance metric learning problem as the minimization of the differential relative entropy between two multivariate Gaussians. It can be further expressed as a Bregman optimization problem with linear constraints.

Methods like NCA, LMNN and ITML are able to identify better distance metrics than the commonly used Euclidean distance metric for classification. Experimental re-

sults have also supported that the classification accuracy of kNN significantly increases when it adopts more appropriate distance metrics. However, the above approaches assume the same metric uniformly applied on the entire data space, which might be suboptimal in practice. To mitigate the above limitation, a local distance metric learning algorithm has been proposed in [5], which assigns a specific distance metric to each instance in the training set. Apart from its computational burden, this local distance metric learning algorithm does not make use of any global information and may lead to overfitting.

Our proposed msNCA combines advantages of both global and local aspects of distance metric learning. It obtains a set of local distance metrics and gating functions by minimizing the negative log-likelihood of observed data. In addition, an appropriate distance metric might observe the property of low rank with sparse eigenvalues. This constraint of sparse eigenvalues is useful for distance metric learning, but has been given less attentions. Recently, two sparse distance metric learning algorithms have been proposed in [27] and [20]. Other approaches [2] use random projection in the classification. msNCA differs from the above sparse distance metric learning algorithms on where to put ℓ_1 -norm. In particular, ℓ_1 -norm is evaluated on elements of the distance metric in [27] and on the determinant of the distance metric in [20]; while msNCA puts ℓ_1 -norm on eigenvalues of the distance metric.

3. Mixtures of Sparse Neighborhood Components Analysis

Let $D = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i \in \{1, 2, \dots, \ell\}, i = 1, 2, \dots, n\}$ be the set of training data, the Mahalanobis distance between two instances x_j and x_i is:

$$d(x_j, x_i | M) = (x_j - x_i)^\top M (x_j - x_i), \quad (1)$$

given $M \in \mathbb{R}^{d \times d}$. If a soft neighbor assignment is considered, then the probability of the instance x_j of being assigned in the neighborhood $\mathcal{N}(x_i)$ of the instance x_i can be approximated as:

$$q(x_j \in \mathcal{N}(x_i)) = \frac{\exp\{-d(x_j, x_i | M) / \|M\|_F\}}{\sum_{l=1, l \neq i}^n \exp\{-d(x_l, x_i | M) / \|M\|_F\}}, \quad (2)$$

where $j \neq i, x_j, x_i \in D$, and $\|M\|_F$ is the Frobenius norm added as the normalization term for mitigating the scaling effect of M on the probability $q(x_j \in \mathcal{N}(x_i))$ [6, 13, 23]. Based on the nearest neighbor classification rule, the conditional distribution $p(y_i | x_i; M)$ of the instance x_i being classified into the class with the label y_i can be approximated as:

$$p(y_i | x_i; M) \approx \sum_{j=1, j \neq i, y_j = y_i}^n q(x_j \in \mathcal{N}(x_i)), \quad (3)$$

where (x_i, y_i) is the training data and $i = 1, 2, \dots, n$. Note that the right-hand side of eqn. (3) was originally proposed as an estimation of the leave-one-out classification accuracy of kNN [6]. It is also quite natural to consider it as an approximation to the conditional distribution $p(y_i | x_i; M)$ [13, 22, 23]. Based on eqn. (3), we get the negative log-likelihood $L(M)$ of observed data as:

$$L(M) = - \sum_{i=1}^n \log \left(\sum_{\substack{j=1 \\ j \neq i \\ y_j = y_i}}^n \exp \left\{ \frac{-(x_j - x_i)^\top M (x_j - x_i)}{\|M\|_F} \right\} \right) + \sum_{i=1}^n \log \left(\sum_{l=1, l \neq i}^n \exp \left\{ -(x_l - x_i)^\top M (x_l - x_i) / \|M\|_F \right\} \right). \quad (4)$$

In eqn. (4), M is positive semi-definite, thus can be decomposed as $M = Q^\top Q$, where $Q \in \mathbb{R}^{d' \times d}$. Instead of minimizing $L(M)$ with respect to M by the gradient decent method, NCA firstly identifies a linear transformation Q that minimizes the above objective function, then M is given by $Q^\top Q$. The above trick is particularly useful for data visualization, where the best choice of d' should be 2 or 3. However, for other applications of distance metric learning and dimensionality reduction, we do not have any prior knowledge about what a good value of d' should be. In both cases, a reasonable value of d' could be d and Q is thus a square matrix. Therefore, optimizing $L(M)$ with respect to M is equivalent to optimizing $L(M)$ with respect to Q up to rotation. To simplify our mathematical derivations, we minimize the above objective function with respect to M directly in this paper. This coincides with most current literatures in distance metric learning [25, 3]. It is worth to mention that $L(M)$ is non-convex with respect to M . However, experimental results in [6] suggested that the gradient decent method was often able to guarantee a high quality M that minimizes $L(M)$ locally.

