Clustering and Projected Clustering with Adaptive Neighbors

Feiping Nie, Xiaoqian Wang, Heng Huang

Presenter: **Dr. Heng Huang**

Department of Computer Science and Engineering,
The University of Texas at Arlington

August 27, 2014

Outline

Deterministic Neighbors and Probabilistic Neighbors

- Exploring the local connectivity of data is a successful strategy for clustering task, e.g., spectral clustering.
- For the i-th data point x_i, all the data points $\{x_1, x_2, ..., x_n\}$ can be connected to x_i as a neighbor with a weight s_{ij}.
- **Deterministic Neighbors.** The weights s_{ij} have only two values: 0 or 1.
- **Probabilistic Neighbors.** The weights s_{ij} have probabilistic values: $s_i^T \mathbf{1} = 1, 0 \leq s_i \leq 1$, where $s_i \in \mathbb{R}^{n \times 1}$ is a vector with the j-th element as s_{ij}.

Rules for Computing the Probabilistic Neighbors

- If there is no any distance information in the data, as a prior, all the data points can be the neighbors of x_i with the same probability $\frac{1}{n}$. It can be achieved by solving the following problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \le s_i \le 1} \sum_{j=1}^{n} s_{ij}^2 \tag{1.1}$$

- A smaller distance $\|x_i - x_j\|_2^2$ should be assigned a larger probability s_{ij}. It can be achieved by solving the following problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \le s_i \le 1} \sum_{j=1}^{n} \|x_i - x_j\|_2^2 \, s_{ij} \tag{1.2}$$

Computing the Probabilistic Neighbors

- Combining the two rules, the probabilistic neighbors of x_i can be computed by solving the following problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \le s_i \le 1} \sum_{j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \tag{1.3}$$

where γ is a parameter to balance the two terms.

- Denote $d_{ij}^x = \|x_i - x_j\|_2^2$, and denote $d_i^x \in \mathbb{R}^{n \times 1}$ as a vector with the j-th element as d_{ij}^x, then the problem (1.3) can be written in vector form as

$$\min_{s_i^T \mathbf{1}=1, 0 \le s_i \le 1} \left\| s_i + \frac{1}{2\gamma} d_i^x \right\|_2^2 \tag{1.4}$$

This problem can be solved with a closed form solution.

Probabilistic Neighbors Assignment for All the Data

- For each data point x_i, we can use Eq.(1.3) to assign its neighbors. Therefore, we can solve the following problem to assign the neighbors for all the data points:

$$\min_{\forall i, s_i^T \mathbf{1}=1, 0 \le s_i \le 1} \sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \tag{2.1}$$

- The obtained matrix S can be seen as a similarity matrix. However, the matrix S can not directly used for clustering, since in most cases the data are all connected with the S.

Clustering with Adaptive Neighbors (CAN)

- If we control the probabilistic neighbors assignment process, such that the data are partitioned into exactly c clusters, we can directly use the matrix S for clustering. We call the obtained neighbors with this process as **adaptive neighbors**.
- Suppose Ω is the set of similarity matrices which partitions the data into exactly c clusters. The Clustering with Adaptive Neighbors (CAN) is to solve the following problem:

$$\begin{aligned} &\min_{S\in\mathbb{R}^{n\times n}} \sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ &s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, S \in \Omega \end{aligned} \tag{2.2}$$

The CAN Clustering Algorithm

The problem (2.2) is very difficult to solve since the constraint $S \in \Omega$ is involved. We propose a novel algorithm as follows to solve this challenge. More details can be found in the paper.

Optimization Algorithm for the Problem (2.2)

Input: *Data matrix* $X \in \mathbb{R}^{n \times d}$, *cluster number* c, *parameter* γ, *a large enough* λ.

Output: $S \in \mathbb{R}^{n \times n}$ *with exact* c *connected components.*

1. *Initialize* S *by the optimal solution to the problem (2.1).*

2. *Update* F, *which is formed by the* c *eigenvectors of* $L_S = D_S - \frac{S^T + S}{2}$ *corresponding to the* c *smallest eigenvalues.*

3. *For each* i, *update the* i*-th row of* S *by solving the problem similar to (1.4).*

4. *Iteratively perform 2-3 until converges.*

Connection to K-means Clustering

It looks the problem (2.2) is far away from the problem of K-means clustering. However, we have the following theorem:

Theorem

When $\gamma \to \infty$, the problem (2.2) is equivalent to the problem of K-means.

