

ADAPTIVE AFFINITY MATRIX FOR UNSUPERVISED METRIC LEARNING

Yaoyi Li, Junxuan Chen, Yiru Zhao, Hongtao Lu*

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
Department of Computer Science and Engineering, Shanghai Jiao Tong University, P.R.China
{dsamuel, chenjunxuan, yiru.zhao, htlu}@sjtu.edu.cn

ABSTRACT

Spectral clustering is one of the most popular clustering approaches with the capability to handle some challenging clustering problems. Only a little work of spectral clustering focuses on the explicit linear map which can be viewed as the distance metric learning. In practice, the selection of the affinity matrix exhibits a tremendous impact on the unsupervised learning. In this paper, we propose a novel method, dubbed Adaptive Affinity Matrix (AdaAM), to learn an adaptive affinity matrix and derive a distance metric. We assume the affinity matrix to be positive semidefinite with ability to quantify the pairwise dissimilarity. Our method is based on posing the optimization of objective function as a spectral decomposition problem. The provided matrix can be regarded as the optimal representation of pairwise relationship on the manifold. Extensive experiments on a number of image data sets show the effectiveness and efficiency of AdaAM.

Index Terms— Affinity Learning, Feature Projection, Dimensionality Reduction, Spectral Clustering

1. INTRODUCTION

Spectral clustering methods which are based on eigendecomposition demonstrate splendid performance on many real-world challenge data sets. During the past decades, a series of spectral clustering methods have been proposed: Multidimensional Scaling (MDS) [1], Local Linear Embedding (LLE) [2], Isomap [3], Laplacian Eigenmaps [4] and variant of Spectral Clustering [5]. There are three shortages of spectral clustering methods mentioned above. First, these approaches only provide the embedding map of the training data. The out-of-sample extension is not straightforward. Second, The complexity of these approaches relies on the number of data points. Third, the performance of spectral clustering methods highly depend on the robustness of the affinity graph.

Many important progresses [6, 7, 8, 9, 10, 11, 12, 13, 14] have been made to mitigate the above issues of the spectral

clustering. Locality Preserving Projections (LPP) proposed in [7] introduces a linear projection obtained from Laplacian Eigenmaps. Their work provides a linear approximation of the embedding mapping, which reduces the time complexity and achieves out-of-sample extension straightforwardly. The linear embedding gives a metric learning perspective of the spectral clustering. Nie, Wang, and Huang proposed the Projected Clustering with Adaptive Neighbors (PCAN) in [14] where they regard the pairwise similarity as an extra variable to be solved in the optimization problem and they set a penalty of the rank of graph Laplacian to restrict specific components in the affinity matrix. With this framework, PCAN alternately update affinity matrix and projection. Although some affinity learning algorithms have been proposed in recent years, the technique of choosing an appropriate affinity matrix is still remained to be addressed.

Our goal is to extract more adaptive similarity information with minimal extra time consumption for the linear approximation of spectral clustering. Such information will take the objective of locality preserving rather than only the distance between images into consideration. Inspired by the recent progress on scalable spectral clustering [10] and data similarity learning [14], we propose a novel approach dubbed Adaptive Affinity Matrix (AdaAM). Our affinity matrix is relatively dense and can capture both global and local information. Specifically, AdaAM decomposes the affinity graph into a product of two low-rank identical matrices. As the ideal case described in [5], if we assume the pairwise affinity in the same class are exceedingly similar, the affinity matrix may turn into a low-rank matrix. We optimize the decomposed matrix with the similar scheme of spectral clustering. The affinity graph obtained by optimization is used as an intermediate affinity matrix, firstly. With the combination of the intermediate affinity matrix and the k -NN affinity graph derived by the heat kernel, we figure out a final adaptive affinity matrix from a naive spectral clustering. We conduct the affinity graph with the data projection and apply LPP to this specific graph to learn a metric for clustering.