3.1. Sparsity

High dimensional data usually locate in a subspace with low intrinsic dimensionality [11]. kNN will become more effective with respect both to the classification accuracy and the search cost at test time, if distances between data are measured in this intrinsic subspace. Therefore, the distance metric M is encouraged to be low rank with sparse eigenvalues from the standpoint of distance metric learning. Another advantage of low rank distance metric is to decrease the complexity of kNN w.r.t. its number of parameters if the distance metric is considered as its component, and thus, likely increase its generalization ability on test data. Moreover, one major application of NCA is dimensionality reduction, that aims to project high dimensional data into low dimensional subspaces. For that reason like M , the linear

transformation Q should also be low rank with sparse eigenvalues.

It is therefore desirable to encourage M to have sparse eigenvalues. This can be realized by adding ℓ_1 -norm of eigenvalues of M into the objective function (4). Let $\{e_i | i = 1, 2, \dots, d\}$ denote all d eigenvalues of M , then ℓ_1 -norm of eigenvalues is equal to $\sum_{k=1}^d |e_k|$. By adding ℓ_1 -norm of eigenvalues into $L(M)$, it leads to a sparse version of NCA that minimizes the following objective function $SL(M)$:

$$SL(M) = L(M) + \lambda \sum_{k=1}^d |e_k|, \quad (5)$$

where $\lambda \geq 0$. Since M is positive semi-definite with non-negative eigenvalues and $\{e_k \geq 0, k = 1, 2, \dots, d\}$. Therefore, ℓ_1 -norm of eigenvalues of M can be computed from its trace:

$$\sum_{i=1}^d |e_i| = \sum_{i=1}^d e_i = \mathbf{Tr}(M), \quad (6)$$

where $\mathbf{Tr}(M)$ is the trace of the distance metric M . After combining (5) and (6), $SL(M)$ can be refined as:

$$SL(M) = L(M) + \lambda \mathbf{Tr}(M). \quad (7)$$

The derivative of eqn. (7) w.r.t. to M can then be derived. It is noted that most of the existing literatures aim to learn a sparse matrix and their ℓ_1 -norm is put on all elements in the matrix [28, 16, 27]; while our purpose is to learn a low rank Mahalanobis distance metric with sparse eigenvalues and our ℓ_1 -norm is thus evaluated on eigenvalues.

3.2. Mixture model

Next, we study eqn. (4), which evaluates the negative log-likelihood of observed data. Note eqn. (4) makes the assumption that the data share the same Mahalanobis distance metric. This condition holds when all the data points locate in a single linear subspace. However, many real datasets are not homogeneous and it is hard to find a single distance metric well describing all pairwise similarities. Based on this observation, it might be worth pursuing a mixture model [12] with adaptive local metrics.

Inspired by the divide-and-conquer approach, we design the mixture of NCA by splitting the data space into several regions and train a set of local models to fit the data in these regions. Let $p(y_i|x_i; v, M)$ be the mixtures of conditional distributions:

$$p(y_i|x_i; v, M) = \sum_{s=1}^S \pi_s(x_i) p_s(y_i|x_i, M_s), \quad (8)$$

where S is the number of components and $\{M_1, M_2, \dots, M_s\}$ is a set of Mahalanobis distance metrics. In eqn. (8), each component $p_s(y_i|x_i, M_s)$ is defined as:

$$p_s(y_i|x_i, M_s) = \frac{\sum_{j=1, j \neq i, y_j=y_i}^n \exp\{-d(x_j, x_i|M_s)/\|M_s\|_F\}}{\sum_{l=1, l \neq i}^n \exp\{-d(x_l, x_i|M_s)/\|M_s\|_F\}},$$

and $\pi_s(x)$ is the gating function for $s = \{1, 2, \dots, S\}$. A commonly used gating function is the multi-class logistic regression model defined as:

$$\pi_s(x) = \frac{\exp\{v_s^\top x + b_s\}}{\sum_{t=1}^S \exp\{v_t^\top x + b_t\}}, \quad (9)$$

where $v_s \in \mathbb{R}^d$ are parameters to learn [12, 7]. The gating function of a multi-class logistic regression is able to divide the feature space into several regions. However, it is in favor of coefficients with large absolute values. Here, we propose a regularized gating function by adding ℓ_1 -norm of coefficients to penalize this bias and the negative log-likelihood $L(v, M)$ becomes:

$$L(v, M) = - \sum_{i=1}^n \log \left(\sum_{s=1}^S \pi_s(x_i) p(y_i|x_i, M_s) \right) + \sum_{s=1}^S \eta_s \left(\sum_{w=1}^{d+1} |v_s^w| \right), \quad (10)$$

where $\eta_s \geq 0$, for $s = 1, 2, \dots, S$.