Denoting $D^x = Diag(XX^T)\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T Diag(XX^T) - 2XX^T$, our objective can be written as:

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i \geq 0} Tr(S_i^T D_i^x) + \gamma \left\| S_i \right\|_F^2 \tag{2.3}$$

When $\gamma \to \infty$, our objective becomes:

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i \geq 0} \left\| S_i \right\|_F^2 \tag{2.4}$$

The optimal solution is that all the elements of S_i are equal to $\frac{1}{n_i}$.

Discussions on CAN Clustering Algorithm

- The theorem reveals that the proposed CAN clustering algorithm is to solve the K-means clustering problem when γ is very large.
- K-means can only partition data with spherical shape. When γ is not very large, the CAN clustering algorithm can partition data with arbitrary shape.
- We will also see in the experiments that the CAN clustering algorithm can find much better solution to the K-means problem even when γ is not very large.
- We also propose an effective method to automatically determine the value of γ in the CAN clustering algorithm. The details can be found in the paper.

Projected Clustering with Adaptive Neighbors (PCAN)

- Clustering high-dimensional data is an important and challenging problem in practice. In this paper, we also propose a Projected Clustering with Adaptive Neighbors (PCAN) to solve this problem.
- In contrast with the problem (2.2) for the CAN clustering, we solve the following problem for the PCAN clustering:

$$\begin{aligned} &\min_{S\in\mathbb{R}^{n\times n}, W\in\mathbb{R}^{d\times m}} \sum_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ &s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, W^T S_t W = I, S \in \Omega \end{aligned} \tag{3.1}$$

where $S_t = X^T H X$, $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix, and $W \in \mathbb{R}^{d\times m}$ is the projection matrix.

The PCAN Clustering Algorithm

Optimization Algorithm for the Problem (3.1)

Input: *Data matrix* $X \in \mathbb{R}^{n \times d}$, *cluster number* c, *reduced dimension* m, *parameter* γ, *a large enough* λ.

Output: $S \in \mathbb{R}^{n \times n}$ *with exact* c *connected components, projection* $W \in \mathbb{R}^{d \times m}$.

1. *Initialize* S *by the optimal solution to the problem (2.1).*

2. *Update* F, *which is formed by the* c *eigenvectors of* $L_S = D_S - \frac{S^T + S}{2}$ *corresponding to the* c *smallest eigenvalues.*

3. *Update* W, *whose columns are the* m *eigenvectors of* $S_t^{-1} X^T L_S X$ *corresponding to the* m *smallest eigenvalues.*

4. *For each* i, *update the* i*-th row of* S *by solving the problem similar to (1.4).*

5. *Iteratively perform 2-4 until converges.*

Connection to Linear Discriminant Analysis (LDA)

Similarly, we have the following theorem to connect PCAN with the unsupervised version of LDA.