We illustrate the effective and efficiency of the proposed approach for clustering on image data sets. We show the advantage of AdaAM for challenging data sets by comparing our approach with k nearest neighborhood heat kernel (k NN)

*Corresponding author

This paper is supported by NSFC (No. 61272247, 61533012, 61472075), the 863 National High Technology Research and Development Program of China (SS2015AA020501) and the Major Basic Research Program of Shanghai Science and Technology Committee (15JC1400103).

[4] and some other state-of-the-art algorithms in Section 3.

Our main contribution is that we integrate the affinity matrix learning into the framework of spectral clustering with the same paradigm, and we employ the low rank trick to make our approach more efficient.

2. ADAPTIVE AFFINITY MATRIX

2.1. Notation

In this paper, we write all matrices as uppercase (English or Greek alphabet) and vectors are written as lowercase. The vector with all elements one is denoted by $\mathbf{1}$. H is the centering matrix denoted by $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$. The origin data matrix is denoted by $X \in \mathbb{R}^{n \times d}$, where n is the number of the data points and d is the dimension of the data. X is assumed to be normalized with zero mean, i.e. $X = HX$. The denotation x_i means the i -th data point vector. We also denote the linear projection by A and denote the metric matrix by $M = A^T A$. Hence the Mahalanobis distance based on is $dis_m(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$. The k -NN heat kernel matrix is denoted by $W \in \mathbb{R}^{n \times n}$ with

$$w_{ij} = \begin{cases} \exp(-\frac{\|x_i - x_j\|_2}{t}), & x_i \in \mathcal{N}_k(x_j) \text{ or } x_j \in \mathcal{N}_k(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\mathcal{N}_k(x)$ is the set of k nearest neighbors of x . The corresponding Laplacian matrix is denoted by $L = D - W$, where D is the diagonal matrix with $d_{ii} = \sum_j w_{ij}$. We also denote both intermediate increment and final adaptive affinity matrix as Δ , the corresponding diagonal weight matrix and Laplacian matrix as D_Δ and $L_\Delta = D_\Delta - \Delta$.

2.2. Intermediate Affinity Matrix

We separate our algorithm into two parts, intermediate affinity matrix and final adaptive affinity matrix. In this section, we will introduce the first part. For the i -th data point x_i , we connect any the data point x_i to the data point x_j with the similarity δ_{ij} . With the hope that small Euclidean distance between two data points leads to a large similarity, we aim to choose δ_{ij} to minimize the following objective function

$$\min \sum_{i,j} \|x_i - x_j\|_2^2 \delta_{ij} \quad (2)$$

under appropriate constraints, where δ_{ij} is the ij -th element of the intermediate affinity matrix Δ .

Different from PCAN [14], we reformulate the equation with graph Laplacian,

$$\min tr(X^T L_\Delta X) \quad (3)$$

under some constraints.

With a straightforward thought we can decompose the Laplacian into two identical matrices, since the graph Laplacian is a positive semidefinite matrix in general. We show this thought is not appropriate in our framework as follows.

If we assume that

$$L_\Delta = UU^T \quad (4)$$

where $U \in \mathbb{R}^{n \times s}$ is a column orthogonal matrix with $U^T U = I$. With the relaxing of the constraints, we finally need to solve the problem

$$\begin{aligned} U &= \arg \min_{U^T U = I} tr(X^T U U^T X) \\ \Rightarrow U &= \arg \min_{U^T U = I} tr(U^T X X^T U) \end{aligned} \quad (5)$$

If we assume the product of matrix X to be K (i.e. $K = X X^T$), the Eq. (5) gives a simple form of the Laplacian Eigenmaps

This optimization problem can be solved by selecting eigenvectors of matrix K corresponding to several smallest eigenvalues. However, K is a low-rank matrix generally with $d \ll n$ and the eigenvectors of K minimizing the objective function in Eq. (5) is in the null space of X . Hence, the solution of above problem is not unique. Inspired by LSC [10] we assume the affinity matrix to be a positive semidefinite matrix and decompose it into the product of a matrix $P \in \mathbb{R}^{n \times t}$ with orthogonal columns and P^T instead of decomposing the Laplacian matrix, where t is the expected rank of Δ .

