Large-scale Spectral Clustering Methods

for Image and Text Data

Sponsor: Verizon Wireless

Jeffrey Lee*, Scott Li*,
Jiye Ding, Maham Niaz, Khiem Pham, Xin Xu, Zhengxia Yi, Xin Zhang

May 23, 2018

Outline

Background

- Clustering Basics
- Spectral Clustering
- Limitations

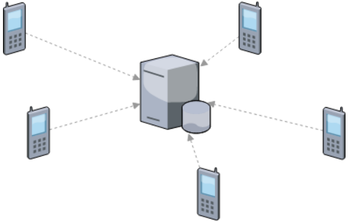
Scalable Methods

- Scalable Cosine
- Landmark Based Methods
- Bipartite Graph Models

Cluster Interpretation
Comparisons
Conclusion

Background

- **Verizon** has a large amount of browsing data from their **cell phone users**.
- **Problem:** How can we draw insights from this data?



CAMCOS

- **Spring 2017**
  - Proof of concept study based on a documents dataset
  - Focused on a general framework: preprocessing, similarity measures, different clustering algorithms
- **Spring 2018**
  - Focused on speed improvements for different spectral clustering algorithms
  - Understanding the content of the clusters

Clustering

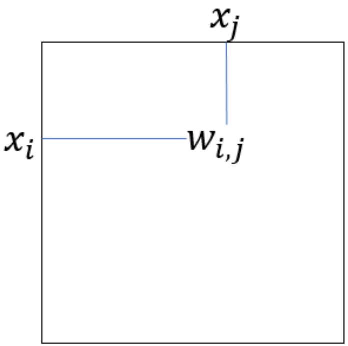
- Clustering is an unsupervised machine learning task that groups data such that:
  - Data within a group are more similar to each other than data in different groups
- Possible applications for Verizon:
  - Customer and market segmentation
  - Grouping web pages

Clustering Components

- Data matrix $x_i, \ldots, x_n \in R^d$
- A specified number of clusters
- Similarity measure
- Criterion to evaluate the clusters

Similarity

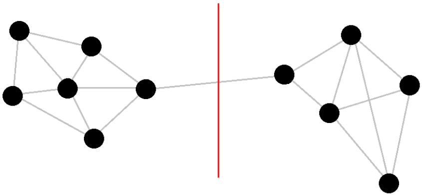
- Similarity describes how alike two observations are
- $w_{i,j} = S(x_i, x_j)$
- Common similarity measures:
  - Gaussian similarity
  - Cosine similarity



A weight matrix, $\boldsymbol{W}$

Spectral Clustering

Spectral clustering = graph cut!



Weighted graphs are composed of:

- Vertices: x_i
- Edges: $x_i \longleftrightarrow x_j$
- Weights: $W = (w_{ij})$

New problem: Find the "best" cut

More Graph Terminology

- Degree matrix - each degree sums the similarities for one observation

$$D = diag(W \cdot \vec{1})$$

- Transition matrix

$$P = D^{-1}W$$

Note: $P\vec{1} = \vec{1}$ ($\vec{1}$ is an eigenvector associated to the largest eigenvalue, 1)

Spectral Clustering (Normalized Cut)

Criterion:

$$min_{A,B}\, Ncut(A,B) = \frac{Cut(A,B)}{Vol(A)} + \frac{Cut(A,B)}{Vol(B)}$$

Can be shown to be approximated by solving an eigenvalue problem:

$$Pv = \lambda v$$

and use the second largest eigenvector for clustering.

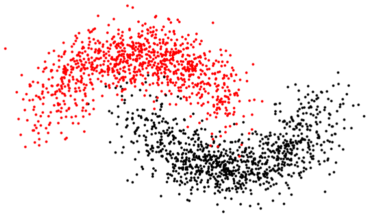
For k clusters, we would use the second to kth eigenvectors for k-means clustering

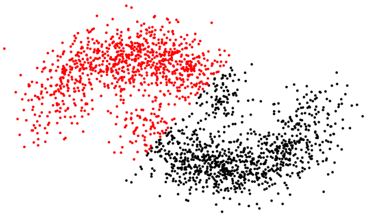
Ng, Jordan, Weiss Spectral Clustering (NJW)