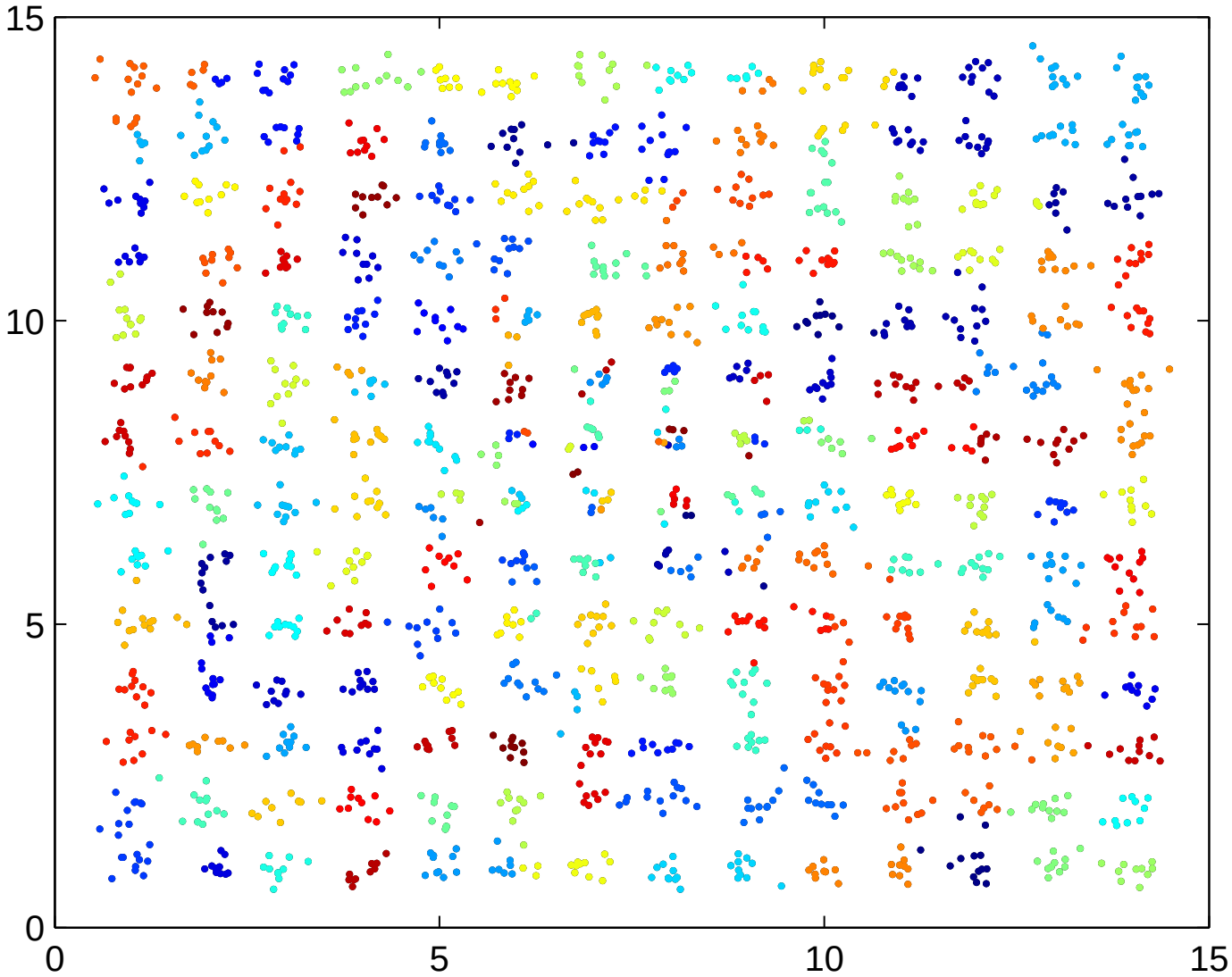
Theorem

When $\gamma \to \infty$, the problem (3.1) is equivalent to the problem of LDA, in which the labels are also variables to be optimized.