Therefore we reformulate Eq. (3) as

$$\min_{P^T P = I} tr(X^T D_\Delta X) + tr(X^T (-P P^T) X) \quad (6)$$

where we abandon the properties that connected weight is non-negative and the graph Laplacian is positive semidefinite. The negative connected weights in Δ can be used to measure the dissimilarity between data points. We will show that the solution of this optimization problem makes D_Δ equal to $\mathbf{0}$.

For the first part of Eq. (6), we can write it as

$$\begin{aligned} \min \sum_{i=1}^n \|x_i\|_2^2 d_{\Delta ii} \\ \text{s.t. } P^T P = I \\ d_{\Delta ii} = (P P^T \mathbf{1})_i \end{aligned} \quad (7)$$

Let $z = (\|x_1\|_2^2, \|x_2\|_2^2, \dots, \|x_n\|_2^2)^T$. With a Lagrange multipliers λ , the one dimensional situation of problem (7) can be rewritten as

$$\min z^T p p^T \mathbf{1} - \lambda(p^T p - 1) \quad (8)$$

Finally, the minimization problem (7) reduces to finding the eigenvector corresponding to the minimum eigenvalue of the problem $\mathbf{1} z^T p = \lambda p$. Because the matrix $\mathbf{1} z^T$ has rank

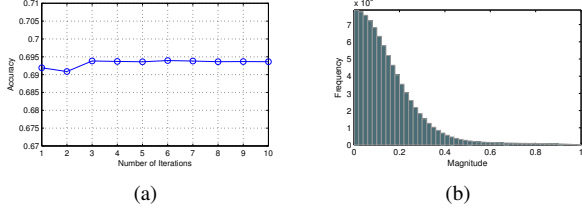


Fig. 1. (a) The evaluation of the clustering performance with different times iterative computation on the data set USPS. The contribution to accuracy made by iteration is less than 0.5%. (b) The histogram of the element magnitude of the final adaptive affinity matrix obtained from data set USPS.

one, there is only one nonzero eigenvalue $\sum_{i=1}^n \|x_i\|_2^2$, which implies $\lambda = 0$. Hence, for the P satisfying problem (7) with arbitrary column number less than n , we have $z^T P P^T \mathbf{1} = 0$. It is equivalent to

$$\sum_{i=1}^n \|x_i\|_2^2 d_{\Delta ii} = 0 \quad (9)$$

Generally, in real-world data set, $\|x_i\|_3 \neq 0$ always holds, thus, the P minimizing the first part of the objective function (6) has the property $D_{\Delta} = \mathbf{0}$. Meanwhile the set of all P with the property $D_{\Delta} = 0$ is the solution of Eq. (7).

The matrix P , which minimizes the second part of the objective function (6), is given by the maximum eigenvalue to the eigen problem:

$$(X X^T) p = \lambda p \Rightarrow \mathbf{1} X X^T p = \lambda \mathbf{1} p \quad (10)$$

As the data X has zero mean, we have $\lambda \mathbf{1} p = \mathbf{1} X X^T p = 0$. Therefore, for the maximum eigenvalue which is larger than 0, the corresponding eigenvector always satisfies $\mathbf{1} p = 0$. Let the minimum solution of the second part of problem (6) be $P = (p_1, p_2, \dots, p_t)$. We have

$$\mathbf{1}^T P = \mathbf{0} \Rightarrow \mathbf{1}^T P P^T = \mathbf{0} \Rightarrow d_{\Delta ii} = 0 \quad (11)$$

which means that the property $D_{\Delta} = \mathbf{0}$ holds for the optimal solution of the second part of (6) and the solution is in the set of the solution of Eq. (7). Therefore the solution of the second part of Eq. (6) can also optimize the object function (7) and the solution of the optimization problem (6) makes D_{Δ} equal to $\mathbf{0}$. The objective function (6) can be reduced to

$$P = \arg \max_{P^T P = I} \text{tr}(P^T X X^T P) \quad (12)$$

which has the solution as singular value decomposition of X with complexity relies on d rather than n . We obtain the intermediate affinity matrix $\Delta = P P^T$ from the distribution of origin data with similarity and dissimilarity information. The graph Laplacian of Δ is $L_{\Delta} = D_{\Delta} - \Delta = -\Delta$.

