Advanced Machine Learning & Perception

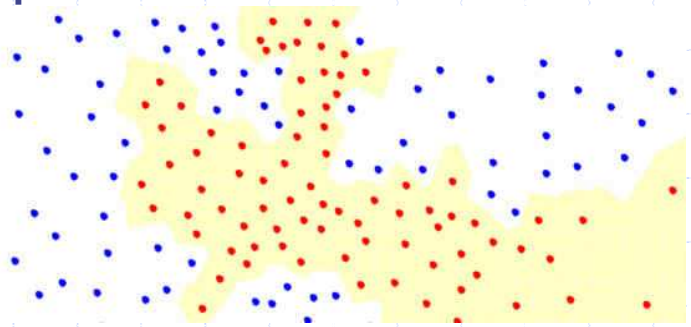
Instructor: Tony Jebara

Topic 12

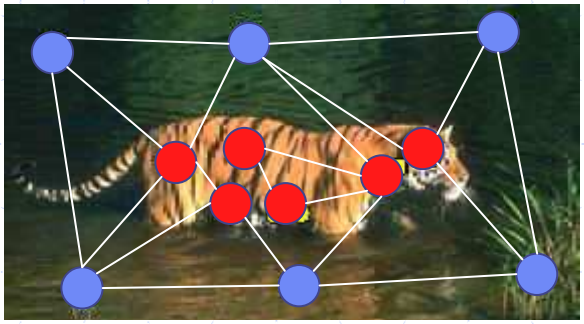
- Graphs in Machine Learning
- Graph Min Cut, Ratio Cut, Normalized Cut
- Spectral Clustering
- Stability and Eigengap
- Matching, B-Matching and k-regular graphs
- B-Matching for Spectral Clustering
- B-Matching for Embedding

Graphs in Machine Learning

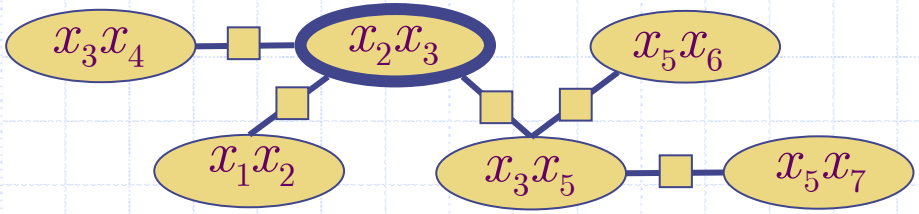
- Many learning scenarios use graphs



- Classification:
  k-nearest neighbors

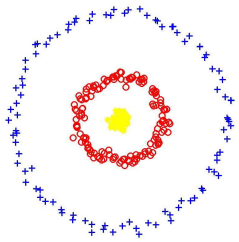


- Clustering:
  normalized cut
  spectral clustering



- Inference:
  Bayesian networks
  belief propagation

Normalized Cut Clustering



- Better than: kmeans, EM, linkage, etc.
- No local minima or parametric assumptions

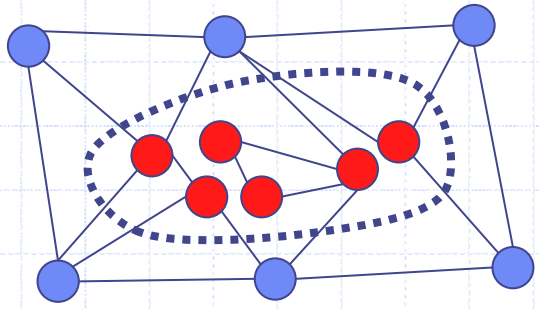
- Given graph (V,E) with weight matrix A, normalized cut is

$$ncut(B) = \frac{\sum_{i \in B, j \in V/B} A_{ij}}{\sum_{i \in B, j \in V} A_{ij}} + \frac{\sum_{i \in V/B, j \in B} A_{ij}}{\sum_{i \in V/B, j \in V} A_{ij}}$$

we could fill in A using pairwise similarities/kernels

$$A_{ij} = k\left(x_i, x_j\right)$$

- But, this is a hard problem need a relaxation…

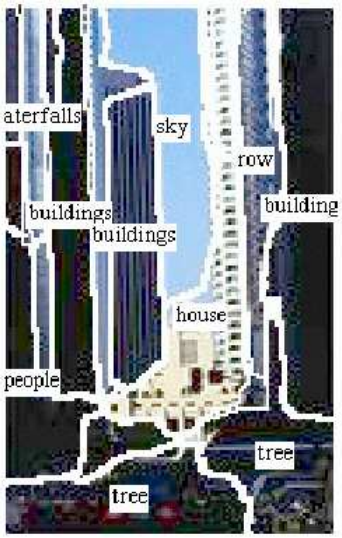


Spectral Clustering

- Typically, use EM or k-means to cluster N data points
- Can imagine clustering the data points only from NxN matrix capturing their proximity information
- This is spectral clustering
- Again compute Gram matrix using, e.g. RBF kernel

$$A_{ij} = k\left(x_i, x_j\right) = \phi\left(x_i\right)^T \phi\left(x_j\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|x_i - x_j\right\|^2\right)$$

- Example: have N pixels from an image, each x = [xcoord, ycoord, intensity] of each pixel
- From eigenvectors of K matrix (or slight, variant), these seem to capture some segmentation or clustering of data points!
- Nonparametric form of clustering since we didn't assume Gaussian distribution...



Spectral Clustering

- Convert data to graph & cut
- Given graph (V,E), weight matrix A, best normalized cut B* is NP

$$ncut(B) = \frac{\sum_{i \in B, j \in V/B} A_{ij}}{\sum_{i \in B, j \in V} A_{ij}} + \frac{\sum_{i \in V/B, j \in B} A_{ij}}{\sum_{i \in V/B, j \in V} A_{ij}}$$

- Define:
  - diagonal degree matrix $D_{ii} = \sum_j A_{ij}$
  - volume $d = \sum_i D_{ii}$
  - volume of cut B $d_B = \sum_{i \in B} D_{ii}$
  - unnormalized Laplacian $L = D - A$
- Solve (combinatorial, NP):

$$\min_y \frac{y^T L y}{y^T D y} \qquad such\ that\ y^T D\vec{1} = 0\ and\ y(i) = \{1, -b\}$$

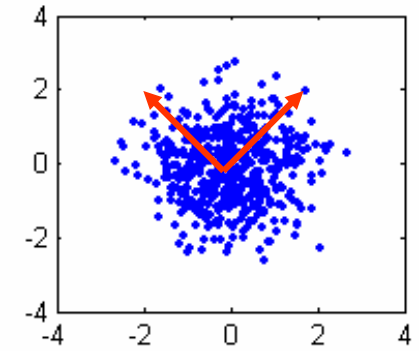
- Relax to continuous y (Shi & Malik):

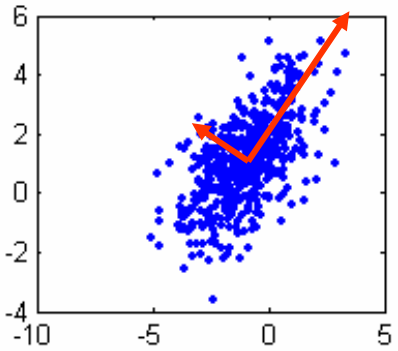
$$\min_y y^T L y \quad such\ that\ y^T D y = 1 \quad and \quad y^T D\vec{1} = 0$$