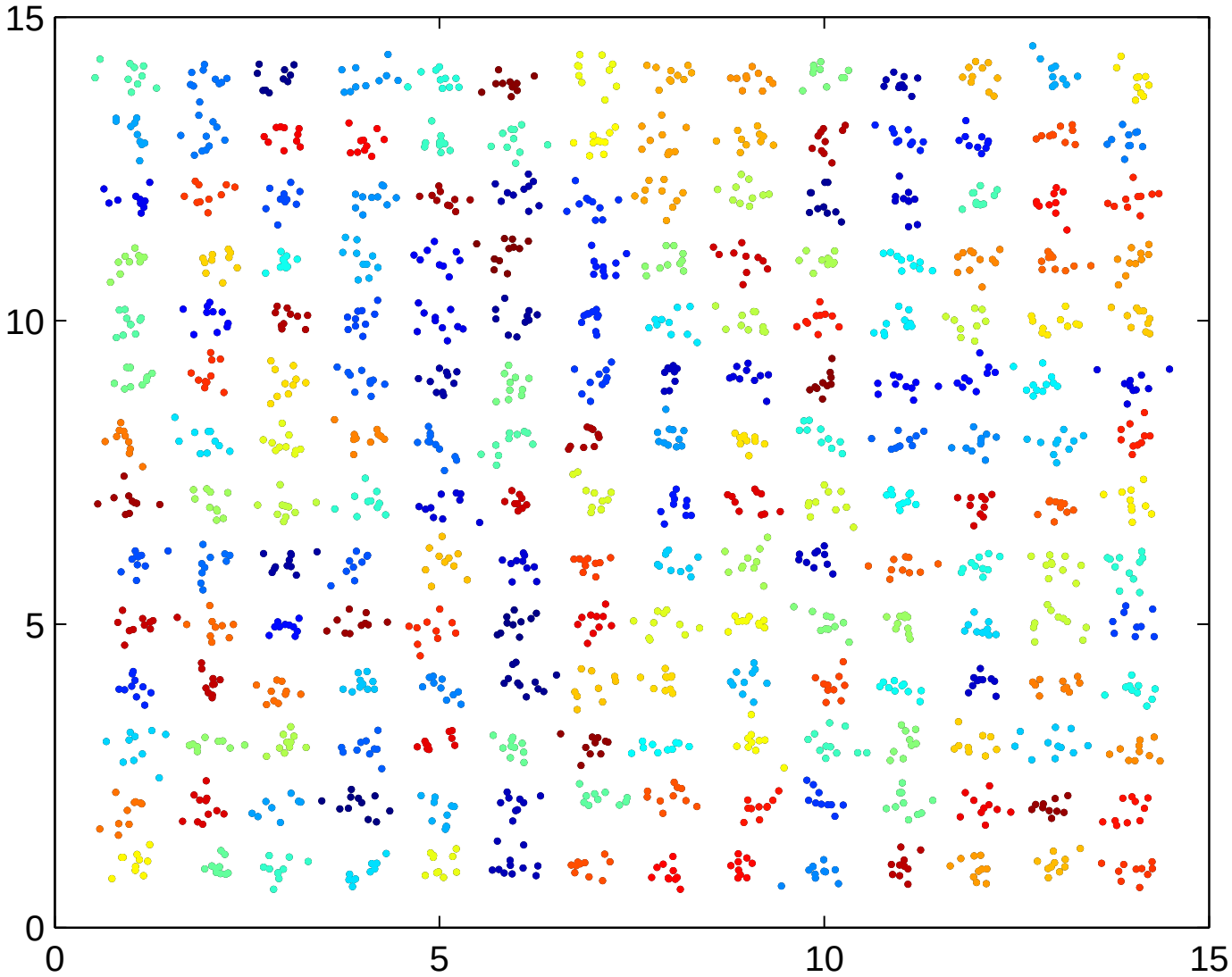
Discussions on PCAN Clustering Algorithm

- The theorem reveals that the proposed PCAN algorithm is to solve the unsupervised version of LDA problem when γ is very large.
- When γ is not very large, the PCAN algorithm can be viewed as a unsupervised version of **local** LDA, which are designed for handling multimodal non-Gaussian data.
- PCAN algorithm performs subspace learning and clustering simultaneously. It can be used as a clustering method, and can also be used as a unsupervised subspace learning (dimensionality reduction) method.