To mitigate the impact of noise and rank reducing problem, we apply sparsification to Δ . We will discuss the sparsification further in Section 2.4.

Algorithm 1 Adaptive Affinity Matrix

Input:

Data points $X \in \mathbb{R}^{n \times d}$; cluster number c ; neighborhood size k ; reduced dimension m ;

Output:

Mahalanobis metric M and linear projection A .

- 1: Construct the k -NN heat kernel W , the corresponding diagonal weight matrix D and the Laplacian matrix L ;
 - 2: Compute the P with orthogonal columns according to Eq. (12) for the intermediate affinity matrix $\Delta = P P^T$;
 - 3: Get the linear projection matrix A according to Eq. (13);
 - 4: Produce a new matrix P according to Eq. (16) for the final adaptive affinity matrix $\Delta = P P^T$;
 - 5: Get linear projection $A \in \mathbb{R}^{m \times d}$ and Mahalanobis metric $M = A^T A$ by applying LPP to the affinity matrix $\Delta + D$;
-

2.3. Final Adaptive Affinity Matrix

In this section, we formulate a naive linear spectral clustering and provide the final adaptive affinity matrix.

With the intermediate affinity matrix Δ , we can solve the following problem for a linear projection A :

$$a = \arg \min_{a^T a = 1} \text{tr}(a^T X^T (L + L_{\Delta}) X a) \quad (13)$$

where a is the one-dimension case of A and $L + L_{\Delta}$ is the combination of the Laplacian of k -NN heat kernel and the intermediate affinity matrix. The projection vector a is given by the minimum eigenvalue of the eigen problem:

$$X^T (L - \Delta) X a = \lambda a \quad (14)$$

Subsequently, to compute L_{Δ} of Eq. (13) given A , we rewrite the affinity optimization problem with the linear projection matrix A as we did in Eq. (6)

$$P = \arg \min_{P^T P = I} \left(c + \text{tr}(A^T X^T D_{\Delta} X A) + \text{tr}(A^T X^T (-P P^T) X A) \right) \quad (15)$$

where we assume the final adaptive affinity matrix to be $\Delta = P P^T$ and $c = \text{tr}(A^T X^T L X A)$. The property $D_{\Delta} = \mathbf{0}$ still holds, because of the zero mean of $X A$. Therefore, Eq. (15) reduces to

$$P = \arg \max_{P^T P = I} \text{tr}(P^T X A A^T X^T P) \quad (16)$$

This can be solved by singular value decomposition of matrix $X A$ and taking the left-singular vectors which correspond to the largest singular values. We apply sparsification on the adaptive affinity matrix $\Delta = P P^T$ obtained from Eq. (16) and attain the sparse affinity matrix.

Intuitively, we can iterate Eq. (13) and Eq. (16) to minimize the value of objective function. However, as Fig. 1(a)

Table 1. Clustering accuracy on image data sets(%)

	AdaAM		k -NN		Cons- k NN		DN		ClustRF-Bi		PCAN- k Means		PCAN
	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	
UMIST	66.06	75.65	58.16	65.39	60.27	69.22	59.15	66.96	64.63	74.44	53.79	56.52	55.30
COIL20	74.72	87.29	71.89	81.18	75.53	84.31	71.95	82.01	76.50	85.07	72.28	83.75	81.74
USPS	69.36	69.61	68.25	68.35	68.21	68.34	68.08	68.31	58.74	65.90	64.04	67.95	64.20
MNIST	60.84	61.34	48.13	48.27	47.88	48.00	49.72	49.76	51.93	52.03	58.93	58.98	59.83
ExYaleB	54.36	57.87	24.17	26.76	25.63	28.75	24.21	27.42	23.10	26.43	25.74	27.63	25.89

shows, the adaptive affinity matrix with only once iteration performs well in practice and the continuing iterations show no remarkable outperformance.