3.3. msNCA

In the above two subsections, we provided two extensions to the existing NCA: sparse NCA and mixtures of NCA. Now we introduce a mixture of sparse NCA (msNCA) that combine the above two extensions together. msNCA can thus be formulated as the following constrained optimization problem:

$$\min G(v, M) = - \sum_{i=1}^n \log \left(\sum_{s=1}^S \pi_s(x_i) p(y_i|x_i, M_s) \right) + \sum_{s=1}^S \lambda_s \mathbf{Tr}(M_s) + \sum_{s=1}^S \eta_s \left(\sum_{w=1}^{d+1} |v_s^w| \right), \quad (11)$$

s.t.

$$M_s \succeq 0,$$

for $s = 1, 2, \dots, S$. ℓ_1 -norm regularizers in eqn. (11) can be considered as double-exponential priors on both M_s and v_s :

$$p(M_s) \propto \prod_{w=1}^d \exp\{-|e_s^w|/\tau_s\} = \exp\{-\mathbf{Tr}(M_s)/\tau_s\}, \quad (12)$$

and

$$p(v_s) \propto \prod_{w=1}^{d+1} \exp\{-|v_s^w|/\xi_s\}, \quad (13)$$

where $\tau_s \geq 0$, $\xi_s \geq 0$, for $s = 1, 2, \dots, S$. Instead of maximizing the log-likelihood of observed data like NCA, msNCA minimizes the posterior of $p(v, M|x, y)$ as:

$$p(v, M|x, y) \propto p(y|x; v, M) p(v) p(M), \quad (14)$$

where parameters v and M are assumed to be independent [24]. We define $p_s(y_{ji}|x_i, M_s)$ as:

$$p_s(y_{ji}|x_i, M_s) = \frac{\exp\{-(x_j - x_i)^\top M_s (x_j - x_i) / \|M_s\|_F\}}{\sum_{l \neq i}^n \exp\{-(x_l - x_i)^\top M_s (x_l - x_i) / \|M_s\|_F\}},$$

for $j \neq i$ and $j, i = 1, 2, \dots, n$. Differentiating $G(v, M)$ with respect to both v and M respectively, we get:

$$\begin{aligned} \frac{\partial G(v, M)}{\partial v_s} = & \\ & - \sum_{i=1}^n \left[\left(\frac{p_s(y_i|x_i, M_s)}{\sum_{t=1}^S \pi_t(x_i) p_t(y_i|x_i, M_t)} - 1 \right) \pi_s(x_i) x_i \right] + \eta_s \mathbf{e}_s, \end{aligned} \quad (15)$$

and

$$\begin{aligned} \frac{\partial G(v, M)}{\partial M_s} = & \\ & - \sum_{i=1}^n \frac{\pi_s(x_i)}{\sum_{t=1}^S \pi_t(x_i) p_t(y_i|x_i, M_t)} \cdot \frac{\partial p_s(y_i|x_i, M_s)}{\partial M_s} + \lambda_s \mathbf{I}, \end{aligned} \quad (16)$$

where

$$\mathbf{e}_s^w = \begin{cases} +1 & \text{if } v_s^w \geq 0 \\ -1 & \text{otherwise} \end{cases}, \quad (17)$$

and

$$\begin{aligned} \frac{\partial p_s(y_i|x_i, M_s)}{\partial M_s} = & - \sum_{\substack{j=1, \\ j \neq i, \\ y_j=y_i}}^n \frac{p_s(y_{ji}|x_i, M_s)}{\|M_s\|_F} \\ & \left[(x_j - x_i)(x_j - x_i)^\top - \frac{(x_j - x_i)^\top M_s (x_j - x_i)}{\|M_s\|_F} \frac{M_s}{\|M_s\|_F} \right] \\ & + p_s(y_i|x_i, M_s) \sum_{\substack{l=1, \\ l \neq i}}^n \frac{p_s(y_{li}|x_i, M_s)}{\|M_s\|_F} \\ & \left[(x_l - x_i)(x_l - x_i)^\top - \frac{(x_l - x_i)^\top M_s (x_l - x_i)}{\|M_s\|_F} \frac{M_s}{\|M_s\|_F} \right]. \end{aligned}$$

The update rule at iteration h is thus:

$$v_s^{h+1} = v_s^h - \gamma_v \frac{\partial G(v, M)}{\partial v_s},$$

and

$$M_s^{h+1} = M_s^h - \gamma_m \frac{\partial G(v, M)}{\partial M_s},$$

where γ_v and γ_m are step sizes. To make sure

$$M_s \succeq 0,$$

we project the obtained metric M_s onto the convex set $C = \{M : M \succeq 0\}$ at each iteration with the following steps employed: (1) to use eigenvalue decomposition to decompose M_s as $U\Lambda U^\top$, where Λ is a diagonal metric; (2) to set all negative elements in Λ as 0, that is

$$M_s = U(\max\{0, \Lambda\})U^\top, \quad (18)$$

Table 1. Classification accuracy on six UCI datasets.