- Solve for y as 2nd smallest eigenvector of: $(D - A)y = \lambda D y$

Stability in Spectral Clustering

- Standard problem when computing & using eigenvectors:

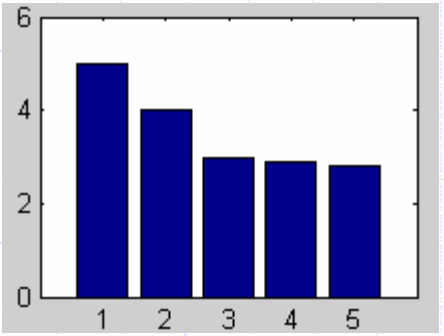




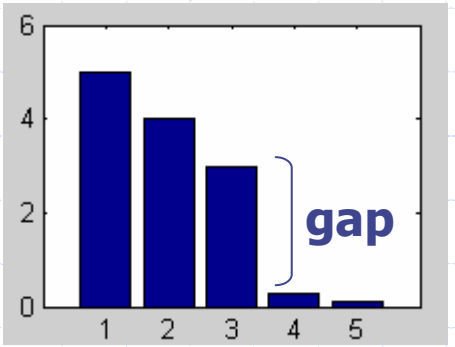
- Small changes in data can cause eigenvectors to change wildly
- Ensure the eigenvectors we keep are distinct & stable: look at eigengap…
- Some algorithms ensure the eigenvectors are going to have a safe eigengap.

Use normalized Laplacian: $L = D^{-1/2} A D^{-1/2}$

3 evecs=unsafe



3 evecs=safe



Stabilized Spectral Clustering

- Stabilized spectral clustering algorithm:

Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-||s_i - s_j||^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Stabilized Spectral Clustering

- Example results compared to other clustering algorithms (traditional kmeans, unstable spectral clustering, connected components).

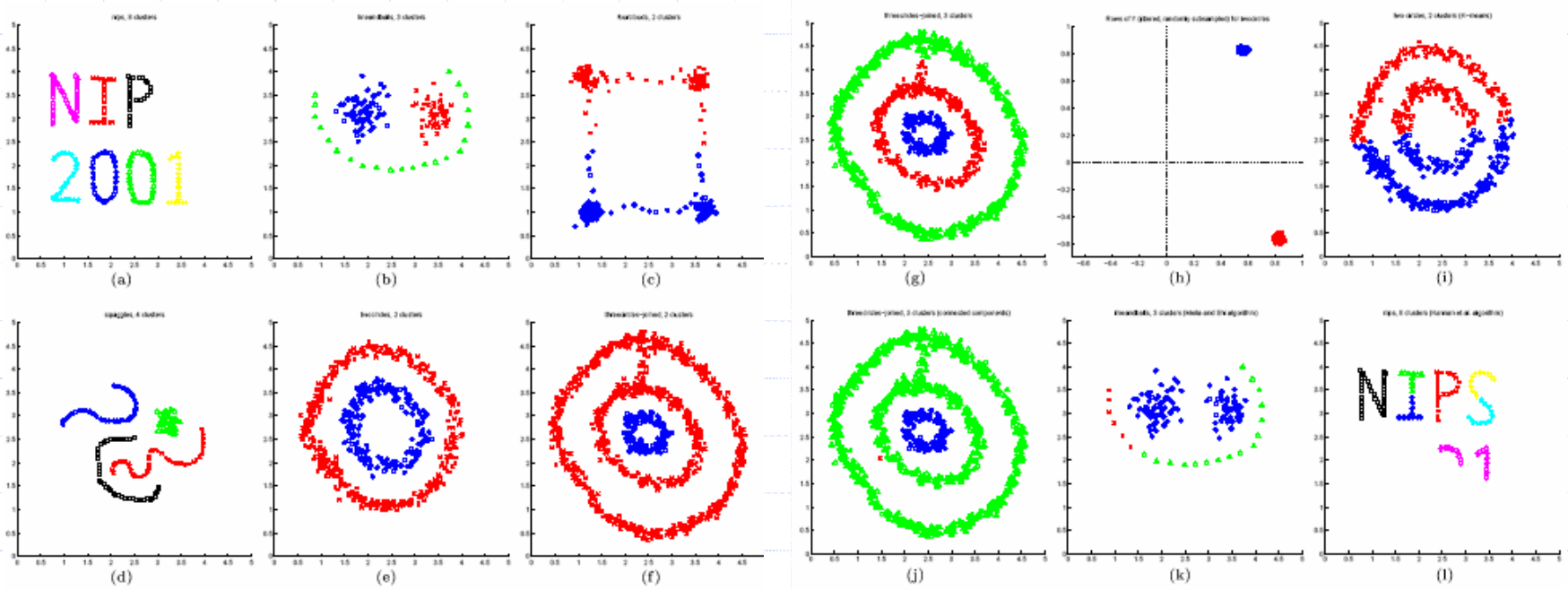


Figure 1: Clustering examples, with clusters indicated by different symbols (and colors, where available). (a-g) Results from our algorithm, where the only parameter varied across runs was k. (h) Rows of Y (jittered, subsampled) for `twocircles` dataset. (i) K-means. (j) A "connected components" algorithm. (k) Meila and Shi algorithm. (l) Kannan et al. Spectral Algorithm I. (See text.)

Matching and B-Matching

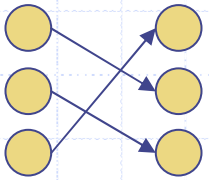
- Matching (or perfect matching) = permutation = assignment
- Maximum Weight Matching = Linear Assignment Problem
  Given weight matrix, find permutation matrix. $O(N^3)$

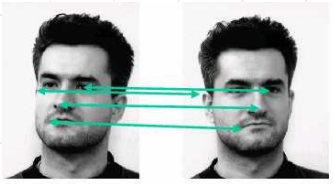
$$\text{wife} \begin{bmatrix} \$1 & \$6 & \$3 \\ \$4 & \$2 & \$4 \\ \$4 & \$2 & \$5 \end{bmatrix} \rightarrow \text{Kuhn-Munkres Hungarian Algorithm} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

husband

$$\max_P tr\left(P^T A\right)$$

$$\sum_i P_{ij} = \sum_j P_{ij} = 1,\ P_{ij} \in \{0,1\}$$

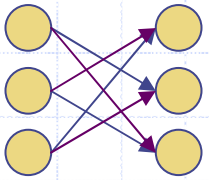




- B-Matching generalizes to multi-matchings (Mormon). $O(bN^3)$

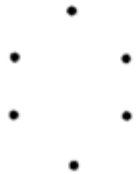
$$\max_P tr\left(P^T A\right)$$

$$\sum_i P_{ij} = \sum_j P_{ij} = b,\ P_{ij} \in \{0,1\}$$

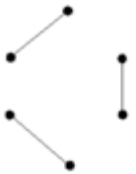


Matching and B-Matching

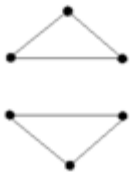
- Multi-matchings or b-matchings are also known as k-regular graphs (as opposed to k-nearest neighbor graphs)