K-Means Result on 196 Clusters



CAN Result on 196 Clusters

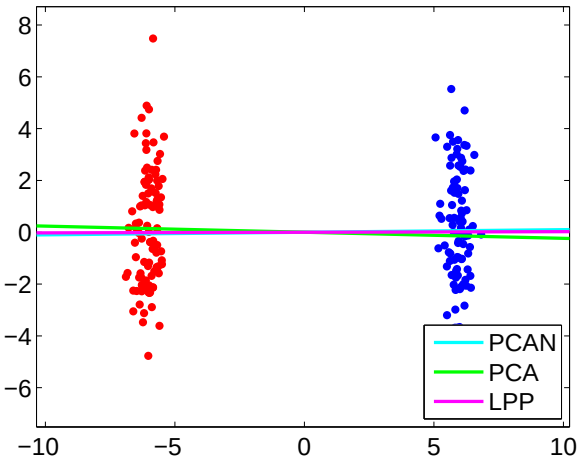


Numerical Results on 196 Clusters

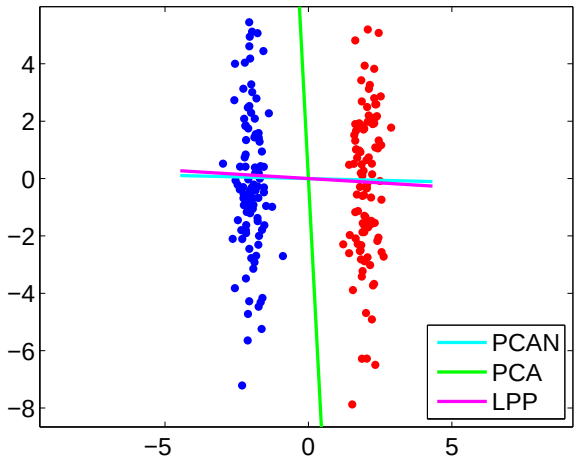
Methods	Acc%(min_obj)	Min_obj
K-Means	69.95	318.29
CAN	**98.98**	**106.12**

Table: Clustering accuracy and minimal K-Means objective value on 196-cluster synthetic data sets.

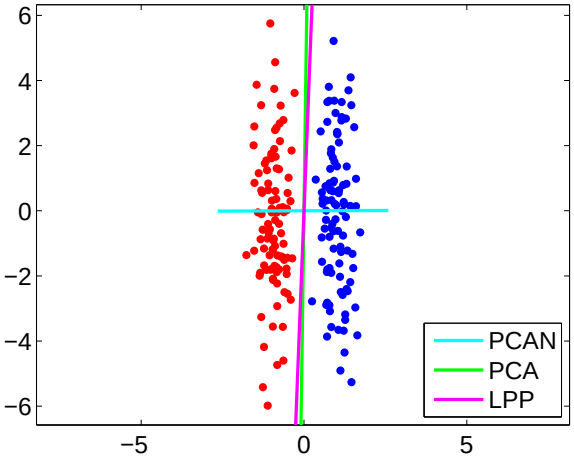
Projection Results of PCAN on Synthetic Data