Since the weight of nodes in the graph plays an important role in some algorithms and methods based on Normalized Cuts [15] like LPP has the constraint relying on D_Δ . In our approach we have $D_\Delta = \mathbf{0}$, therefore we add the weight matrix D computed from the k -NN heat kernel to our affinity matrix. Finally, we replace the affinity matrix in LPP with the matrix $\Delta + D$ to get the linear projection A and the metric matrix $M = A^T A$.

2.4. Sparsification Strategy

From the optimization problem (12) and (16), we can observe that the matrices XX^T and $XAA^T X^T$ are both low-rank matrix. Seeing that the solution of the optimization problem mentioned above is based on the singular value decomposition, this low-rank fact will result in that the column number of the solution P could be far less than the rank of XX^T and $XAA^T X^T$. This process will produce a low-rank affinity matrix which leads to a progressively rank decreasing in our approach. To prevent the rank decreasing happening, we implement sparsification in our approach. The sparsification strategy may mitigate the problem of noise edges as well.

Fig. 1 justifies our sparsification procedure by demonstrating the histogram of the magnitude of the final adaptive affinity matrix obtained from Eq. (16) without sparsification. We can observe that most elements of the affinity matrix concentrate in the range with small magnitude, and the sparsification procedure may reserve a portion of the affinity elements which are more representative.

Inspired by the thought of k -NN heat kernel, we sort all the elements of affinity matrix Δ by decreasing magnitude and only reserve the first t elements. We consider that the parameter t is better to be in inverse proportion to the number of clusters, in which case the average elements reserved for each cluster will be proportionate to the number of data points in each cluster. The t is selected by following equation:

$$t = \lfloor \frac{n^2}{\alpha c} \rfloor \quad (17)$$

where $\lfloor \cdot \rfloor$ is the floor function, n^2 is the number of elements in Δ , c is the number of clusters and α is a coefficient.

We set α to be 2.5 for the first sparsification in the computation of the intermediate affinity matrix and set α to be 5 for the second sparsification in the computation of the final adaptive affinity matrix. The α is decided by a rough parameter search, and it gives a stable performance in most data sets.

We summarize our algorithm in Algorithm 1. We set reduced dimension m to be the same as the number of classes

3. EXPERIMENTS

In this section, we conduct several experiments to demonstrate the effectiveness and efficiency of the proposed approach AdaAM.

3.1. Data Sets

We evaluate the proposed approach on five image data sets:

UMIST The UMIST Face Database consists of 575 images of 20 individuals with 220×220 pixels [16]. We use the images resized to 40×40 pixels in our experiments.

COIL20 A data set consists of 1,440 images of 20 objects with discarded background [17].

USPS The USPS handwritten digit database has 9,298 images of 10 digits with 16×16 pixels [18].

MNIST The MNIST database of handwritten digits has 70,000 images of 10 classes [19]. In our experiments, we select the first 10,000 images of this database.

ExYaleB The Extended Yale Face Database B consists of 2,414 cropped images with 38 individuals and around 64 images under different illuminations per individual [20].

The statistics of data sets are summarized in Tab. 2.

Table 2. Statistics of five benchmark data sets

Data set	# of instances	# of features	# of classes
UMIST	575	1600	20
COIL20	1440	1024	20
USPS	9298	256	10
MNIST	10000	784	10
ExYaleB	2414	1024	38

3.2. Compared Algorithms

We compare our approach with the other affinity learning algorithms described in Section Related Work. We adopt LPP