	Iris	Wine	Segment
SVM	96.16 ± 4.71	98.88 ± 2.34	90.73 ± 3.47
kNN	95.19 ± 1.06	96.38 ± 0.76	92.36 ± 0.31
LMNN	95.33 ± 4.05	97.78 ± 2.87	93.04 ± 2.24
ITML	96.44 ± 0.98	98.11 ± 0.95	92.13 ± 0.49
NCA	94.67 ± 3.26	98.33 ± 2.68	90.26 ± 0.77
msNCA	96.00 ± 0.57	99.44 ± 1.76	92.86 ± 0.58
	Iono	Vehicle	Waveform
SVM	83.51 ± 4.93	70.77 ± 7.41	85.96 ± 1.17
kNN	84.64 ± 0.66	72.87 ± 1.10	79.71 ± 0.67
LMNN	86.85 ± 5.79	78.08 ± 4.17	76.90 ± 1.63
ITML	88.39 ± 1.79	74.76 ± 2.79	79.61 ± 0.83
NCA	86.57 ± 5.10	74.04 ± 4.54	83.20 ± 1.09
msNCA	91.28 ± 6.34	82.45 ± 5.30	83.60 ± 1.23

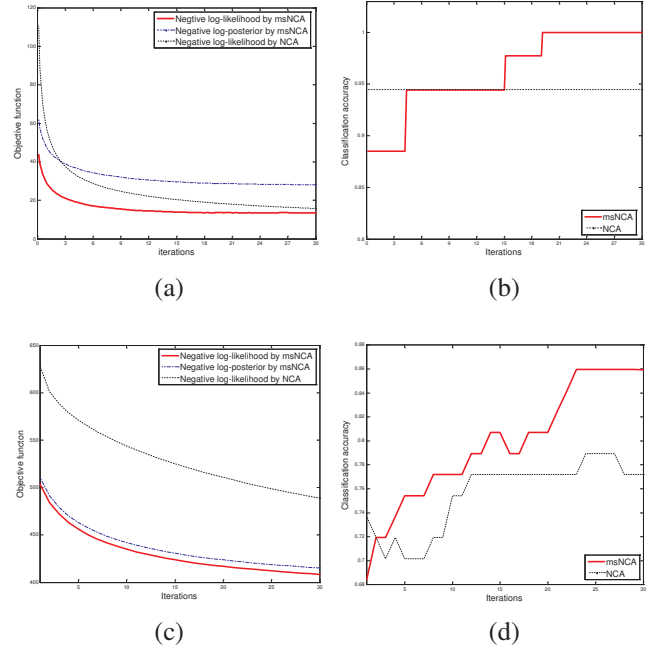


Figure 1. Values of objective functions and classification accuracies on testing data (a) Wine dataset; (b) Wine dataset; (c) Vehicle dataset; (d) Vehicle dataset.

for $s = 1, 2, \dots, S$ [8]. If the task is to do classification, for a new test example x we first determine which of these distance metrics M_1, M_2, \dots, M_s will be used, then calculate its label by kNN under this distance metric. If the task is to do dimension reduction, we compute a set of projection matrices $\{Q_1, Q_2, \dots, Q_s\}$ by $(U \max\{0, \sqrt{\Lambda}\})^\top$, then for each instance in the dataset determine which of these projection matrices should be used. During the testing process, each instance y is assigned to the distance metric M_k with the largest value of $\pi_k(y)$.

4. Experimental Results

In our experiments, msNCA is evaluated on UCI machine learning datasets, two face recognition datasets and two image categorization datasets.

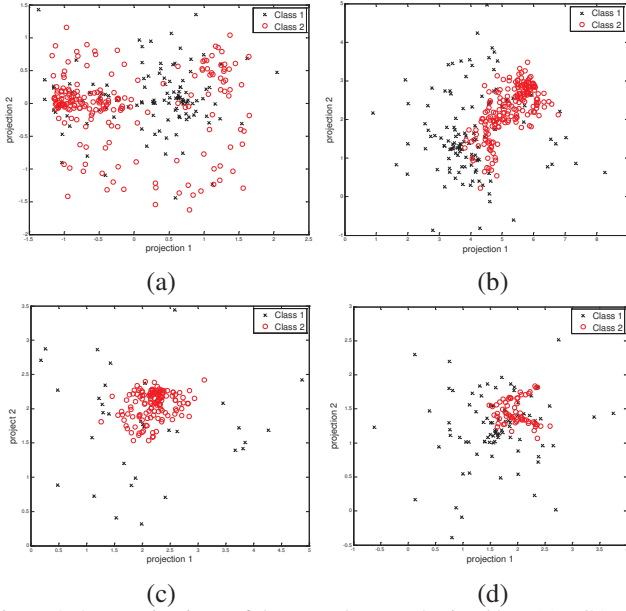


Figure 2. 2-D projections of the Iono dataset obtained by: (a) PCA; (b) NCA; (c) the first distance metric of msNCA; (d) the second distance metric of msNCA.