(a) Clusters far away

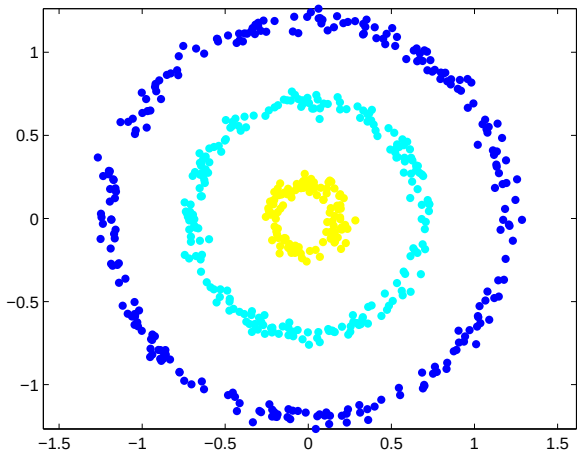


(b) Clusters relatively close

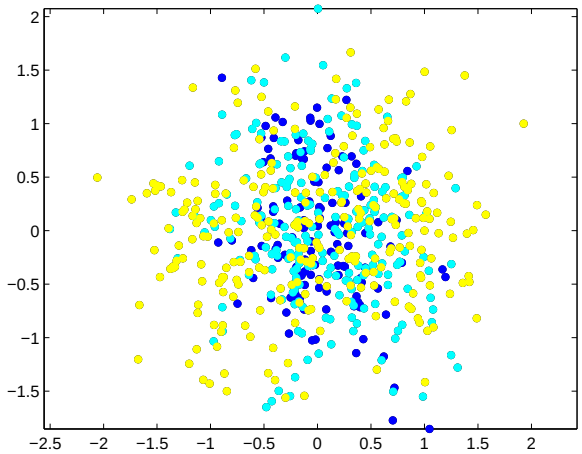


(c) Clusters fairly close

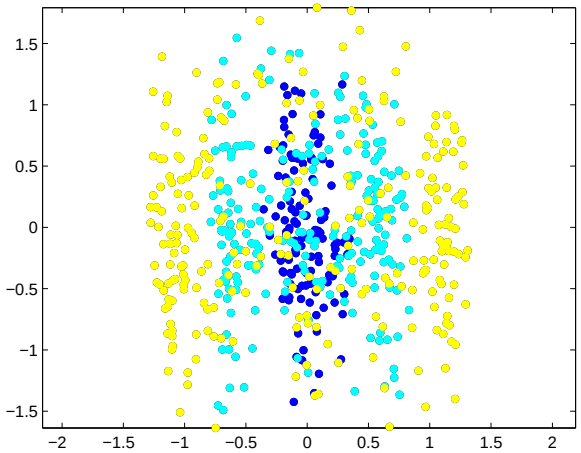
Projection Results of PCAN on Synthetic Data