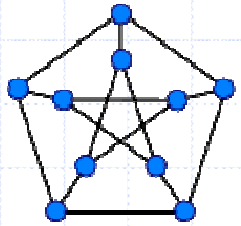
0-regular



1-regular



2-regular



3-regular

Matching and B-Matching

- Balanced versions of k-nearest neighbor

$$A = \begin{bmatrix} 27 & 89 & 6 & 43 & 21 & 79 \\ 25 & 20 & 99 & 23 & 38 & 6 \\ 88 & 30 & 58 & 58 & 78 & 60 \\ 74 & 66 & 42 & 76 & 68 & 5 \\ 14 & 28 & 52 & 53 & 46 & 42 \\ 1 & 47 & 33 & 64 & 57 & 30 \end{bmatrix}$$

$$\max_P tr\left(P^T A\right)$$

$$where\, P_{ij} \in \{0,1\}$$

Neighbors $O\left(bN^2\right)$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\sum_j P_{ij} = 1$$

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\sum_j P_{ij} = 3$$

Matchings $O\left(bN^3\right)$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\sum_i P_{ij} = \sum_j P_{ij} = 1$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\sum_i P_{ij} = \sum_j P_{ij} = 3$$

B-Matched Spectral Clustering

- Try to improve spectral clustering using B-Matching
- Assume w.l.og. two clusters of roughly equal size
- If we knew the labeling y = [+1 +1 +1 -1 -1 -1]
  the "ideal" affinity matrix A = $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$
  in other words...

$$A = \tfrac{1}{2}\left(yy^T + 1\right)$$

  and spectral clustering and eigendecomposition is perfect
- The "ideal" affinity is a B-Matching with b=N/2
- Stabilize affinity by finding the closest B-Matching to it:

$$\min_P \|A - P\|^2 \ such\,that \sum_i P_{ij} = \sum_j P_{ij} = \tfrac{N}{2} \ and \ P_{ij} \in \{0,1\}$$

- Then, spectral cluster B-Matching or use it to prune A

$$A^{new} = P \quad or \quad A^{new} = P \circ A$$

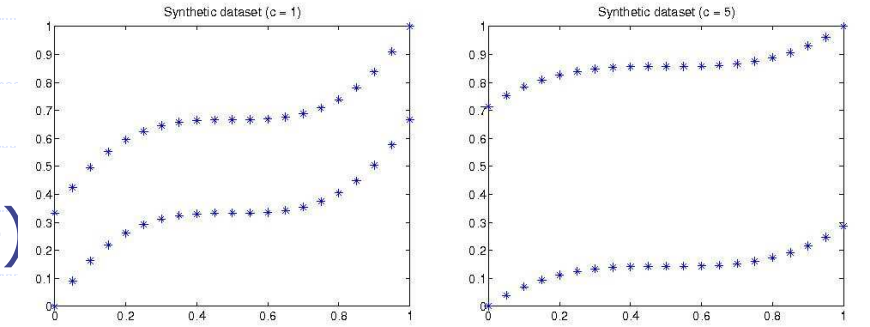
- Also, instead of B-Matching, can do kNN (lazy strawman).

B-Matched Spectral Clustering

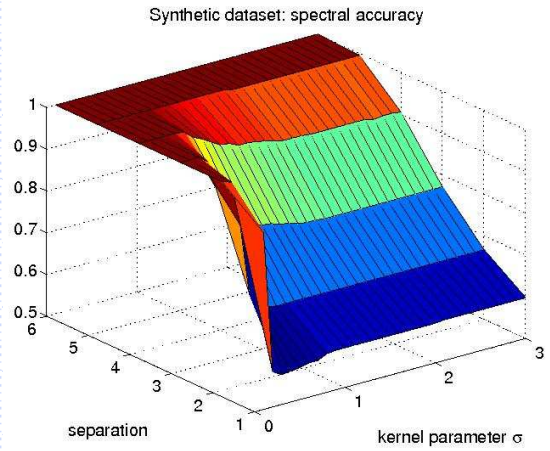
- Synthetic experiment
- Have 2 S-shaped clusters
- Explore different spreads
- Affinity $A_{ij}=\exp(-||X_i-X_j||^2/\sigma^2)$
- Do spectral clustering on

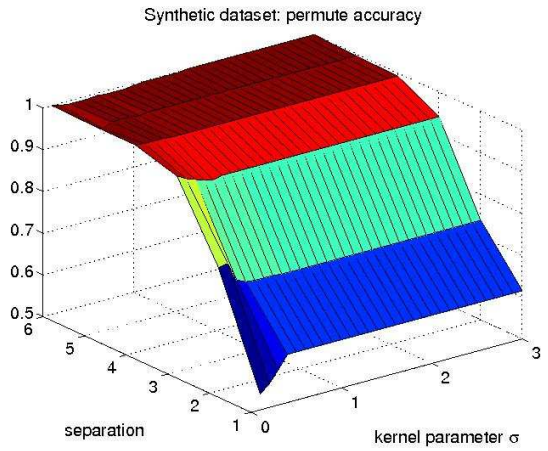
$$A \quad or \quad P \quad or \quad P \circ A$$

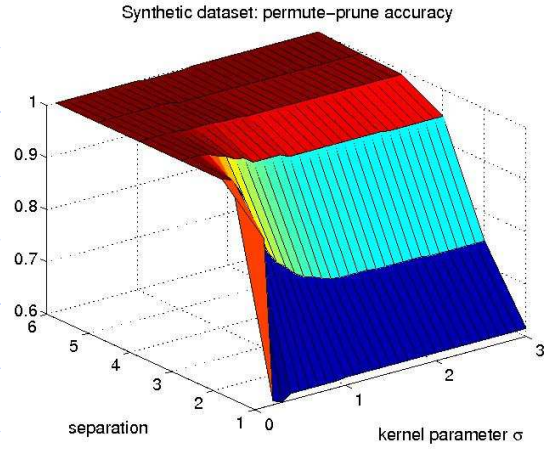
- Evaluate cluster labeling accuracy



(a) Curve separation $c = 1$. (b) Curve separation $c = 5$.