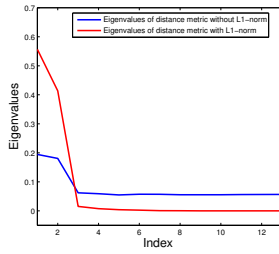


Figure 3. Eigenvalues of distance metrics with and without ℓ_1 -norm. It tells us that ℓ_1 -norm leads to distance metrics with sparse eigenvalues.

4.1. UCI machine learning datasets

4.1.1 Distance metric learning

To validate the feasibility of msNCA for distance metric learning, the classification accuracy of kNN with the distance metric learned by msNCA is compared with those obtained by (1) multi-class support vector machine (SVM) with linear kernel, (2) kNN with Euclidean distance metric (kNN), (3) kNN with the distance metrics learned by LMNN (LMNN), (4) kNN with distance metrics learned by ITML (ITML), and (5) kNN with the distance metrics learned by NCA (NCA). Parameters of msNCA were fixed in experiments as follows: the number of components $S = 4$, $\lambda_s = 0.02$ and $\eta_s = 0.02$ for $s = 1, 2, \dots, S$; the learning rates $\gamma_v = 0.02$ and $\gamma_m = 0.02$. The classification accuracy was evaluated by 10-fold cross-validation.

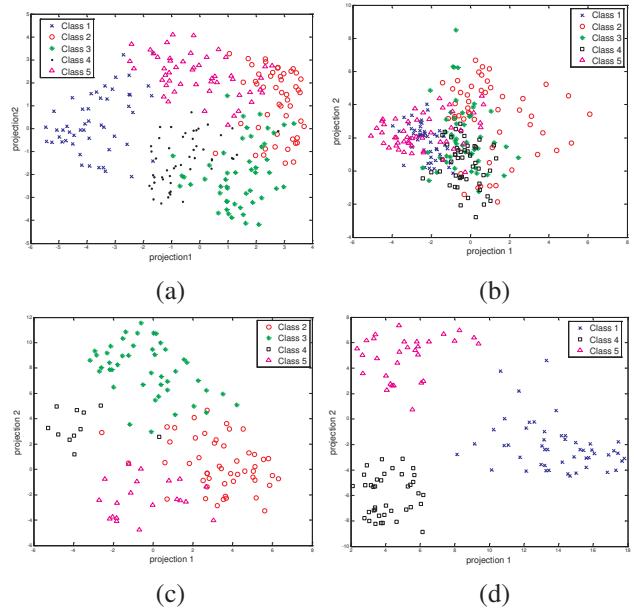


Figure 4. 2-D projections of the Semeion handwritten digit dataset (a) PCA; (b) NCA; (c) the first distance metric of msNCA; (d) the second distance metric of msNCA.

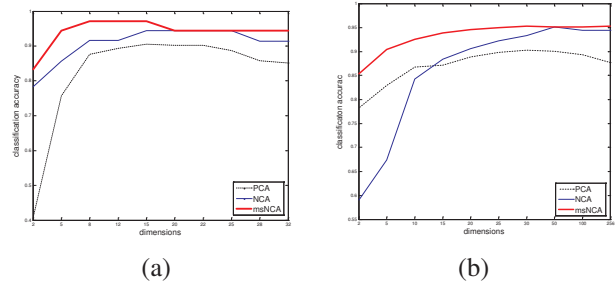


Figure 5. Classification accuracies of kNN on: (a) the Iono dataset, (b) the Semeion handwritten digit data set obtained by PCA, NCA and msNCA by varying the number of reduced dimensionality.

Table (1) shows the results. As discussed in the previous section, NCA also minimizes the negative log-likelihood of observed data. Fig. (1a) and Fig. (1c) display values of negative log-likelihood of observed data by NCA and msNCA. From Fig. (1), we can see that, the gap between values of negative log-likelihood of NCA and msNCA is quite large. This large gap of log-likelihood of observed data suggests that a single distance metric sometimes may not be sufficient to well fit the data and kNN with the mixtures of local distance metrics might be more appropriate. Fig. (1c) and Fig. (1d) show the classification accuracies of kNN with a single distance metric learned by NCA and kNN with the mixtures of distance metrics learned by msNCA and we can observe consistent results as in Fig. (1a) and Fig. (1c). Fig.(3) gives the comparison between distance met-

Table 2. Classification accuracy on AT&T face recognition dataset.