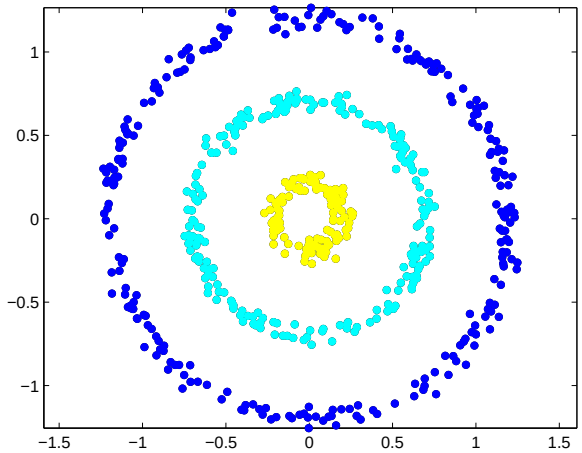
(a) The First Two Dimensions



(b) Learnt Subspace by PCA



(c) Learnt Subspace by LPP



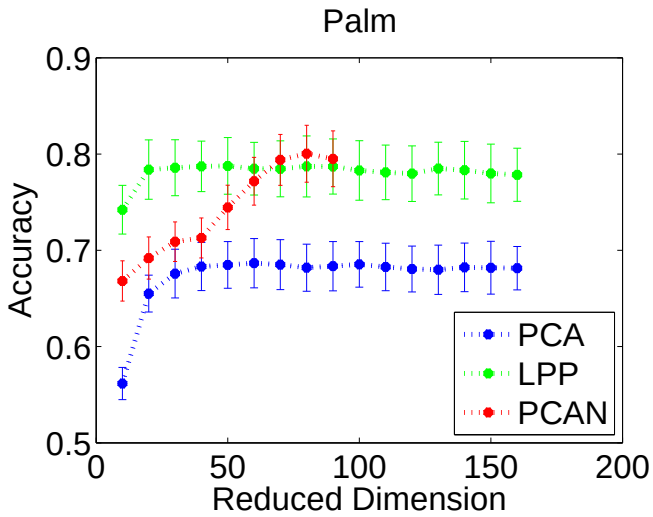
(d) Learnt Subspace by PCAN

Clustering Results on Real Benchmark Datasets

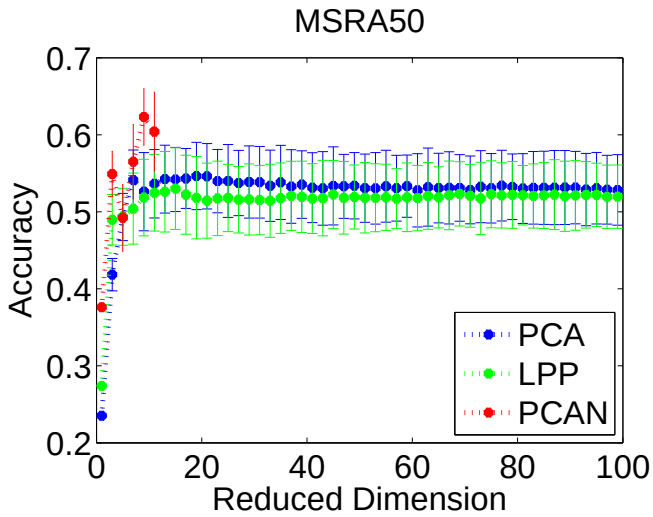
	K-Means		Ratio Cut		Normalized Cut		NMF	CAN	PCAN
	min_obj	Ave	min_obj	Ave	min_obj	Ave			
Umist	42.61	43.31	66.09	60.44	62.26	59.51	62.26	**77.57**	68.17
Coil20	71.67	56.54	77.50	69.42	79.31	70.30	70.42	**90.14**	83.33
Jaffe50	91.55	73.70	96.71	84.78	96.71	81.63	96.71	96.71	**100.00**
USPS	65.21	64.27	69.20	67.59	69.53	68.43	67.37	**78.96**	63.81
Palm	72.40	68.43	65.00	61.35	63.50	60.97	61.40	84.85	**88.85**
MSRA50	57.14	53.50	52.47	49.28	52.47	47.90	50.69	**57.87**	**57.87**
Stock	66.74	74.02	57.68	55.23	57.68	55.39	56.21	67.79	**77.16**
Pathbased	74.33	74.34	77.67	77.67	77.67	77.67	78.00	**87.00**	**87.00**
Movements	48.33	44.04	48.33	46.01	45.56	44.49	43.33	**49.17**	**49.17**
Spiral	34.29	34.56	99.68	96.92	99.68	96.03	91.03	**100.00**	**100.00**
Wine	95.51	94.65	95.51	95.44	94.94	94.99	94.94	97.19	**100.00**
Compound	69.42	65.49	53.63	52.94	53.13	52.67	52.38	**80.20**	79.70
Yeast	40.50	38.00	41.44	38.11	40.03	36.99	35.65	**50.27**	50.07
Glass	43.46	45.57	37.85	38.28	37.85	38.26	37.85	**50.00**	49.53
Ecoli	62.50	57.10	59.23	54.08	57.44	53.10	54.17	83.04	**83.33**

Table: Clustering Accuracy (%) on Real Data Sets

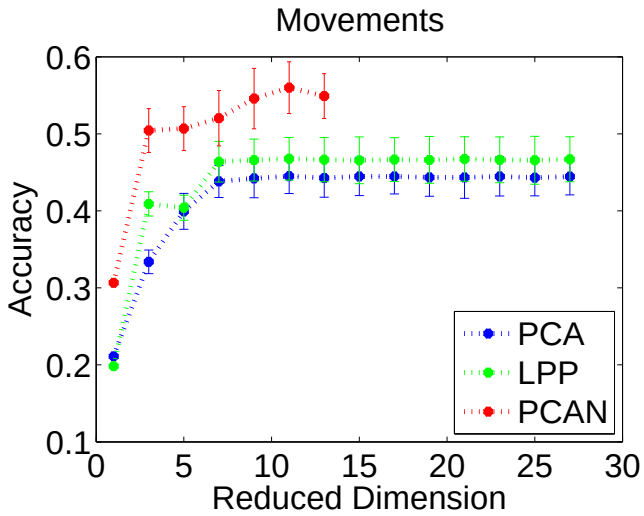
Results of PCAN on Real Benchmark Datasets



(a)

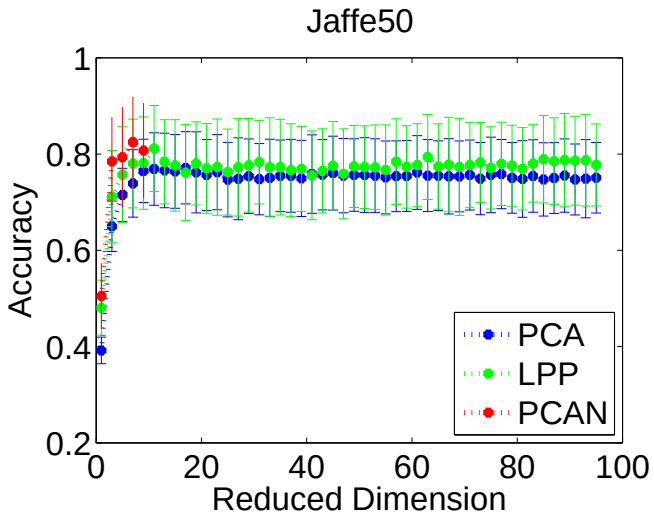


(b)