Other clustering algorithms use similar weight matrices for decomposition:

- $\tilde{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is similar to P from Ncut
- NJW uses the eigenvectors of $\tilde{W}$ for spectral clustering
- Note: Diffusion maps is another clustering method. It uses the eigenvectors and eigenvalues of P^t for clustering

Spectral Clustering vs kmeans Clustering





Pros and Cons of Spectral Clustering

Pros

- Relatively simple to implement
- Equivalent to some graph cut problems
- Handles arbitrarily shaped clusters

Cons

- Computationally expensive for large datasets
- $O(n^2)$ storage
- $O(n^3)$ time

Project Overview

Goal: Each team focused on one idea for improving the scalability

- Team 1
  - Use cosine similarity and clever matrix manipulations to avoid the calculation of W
- Team 2
  - Use landmarks to find a sparse representation of the data
- Team 3
  - Use landmarks and given data to build bipartite graph models

Datasets Considered

Type	Dataset	Instances	Features	Classes
Text	20Newsgroups	18,768	55,570	20
	Reuters	8,067	18,933	30
	TDT2	9,394	36,771	30
Image	USPS	9,298	256	10
	Pendigits	10,992	16	10
	MNIST	70,000	784	10

Sample Text Data - Sparse

Word Count	**Word 1**	**Word 2**	**Word 3**	...	**Word d**
Document 1	0	0	6	...	0
Document 2	2	0	1	...	2
Document 3	1	4	0	...	0
...	...	...	...	...	...
...	...	...	...	...	...
Document n	0	8	0	...	0

Sample Image Data - Low Dimension

Pixel Intensity	**Pixel 1**	**Pixel 2**	**Pixel 3**	...	**Pixel d**
Image 1	41	100	6	...	80
Image 2	20	100	25	...	70
Image 3	20	95	40	...	44
...	...	...	...	...	...
...	...	...	...	...	...
Image n	100	0	0	...	50

Scalable Spectral Clustering using Cosine Similarity

Team 1

Group Leader: Jeffrey Lee

Team Members: Xin Xu, Xin Zhang, Zhengxia Yi

Overview of NJW Spectral Clustering

Input: Data A, specified number k, α fraction cutoff for outliers

1. $W = (w_{i,j}) \in R^{n \times n}$, where $w_{i,j} = S(x_i, x_j)$
2. $D = diag(W \cdot \vec{1})$
3. Symmetric normalization: $\tilde{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
4. Compute the top k eigenvectors of $\tilde{W}$
5. Run K-means on $\tilde{U}$ to cluster.

Output: Cluster labels

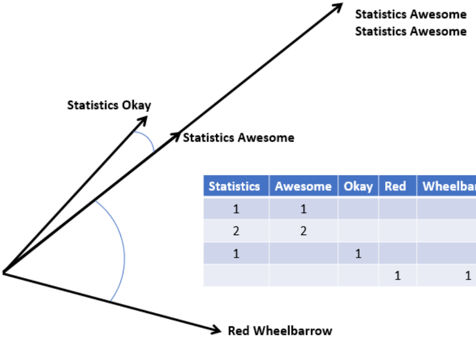
Setting for Scalable Spectral Clustering

- **Relevance of Cosine Similarity:** Many clustering problems involve document data or image data. For these types of data, cosine similarity is appropriate to use.
- **Main idea:** Although the similarity matrix is very expensive in spectral clustering, we can omit the similarity matrix calculation and still be able to cluster under cosine similarity.
- **Assumptions**:
  - The data is sparse or low dimensional
  - Cosine similarity is used: $W = AA^T - I$

Cosine Similarity

$$S(x, y) = cos\theta = \frac{x \cdot y}{||x|| \cdot ||y||}$$

- Measures content overlap with the bag-of-words model
- Removes influence of document length
- Fast to compute



Statistics	Awesome	Okay	Red	Wheelbarrow
1	1			
2	2			
1		1		
			1	1

Math derivation: If plug in $W = AA^T - I$, we will have:

1. $D = diag(W \cdot \vec{1})$

   $= diag((AA^T - I) \cdot \vec{1})$

   $= diag(A(A^T\vec{1}) - \vec{1})$

   without the need of W

2. $\tilde{W} = D^{-\frac{1}{2}}(AA^T - I)D^{-\frac{1}{2}}$

   $= D^{-\frac{1}{2}}AA^TD^{-\frac{1}{2}} - D^{-1}$

   $= \tilde{A}\tilde{A}^T - D^{-1}$

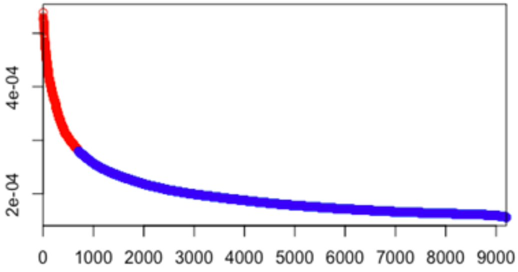
   where $\tilde{A} = D^{-\frac{1}{2}}A$

If D^{-1} has constant diagonals, then left singular vectors of $\tilde{A}$ = eigenvectors of $\tilde{W}$.

So, with just A, clustering is more efficient and does not rely on W.

Outlier Cutoff

Entries of D^{-1} ordered from largest to smallest (USPS data)



Discard outliers without changing the eigenspace of $\tilde{W}$

Implementing the Scalable Spectral Clustering Algorithm

Input: Data A, Specified number k, clustering method (NJW, Ncut or DM) and α fraction cutoff for outliers

1. L2 normalize data A. Compute degree matrix D, remove outliers from D and A
2. Compute $\tilde{A} = D^{-\frac{1}{2}}A$
3. Compute the $\tilde{U}$, the top k left singular vectors of $\tilde{A}$
4. Convert $\tilde{U}$ according to clustering method and run K-means

Output: Cluster labels, including a label for outliers

Experimental Settings

- $\alpha = 1\%$
- methods: NJW and Scalable NJW
- both algorithms coded by our team
- golub server at San José State University
- six data sets (three image data, three text data)

Benchmark - Accuracy Comparison

Scalable Spectral Clustering vs. Plain NJW Spectral Clustering

Accuracy (%)		
Dataset	**Scalable**	**Plain**
20Newsgroup	64.40	64.95
Reuters	24.60	25.23
TDT2	51.20	51.80
USPS	67.53	67.47
Pendigits	73.56	73.56
Mnist	52.60	Out of Memory

- Both methods are similar in accuracy. The Plain method is slightly more accurate.

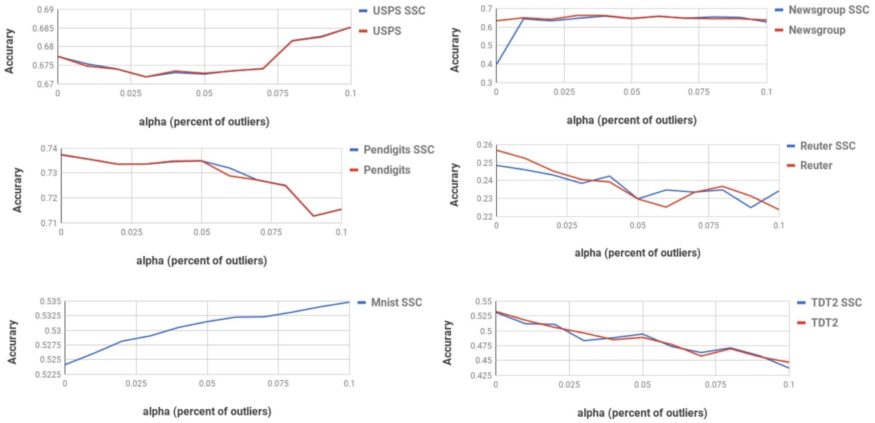
Benchmark - Runtime Comparison

Scalable Spectral Clustering vs. Plain NJW Spectral Clustering

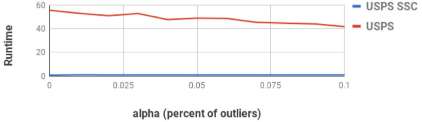
Runtime (Seconds)		
Dataset	**Scalable**	**Plain**
20Newsgroup	57.7	154.9
Reuters	5.9	51.1
TDT2	25.3	53.9
USPS	1.1	52.9
Pendigits	3.4	102.0
Mnist	36.2	Out of Memory

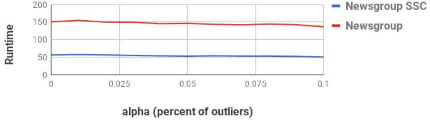
- The Scalable method is much faster than the Plain method.