Methods	SVM	kNN	LMNN
Yale	81.67 ± 0.50	78.33 ± 4.72	94.02 ± 2.38
AT&T	95.83 ± 1.67	92.74 ± 1.99	97.81 ± 0.90
Methods	ITML	NCA	msNCA
Yale	88.48 ± 1.15	90.15 ± 1.20	93.67 ± 1.76
AT&T	97.22 ± 1.27	96.39 ± 2.09	98.24 ± 0.80

rics learned by NCA with and without ℓ_1 norm.

4.1.2 Dimensionality reduction

To validate the feasibility of msNCA for dimension reduction, we evaluate principle components analysis (PCA), NCA and msNCA on the Iono dataset and the Semeion handwritten digit data set. The Iono dataset has 351 instances and 32 features; while the Semeion handwritten digit data set contains 1,593 instances and 256 features. To simplify the visualization of our experimental results, we fix the number of components of msNCA as 2 and select a subset of the Semeion handwritten digit dataset that contains all instances from five classes for comparing PCA, NCA and msNCA. For each class, 50 instances are randomly picked for training and the remaining ones are used for testing. Fig. (2) and Fig. (4) show 2D projections of the Iono dataset and the Semeion handwritten digit dataset obtained by PCA, NCA and msNCA. Noted that in msNCA, the gating function divides the whole dataset into two subsets and each subset has a transformation matrix that projects instances in the subset onto a 2-D subspace. We observe from Fig. (2) and Fig. (4) that 2D projections of instances in the same class obtained by msNCA are close to each other, while instances in different classes become separate. Fig. (5) gives the classification accuracies of kNN on the Iono dataset and the Semeion handwritten digit dataset obtained by PCA, NCA and msNCA by varying the number of reduced dimensionality. We observe from Fig. (5) that on both of the Iono dataset and the Semeion handwritten digit dataset msNCA achieved consistently higher accuracies than PCA and NCA. It therefore can be concluded from the above experimental results that msNCA is able to achieve a better dimensionality reduction than PCA and NCA.

4.2. Face recognition

msNCA was tested on two face recognition datasets: Yale and AT&T. The Yale dataset includes 165 faces of 15 different persons and the AT&T dataset contains 400 faces of 40 different persons. We applied PCA to obtain 30-dimensional eigenfaces as descriptors that is enough to capture 96% variance [25]. We randomly selected 7 images of each person for training and the other images for

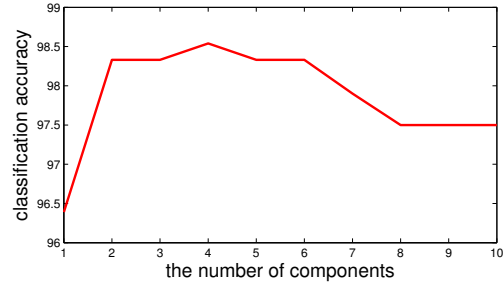


Figure 6. Classification accuracy of msNCA with different number of components on AT&T face dataset.

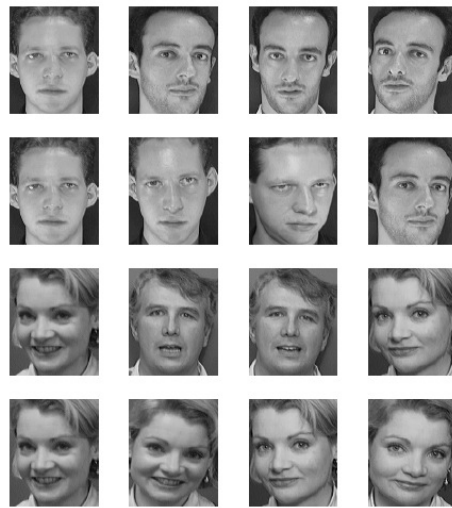


Figure 7. Two face images that were classified incorrectly by NCA, but correctly by msNCA. The first and the third rows are two images and their 3-nearest neighbors obtained by NCA; the second and the fourth rows are the same two images and their 3-nearest neighbors obtained by msNCA. The first column is the test image and the following three columns are 3-nearest neighbors of the test image. It is observed that neighborhoods obtained by msNCA are much cleaner than NCA.

testing. Table (2) shows the average results after 50 executions. It is observed from Table (2) that learned Mahalanobis distance metric significantly improved the performance of kNN. In addition, LMNN and msNCA achieved higher classification accuracies than NCA, ITML and SVM both on Yale and AT&T face datasets. Fig. (8) shows the 2D projections of the Yale face dataset obtained by PCA, NCA and msNCA. It is observed from Fig. (8) that a single distance metric sometimes can not describe the similarities between instances well and the mixtures of local distance metric are more effective than a single distance metric. To further demonstrate the effectiveness of msNCA, we compared the neighborhood of images obtained by NCA and msNCA.