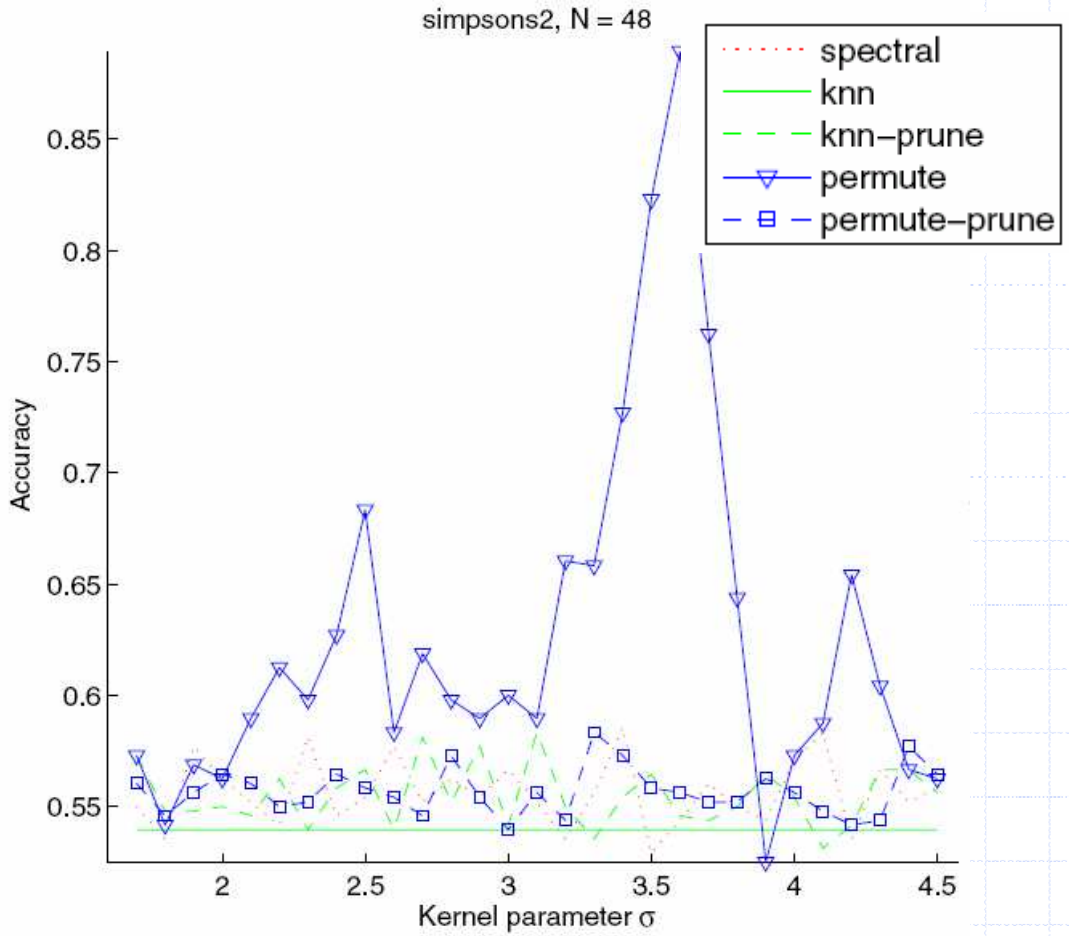






B-Matched Spectral Clustering

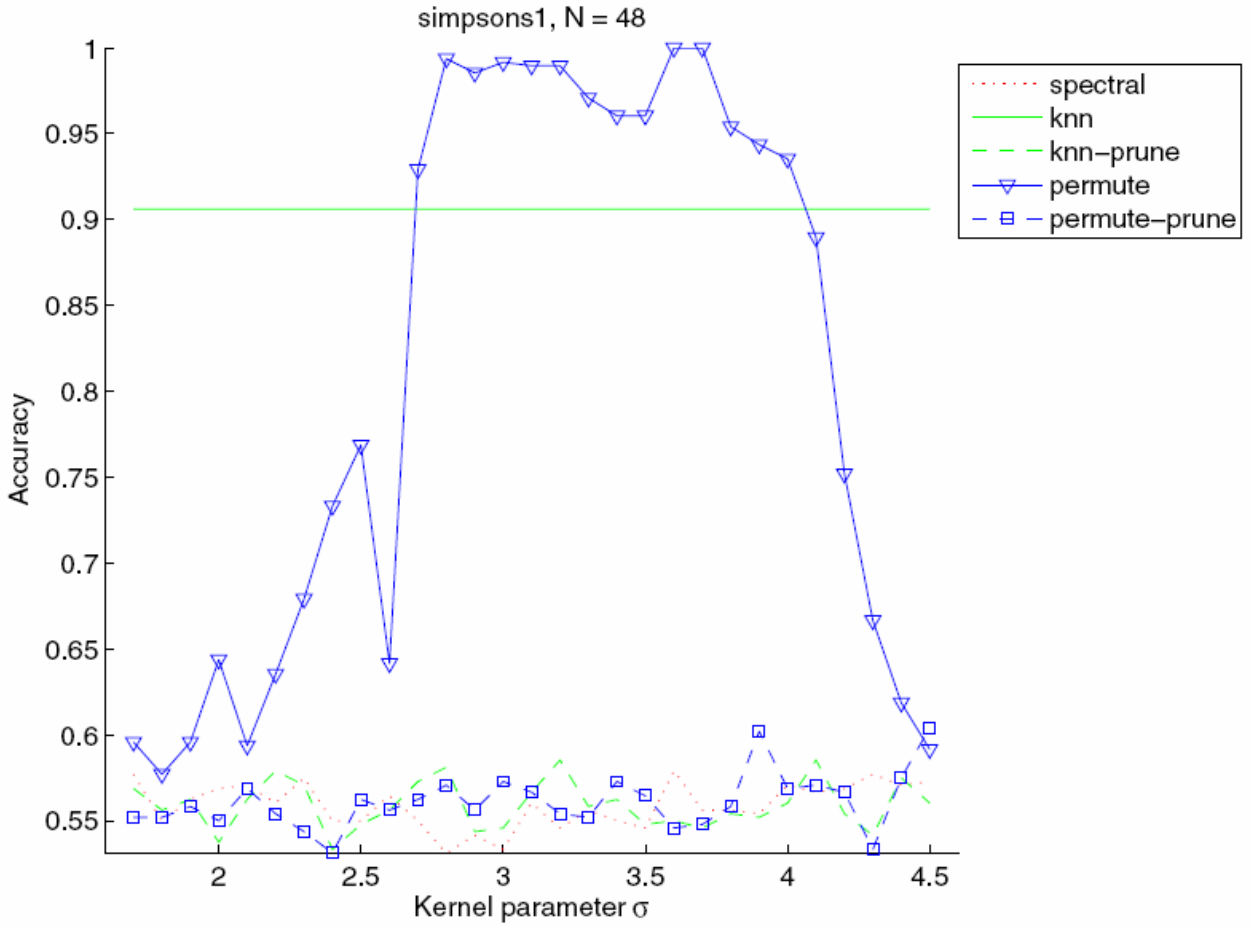
- Clustering images from real video with 2 scenes in it.
- Accuracy is how well we classify both scenes (10-fold)
- Evaluated also with kNN
- Only adjacent frames have high affinity
- BMatching does best since it boosts connection to far frames