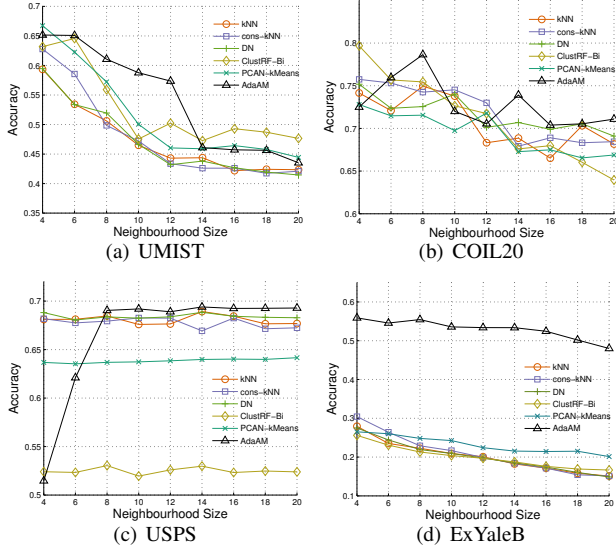


Fig. 2. Comparison between different with different of neighborhood size k

to the affinity matrices generated by these state-of-the-art approaches to obtain the distance metric.

Con- k NN Cons- k NN Consensus k -NNs [12] with the aim of selecting robust neighborhoods.

DN Dominant Neighborhoods proposed in [11].

ClustRF-Bi A special case of ClustRF-Strct [13], which is also proposed in [21, 22]. Due to the huge memory requirement of ClustRF-Strct on the data set with thousands instances, we implement this special case in our experiments.

PCAN Projected Clustering with Adaptive Neighbors proposed in [14]. Because PCAN is an algorithm which can generate the linear projection and clusters simultaneously, we denote the method combining the projection of PCAN and k -Means as PCAN- k Means and we also show the clustering result of PCAN in Tab. 1 for reference.

We also compare our approach with the k -NN heat kernel affinity matrix. We use k -NN to denote this typical approach.

3.3. Parameter Selection and Experiment Details

Because there is no validation data set in unsupervised learning tasks, for more general case, we impose the same parameter selection criteria on all the algorithms in our experiments. We set the size of neighborhood to be $k = \text{Round}(\log_2(n/c))$, where n is the number of data instances and c is the number of classes. We also set the projected dimension, which is equal to the rank of metric matrix, to be the same as the number of classes [5]. All the other parameters in our approach are fixed in every experiment.

We denote 10 times of k -Means as a round and select the clustering result with the minimal within-cluster sum as the result of each round of k -Means. We apply 100 rounds

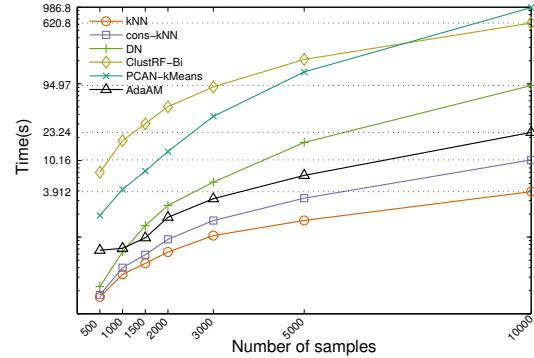


Fig. 3. Time consumption of six approaches with different number of data instances

k -Means to each algorithms for the evaluation of the performance (Tab. 1), 10 rounds k -Means for the experiment of the sensitivity to the neighborhood size k (Fig. 2) and one round k -Means for the experiment of execution time (Fig. 3).

3.4. Experiment Results

In the experiment of clustering accuracy, we evaluate the projection ability of AdaAM with other five algorithms on five benchmark data sets mentioned above. Tab. 1 gives the average and the maximal accuracy of 100 rounds k -Means of each model. From Tab. 1, we can observe that superiority of AdaAM on the task of the unsupervised metric learning. In most case, AdaAM performs much better than the other approaches. Our approach attains four best results of the average accuracy and five best maximal accuracy on five data sets. We can also observe that the proposed AdaAM decisively outperforms other five methods on ExYaleB data set. Different from the other data sets, the image data in ExYaleB are properly aligned and under different illumination. This difference makes some images more similar to the image in different class under the same illumination, which result in a high rank affinity matrix. Our approach is based on a low rank approximation of the optimal affinity matrix with the ability to handle such noises in the affinity matrix.