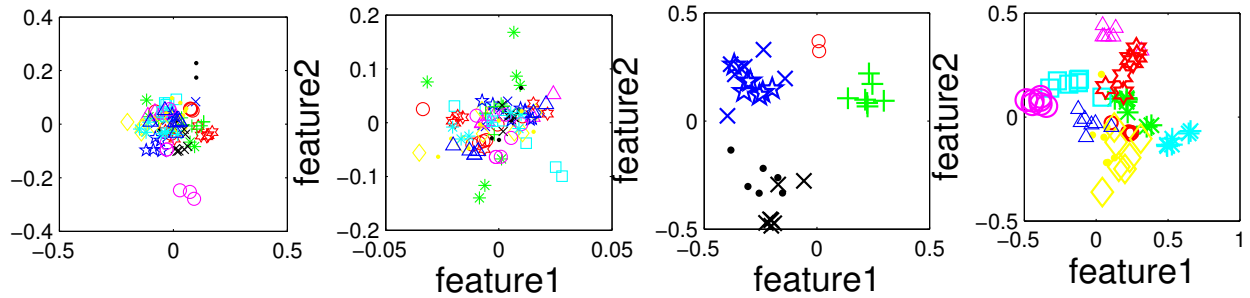


Figure 8. 2-D projections of the Yale face dataset (a) PCA; (b) NCA; (c) the first distance metric of msNCA; (d) the second distance metric of msNCA.

Table 3. Accuracies on image categorization datasets.

Methods	SVM	kNN	LMNN
Graz-02	79.6 \pm 0.4	75.6 \pm 1.1	78.3 \pm 0.2
Graz01 Person	82.9 \pm 0.9	76.4 \pm 0.9	83.9 \pm 0.6
Methods	ITML	NCA	msNCA
Graz-02	77.4 \pm 0.6	80.1 \pm 1.2	82.2 \pm 0.9
Graz01 Person	80.2 \pm 0.7	82.4 \pm 0.3	85.8 \pm 0.5

Two face images in AT&T face dataset, which were classified incorrectly by NCA, but correctly by msNCA, together with their 3-nearest neighbors are given in Fig. (7). In Fig. (7), the first and the third rows are two test face images together with their 3-nearest neighbors in the training set under the distance metric obtained by NCA; while the second and the fourth rows are the same images and their 3-nearest neighbors obtained by msNCA. By comparing their neighborhoods, we can conclude from Fig. (7) that msNCA is able to obtain a much cleaner neighborhood than NCA. In addition, we studied the effect of the number of components on the performance of msNCA. Fig. (6) shows that a large number of components may lead to the overfitting of msNCA.

State-of-the-art results on Yale and AT&T face recognition datasets are given as follows: In [19], the authors adopted V1-like descriptors and achieved perfect performance ($\geq 98\%$) both on Yale and AT&T datasets. LMNN was tested on AT&T face recognition dataset in [25] and its classification accuracy was 97.3%. Other state-of-the-art results include 81.7% on Yale and 97.4% on AT&T [1], and 90.6% on Yale and 98.40% on AT&T [9], where 5 images were used for training and 5 images for testing. msNCA was also tested under this experimental setup and its classification accuracy is 91.33% on Yale and 97.50% on AT&T.

4.3. Image categorization

We tested msNCA on Graz-01 and Graz-02 datasets for image categorization. We adopted the same experimental setup as [18], where 100 positive and 100 negative images were used for training. All color images were firstly trans-

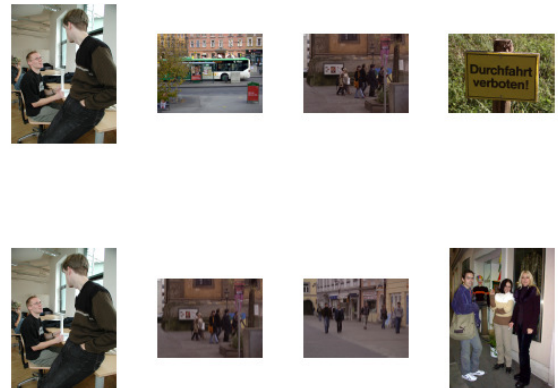


Figure 9. Three nearest neighbors obtained by NCA and msNCA. The first row by NCA and the second row by msNCA.

formed into gray-level images; dense sift descriptors were extracted and quantized by k-means clustering algorithm to 300 components. PCA was employed to remove redundant dimensions and the dimension was reduced to 30. Therefore, each image was represented as a vector of 30 components. Table (3) gives classification accuracies of SVM, kNN, LMNN, ITML, NCA and msNCA on Graz-01 Person and Graz-02 datasets. Current literatures about experimental results on these two datasets include 79.5% in [14] and 80.5% in [18] on Graz-01 person dataset, and 76.1% in [17] and 82.7% in [15] on Graz-02 dataset. Fig. (10) shows 2D projections of the Graz-02 dataset. Fig. (9) gives 3-nearest neighborhoods of a particular image in the Graz-01 dataset.