(a) Maggie vs. Marge Scene

B-Matched Spectral Clustering

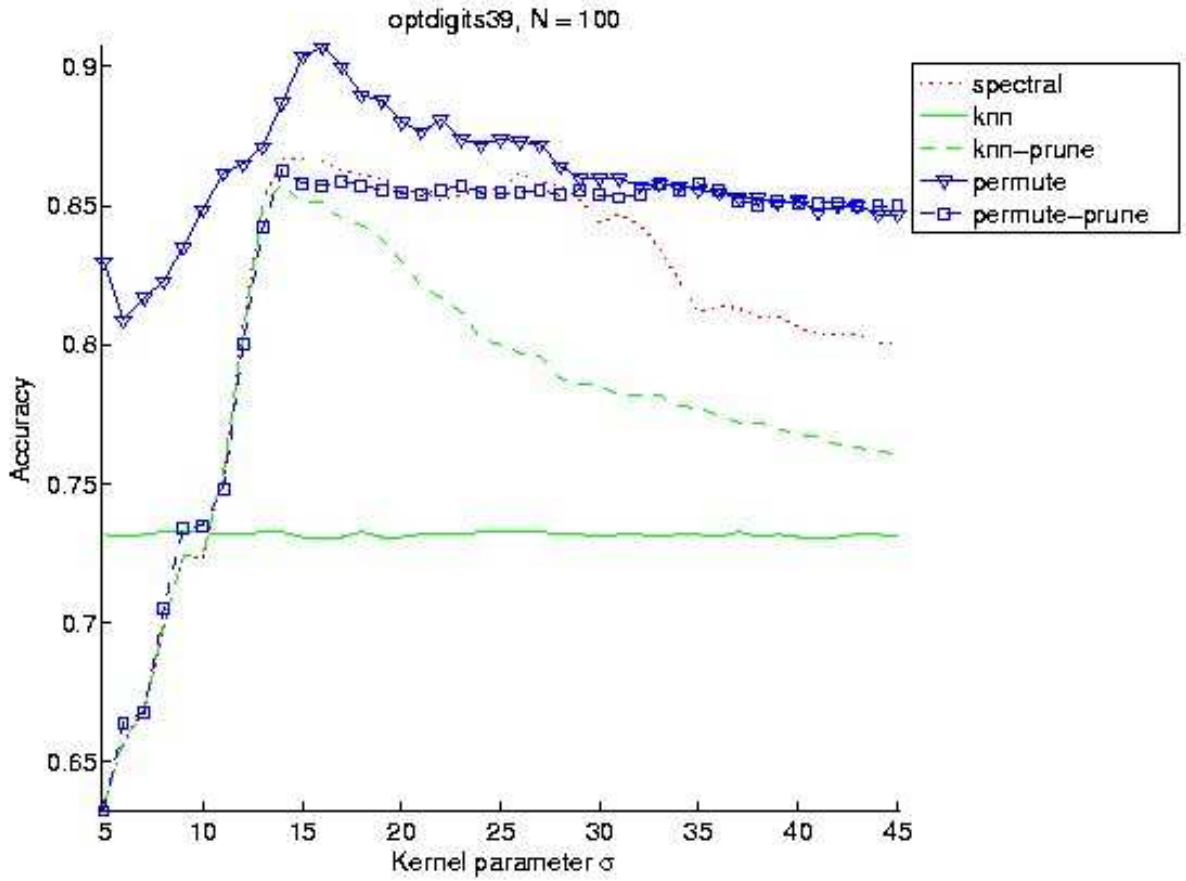
- Clustering images from same video but 2 other scenes