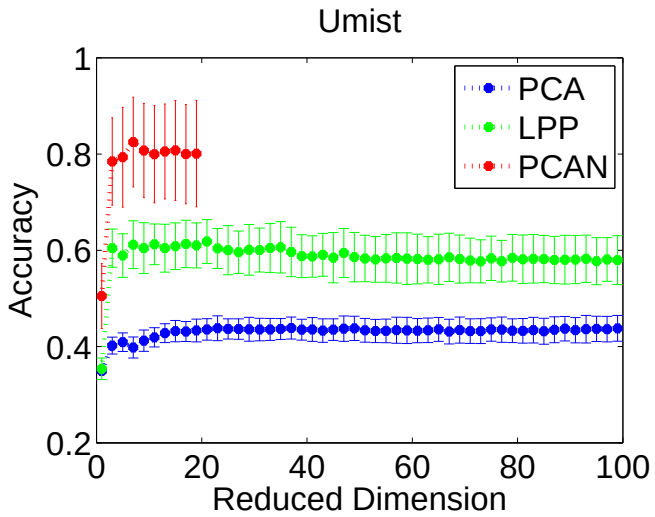


(c)

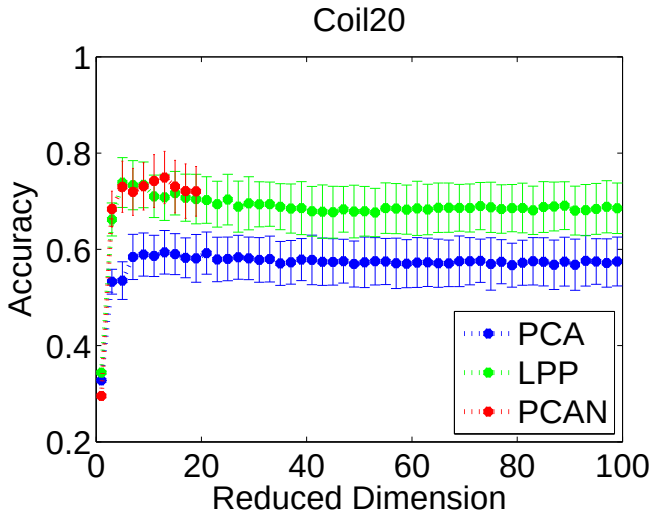
Results of PCAN on Real Benchmark Datasets



(a)



(b)



(c)

Results of PCAN on Real Benchmark Datasets

	PCA	LPP	PCAN
Palm	68.66±2.55	78.77±2.94	**80.03±2.96**
MSRA50	54.65±4.37	52.93±4.31	**62.31±3.68**
Movements	44.46±2.48	47.47±2.55	**55.99±3.06**
Jaffe50	77.45±7.30	79.41±9.61	**81.78±10.15**
Umist	43.85±2.28	64.55±4.66	**67.39±4.53**
Coil20	59.39±4.63	73.32±5.10	**74.93±5.43**

Table: The best results with optimal dimensions.

Conclusions

- We proposed a CAN clustering algorithm with adaptive neighbors, the learned similarity matrix can be directly used for clustering, without having to perform K-means or other discretization procedures.
- Theoretical analysis reveals the proposed CAN clustering algorithm is connected with the K-means clustering problem, and the CAN can achieve much better clustering results than traditional K-means algorithm does.
- For the high-dimensional clustering problem, we propose a Projected CAN (PCAN) algorithm, which performs clustering and dimensionality reduction simultaneously.
- Theoretical analysis reveals the proposed PCAN clustering algorithm is connected with unsupervised LDA, and the PCAN can achieve better clustering or dimensionality reduction results than previous clustering algorithms or unsupervised dimensionality reduction algorithms do.

Acknowledgement

- UT Arlington Team: Feiping Nie, Xiaoqian Wang, Heng Huang

- Thanks to NSF support:
  NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152,
  NSF-DBI 1356628

- Thanks to all audience

- Questions?