5. Conclusions

In this paper, we have proposed a distance metric learning approach using a mixture of sparse neighborhood components analysis (msNCA). msNCA extends existing distance metric learning approaches in two aspects. The first aspect adds ℓ_1 -norm of eigenvalues of Mahalanobis distance metrics, thus bias to low-rank matrices with sparse eigenvalues. The second extension localizes NCA by gat-

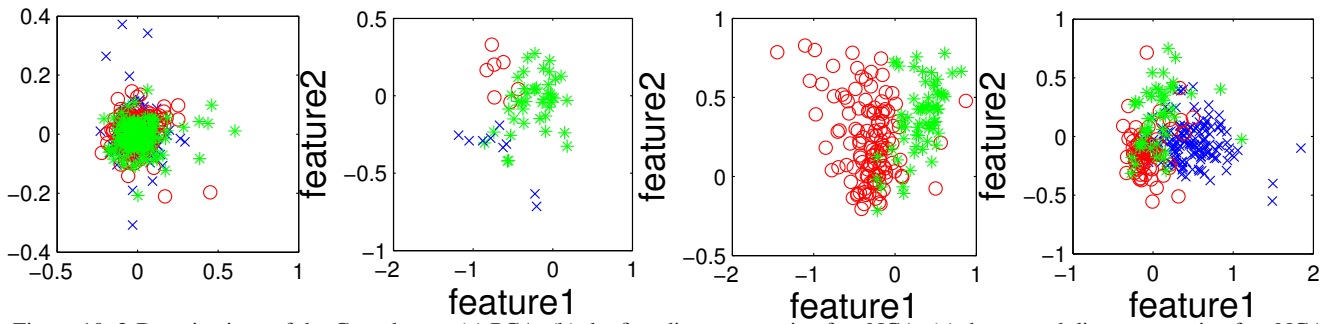


Figure 10. 2-D projections of the Graz dataset (a) PCA; (b) the first distance metric of msNCA; (c) the second distance metric of msNCA; (d) the third distance metric of msNCA.

ing functions. A set of local Mahalanobis distance metrics are obtained and mixed. We formulate our problem as a constrained optimization problem for minimizing the log-likelihood of observed data with exponential priors. We have tested msNCA on several standard datasets and demonstrated its feasibility for both distance metric learning and dimensionality reduction. Advantages of msNCA over existing distance metric learning approaches include: (1) adding prior knowledge of low-rank matrices with sparse eigenvalues; (2) combining advantages of flexibility and strong resistance to overfitting of local and global distance metric learning by mixture models.

6. Acknowledgment

This work is supported by Office of Naval Research Award N000140910099 and NSF CAREER award IIS-0844566.

References

- [1] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning a spatially smooth subspace for face recognition. *CVPR*, 2007.
- [2] S. Dasgupta and Y. Freund. Random projection trees for vector quantization. *IEEE Transactions on Information Theory*, 2009.
- [3] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. *ICML*, 2007.
- [4] R. Duda, P. Hart, and D. Stork. *Pattern classification*. Wiley, 2000.
- [5] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. *NIPS*, 2006.
- [6] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. *NIPS*, 2005.
- [7] M. Gonen and E. Alpaydin. Localized multiple kernel learning. *ICML*, 2008.
- [8] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, New York, 1985.
- [9] G. Hua and A. Akbarzadeh. A robust elastic and partial matching metric for face recognition. *ICCV*, 2009.
- [10] P. Jain, B. Kulis, and I. Dhillon. Inductive regularized learning of kernel functions. *NIPS*, 2010.
- [11] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 1984.
- [12] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 1994.
- [13] P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. *ICML*, 2006.
- [14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.
- [15] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. *ICCV*, 2007.
- [16] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse lda. *ICML*, 2006.
- [17] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. *ECCV*, 2006.
- [18] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE PAMI*, 2006.
- [19] N. Pinto, J. DiCarlo, and D. Cox. Establishing good benchmarks and baselines for face recognition. *ECCV*, 2008.
- [20] G. Qi, J. Tang, Z. Zha, T. Chua, and H. Zhang. An efficient sparse metric learning in high-dimensional space via l_1 -penalized log-determinant regularization. *ICML*, 2009.
- [21] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. *ECCV*, 2002.
- [22] N. Singh-Miller and M. Collins. Learning label embeddings for nearest-neighbor multi-class classification with an application to speech recognition. *NIPS*, 2009.
- [23] N. Sprague. Predictive projections. *IJCAI*, 2009.
- [24] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1996.
- [25] K. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *NIPS*, 2006.
- [26] K. Weinberger and L. Saul. Fast solvers and efficient implementations for distance metric learning. *ICML*, 2008.
- [27] Y. Ying, K. Huang, and C. Campbell. Sparse metric learning via smooth optimization. *NIPS*, 2009.
- [28] H. Zou, T. Hastie, and R. Tibshirani. Sparse principle component analysis. *Technical Report, statistics department, Stanford University*, 2004.