(b) Homer vs. Bart Scene

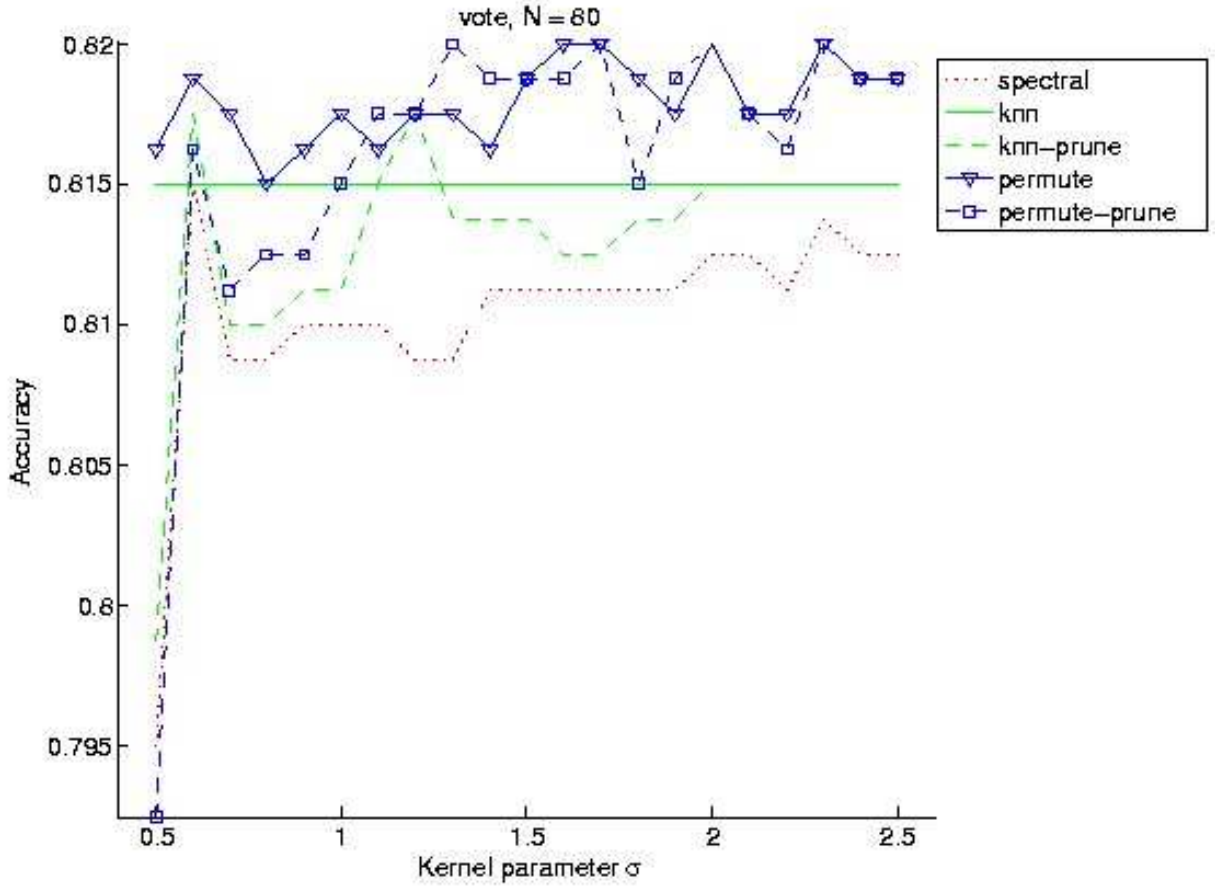
B-Matched Spectral Clustering

- Unlabeled classification via clustering of UCI Optdigits data



B-Matched Spectral Clustering

- Unlabeled classification via clustering of UCI Vote dataset

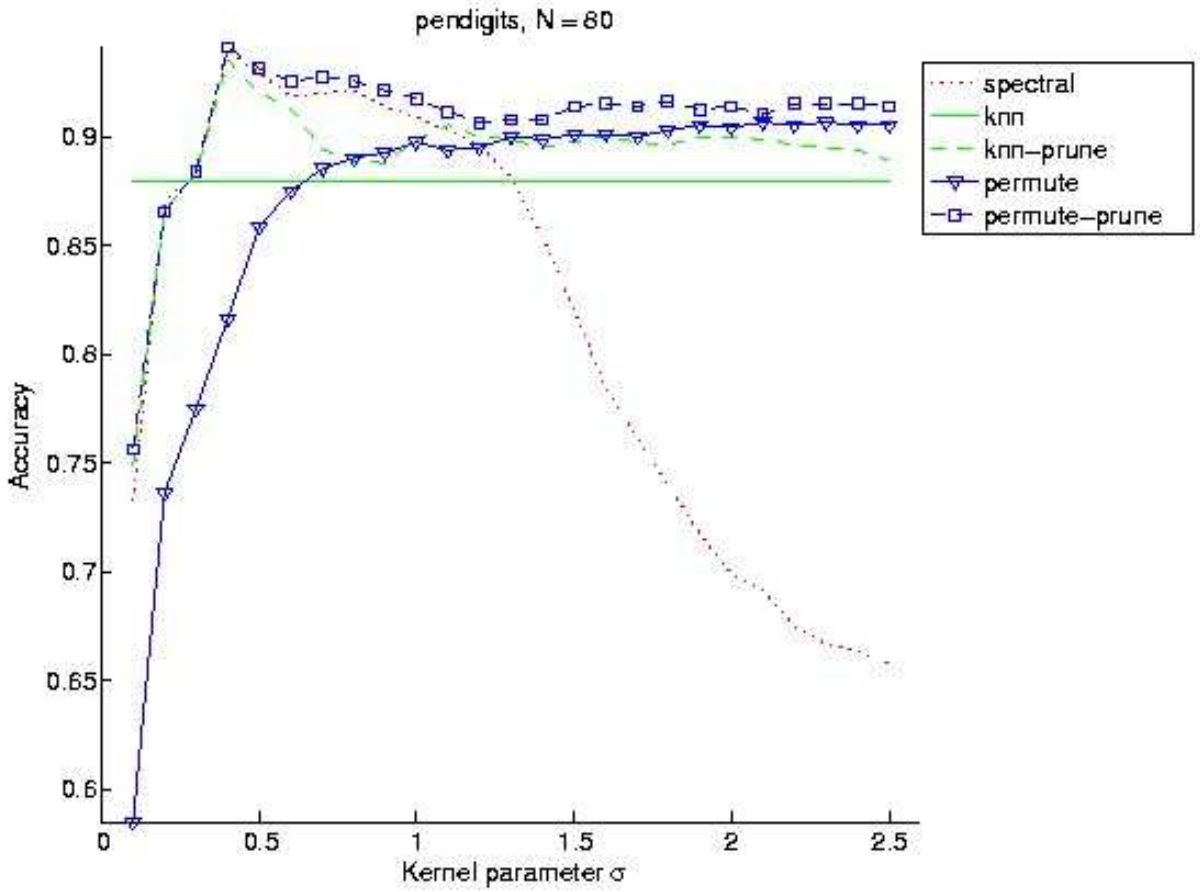


B-Matched Spectral Clustering

- Classification accuracy via clustering of UCI Pendigits data

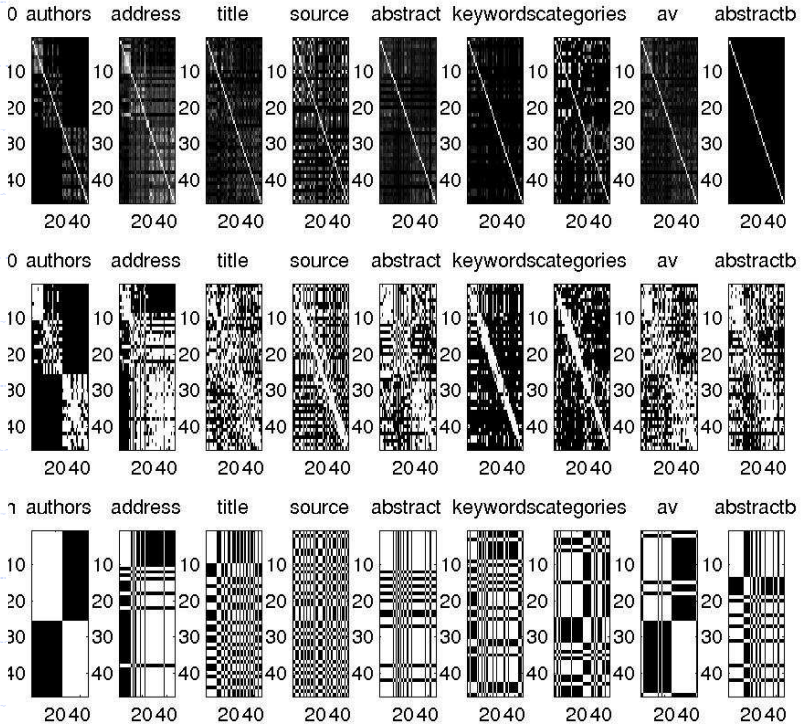
- Here, using the B-Matching to just prune A is better