Since the neighborhood size k selection criteria is fixed in the experiment of accuracy, which may cause the loss of the best performance, we show the trend of accuracy according to the size of neighborhood in Fig. 2. Fig. 2 shows that AdaAM attains the best result in most cases and the sensitivity to the size of neighborhood is better or comparable to the other models. Since our approach is based on the low rank approximation of the optimal affinity matrix, it requires more information from the pairwise similarity. Hence, for small k , baseline methods are sometimes better than our approach.

Fig. 3 illustrates the efficiency of AdaAM by the semi-log graph of execution time with different number of data points selected from MNIST. It can be observed that our ap-

proach is a inexpensive algorithm in practice with much lower time consumption to PCAN- k Means, ClustRF-Bi and DN. We also show that AdaAM keeps approximately double time consumption to Cons- k NN with the much better performance.

4. CONCLUSION

In this paper, we present a novel affinity learning approach for unsupervised metric learning, called Adaptive Affinity Matrix (AdaAM). In our new affinity learning model, the affinity matrix is learned from the same framework of spectral clustering. More specifically, we show that the affinity learning can be reduced to a singular value decomposition problem. With the affinity matrix learned, the distance metric can be derived by some off-the-shelf approaches based on the affinity graph like LPP. Extensive experiments on clustering image data sets demonstrate the superiority of the proposed method AdaAM.

5. REFERENCES

- [1] Trevor F Cox and Michael AA Cox, *Multidimensional scaling*, CRC Press, 2000.
- [2] Sam T Roweis and Lawrence K Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [3] Joshua B Tenenbaum, Vin De Silva, and John C Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [4] Mikhail Belkin and Partha Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *NIPS*, 2001.
- [5] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al., “On spectral clustering: Analysis and an algorithm,” in *NIPS*, 2002.
- [6] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet, “Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering,” in *NIPS*, 2004.
- [7] Xiaofei He and Partha Niyogi, “Locality preserving projections,” in *NIPS*, 2004.
- [8] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik, “Spectral grouping using the nystrom method,” *IEEE TPAMI*, vol. 26, no. 2, pp. 214–225, 2004.
- [9] Donghui Yan, Ling Huang, and Michael I Jordan, “Fast approximate spectral clustering,” in *ACM SIGKDD*, 2009.
- [10] Xinlei Chen and Deng Cai, “Large scale spectral clustering with landmark-based representation,” in *AAAI*, 2011.
- [11] Massimiliano Pavan and Marcello Pelillo, “Dominant sets and pairwise clustering,” *IEEE TPAMI*, vol. 29, no. 1, pp. 167–172, 2007.
- [12] Vittal Premachandran and Ramakrishna Kakarala, “Consensus of k-nns for robust neighborhood selection on graph-based manifolds,” in *CVPR*, 2013.
- [13] Xiatian Zhu, Chen Change Loy, and Shaogang Gong, “Constructing robust affinity graphs for spectral clustering,” in *CVPR*, 2014.
- [14] Feiping Nie, Xiaoqian Wang, and Heng Huang, “Clustering and projected clustering with adaptive neighbors,” in *ACM SIGKDD*, 2014.
- [15] Jianbo Shi and Jitendra Malik, “Normalized cuts and image segmentation,” *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000.
- [16] Daniel B Graham and Nigel M Allinson, “Characterising virtual eigensignatures for general purpose face recognition,” in *Face Recognition*, pp. 446–456. Springer, 1998.
- [17] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al., “Columbia object image library (coil-20),” Tech. Rep., Technical Report CUCS-005-96, 1996.
- [18] Jonathan J Hull, “A database for handwritten text recognition research,” *IEEE TPAMI*, vol. 16, no. 5, pp. 550–554, 1994.
- [19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [20] K.C. Lee, J. Ho, and D. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE TPAMI*, vol. 27, no. 5, pp. 684–698, 2005.
- [21] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu, *Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning*, Now Publishers Inc., 2012.
- [22] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha, “Unsupervised random forest manifold alignment for lipreading,” in *ICCV*, 2013.