  kNN always seems a little worse…



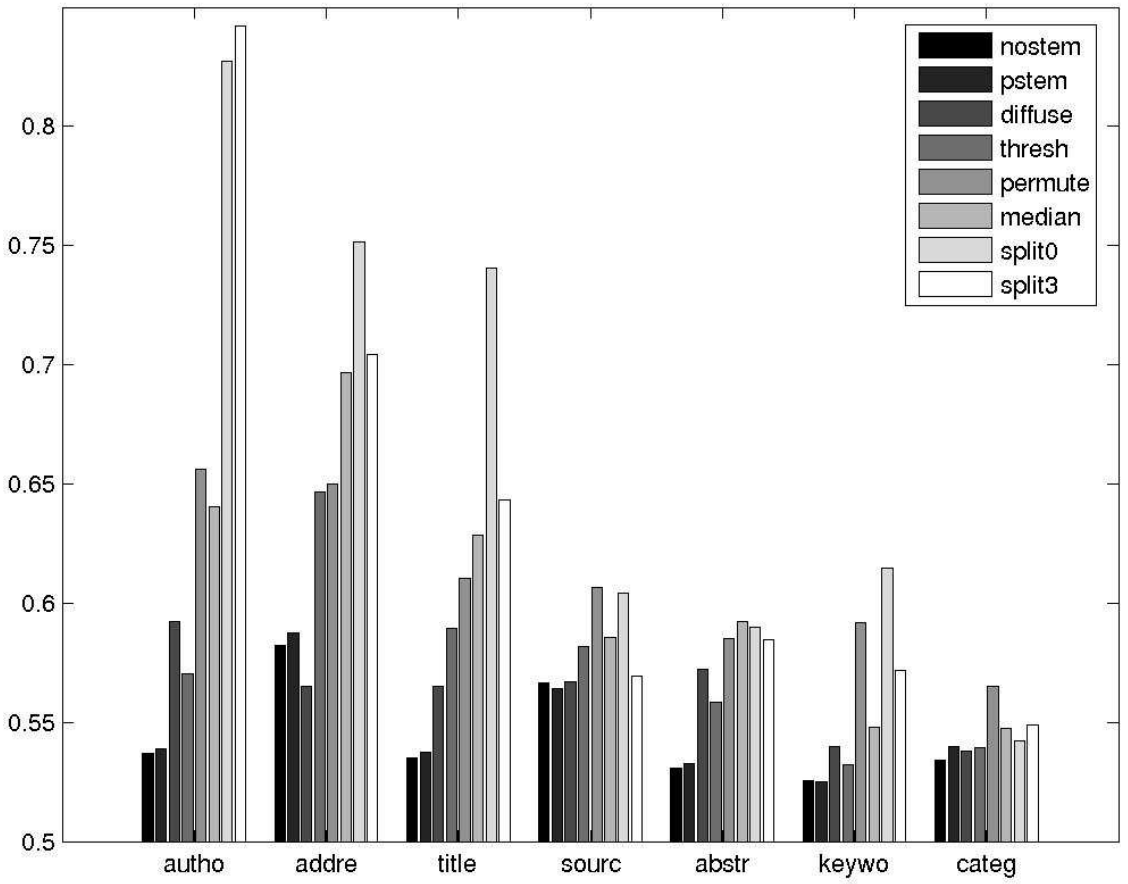
B-Matched Spectral

- Applied method to KDD 2005 Challenge. Predict authorship in anonymized BioBase pubs database
- Challenge gives many splits of N~100 documents
- For each split, find NxN matrix saying if documents i & j were authored by same person (1) or different person (0)
- Documents have 8 fields
- Compute affinity A for each field via text frequency kernel
- Find B-Matching P
- Get spectral clustering y and compute ½ (yy^T+1)



B-Matched Spectral Clustering

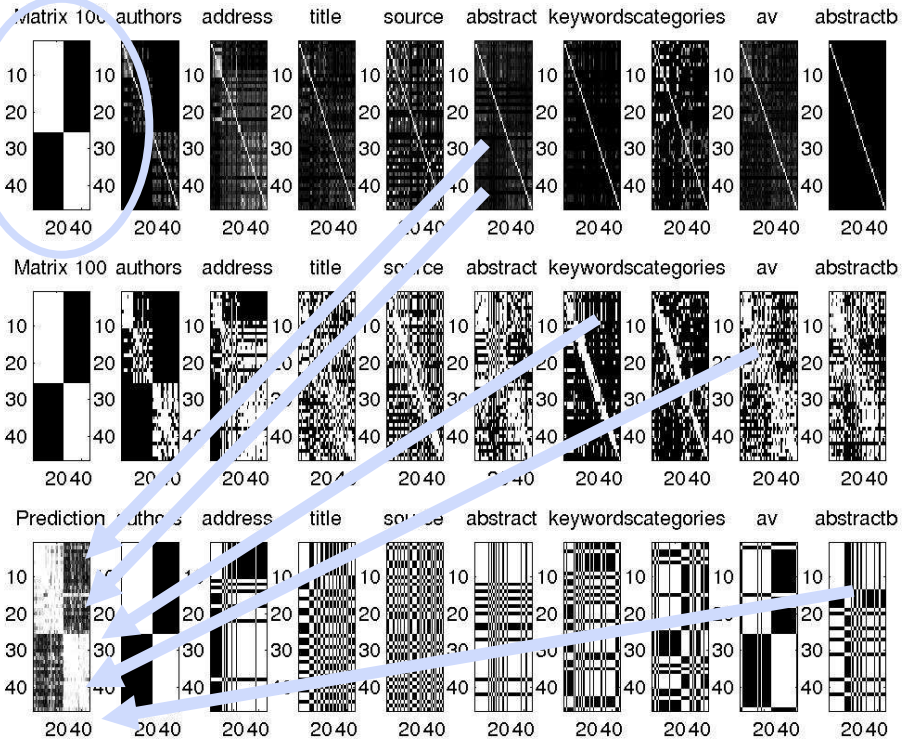
- Accuracy is evaluated using the labeled true same-author & not-same-author matrices
- Explored many processings of the A matrices. Best accuracy was by using the spectral clustered values of the P matrix found via B-Matching



B-Matched Spectral Clustering

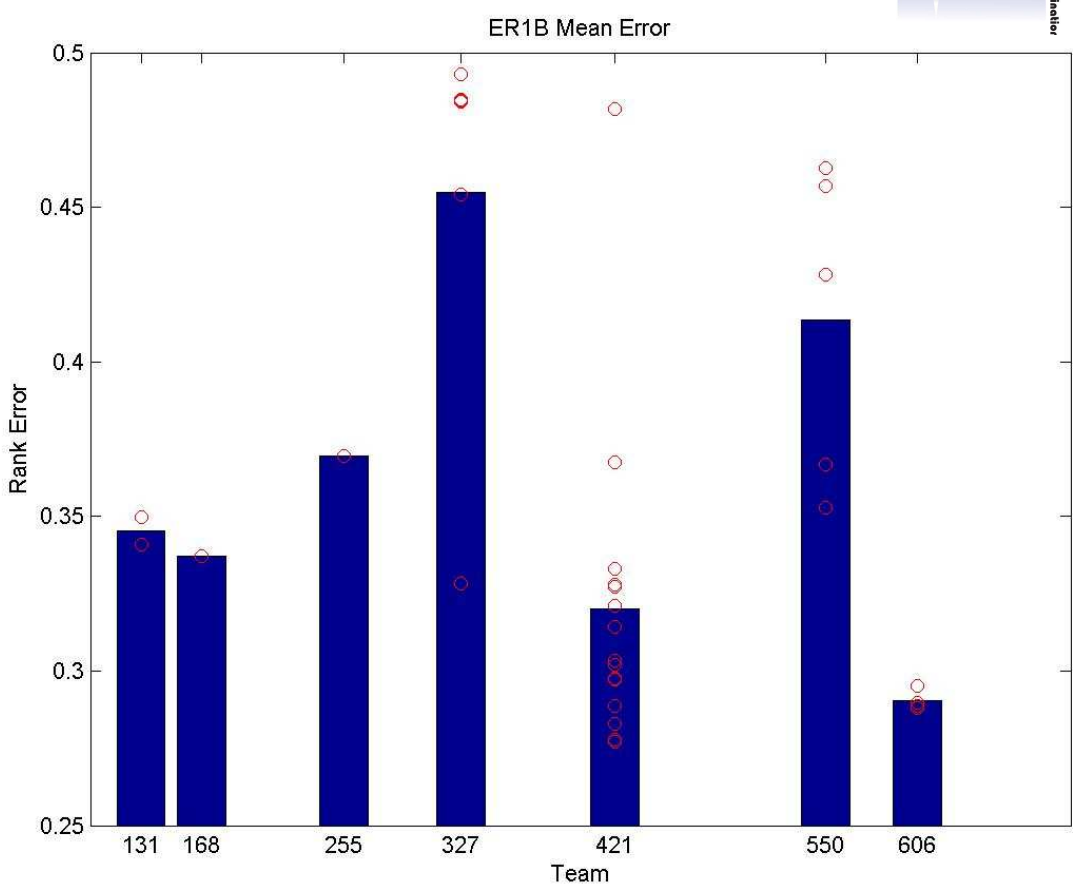
- Merge all the 3x8 matrices into a single hypothesis using an SVM and a quadratic kernel. SVM is trained on labeled data (same author, not same author matrices).

- For each split, we get a single matrix of same-author and not-same-author which was uploaded to KDD Challenge anonymously



B-Matched Spectral Clustering

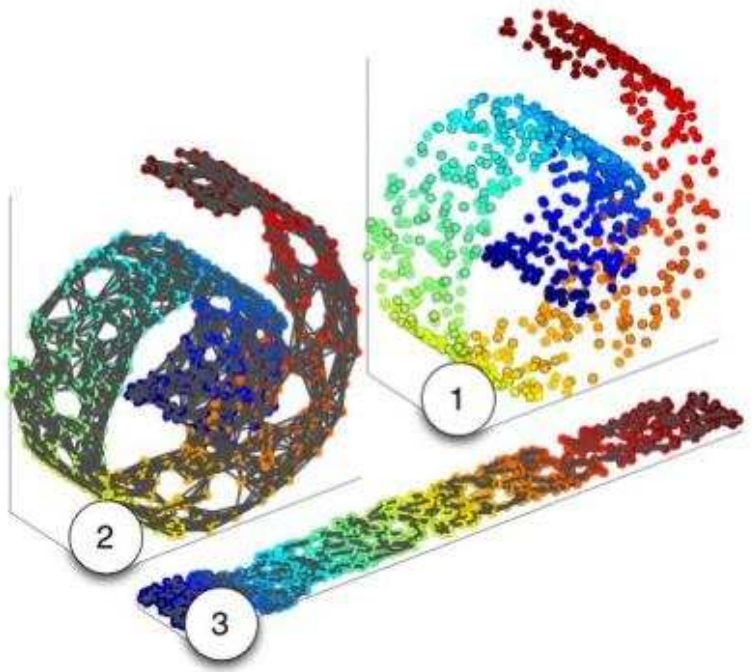
- Total of 7 funded teams attempted this KDD Challenge task and were evaluated by a rank error
- Double-blind submission and evaluation
- Our method had lowest average error



B-Matched Embedding

- Replace k-nearest-neighbor in machine learning algorithms
- Example: Semidefinite Embedding (Weinberger & Saul)

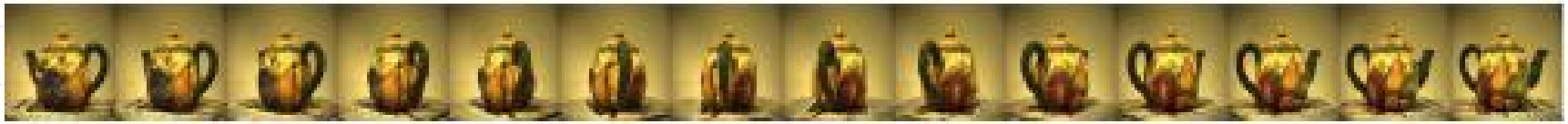
1) Get matrix A by computing affinities between all pairs of points
2) Find k-nearest neighbors graph
3) Use SDP to find P.D. matrix K which preserves distances on graph yet is as stretched as possible. Eigen-decomposition of K finds embedding of points in low dimension that preserve distance structures in high dimensional data.



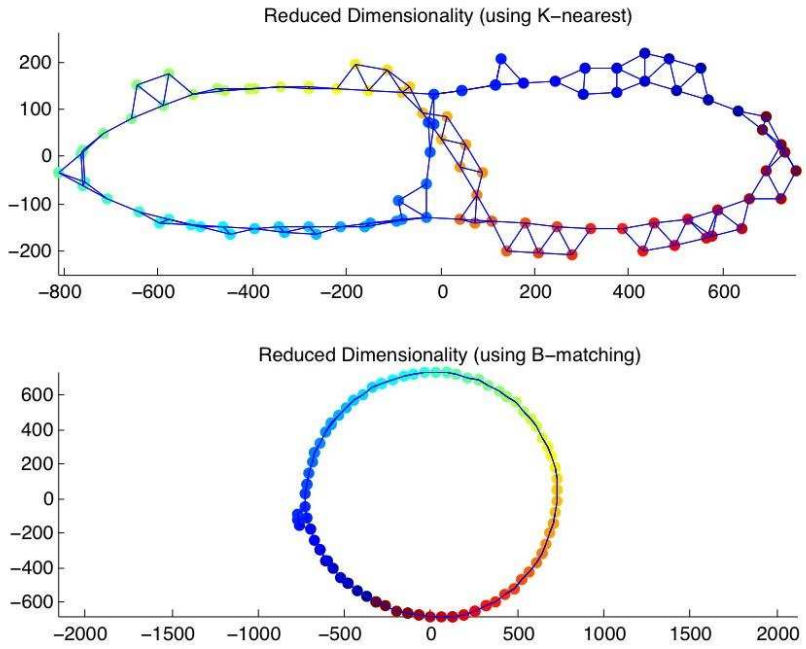
Maximize $\mathrm{Tr}(K)$ subject to $K \succeq 0$, $\sum_{ij} K_{ij} = 0$,
and $\forall$ i,j such that $h_{ij}=1$ or $[h^T h]_{ij}=1$,
$K_{ii} + K_{jj} - K_{ij} - K_{ji} = A_{ii} + A_{jj} - A_{ij} - A_{ji}$.

B-Matched Embedding

- Visualization example: images of rotating tea pot.



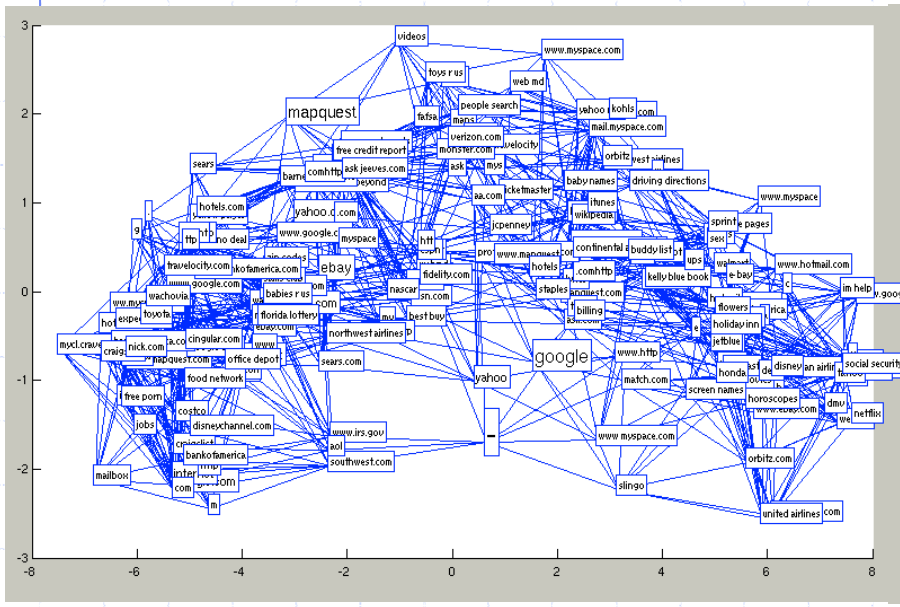
- Get affinity $A_{ij} = \exp(-||X_i - X_j||^2)$ between pairs of images
- Should get ring but noisy images confuse kNN. Greedily connects nearby images without balancing in-degree and out-degree. Get folded over ring even for various values of b (k)
- B-Matching gives clean ring for many values of b.



B-Matched Embedding

- AOL data
- BMatching: Growing initial connectivity using b-matching algorithm instead of k-nearest neighbor

B-matching



K-nearest neighbors