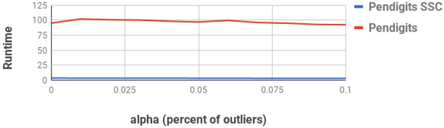
Robustness To Outliers (Accuracy)

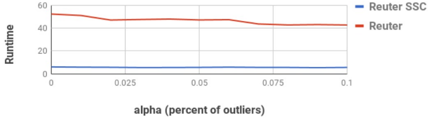


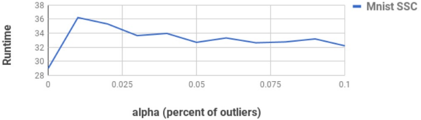
Robustness To Outliers (Runtime)

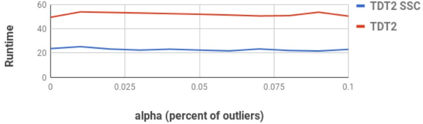












General Remarks and Results From Experiments

- The scalable spectral clustering method is fast and comparably accurate.
- In general insensitive to choice of α.

Further Studies and Considerations

- More experiments on other clustering methods (NCut, DM).
- Extend our method to handle other similarities (Gaussian).

Landmark-based Spectral Clustering

Team 2

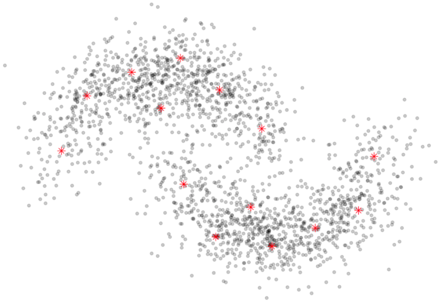
Group Leader: Scott Li

Team Members: Jiye Ding, Maham Niaz

Landmark-based Spectral Clustering (LSC) Steps:

Main Idea: Use landmarks to find a sparse representation of the data

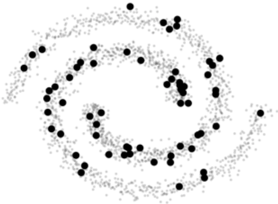
- Landmark selection
- Affinity matrix computation
- Nearest landmarks
- Normalization, SVD, k-means



Landmark Selection

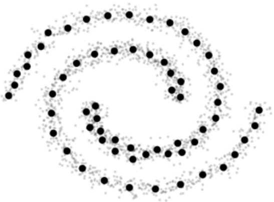
Random Selection

- Very fast



k-means Selection

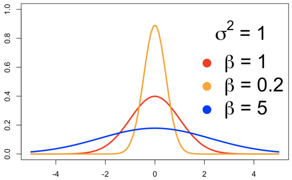
- Very slow for larger datasets
- Can be more representative



Affinity Matrix Computation

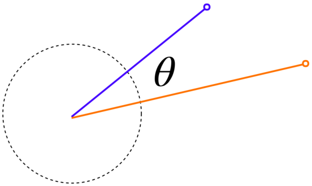
Gaussian Similarity

$$S(x,y) = e^{-\frac{||x-y||^2}{2\beta\sigma^2}}$$



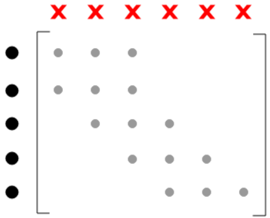
Cosine Similarity

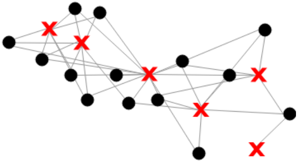
$$S(x,y) = cos\theta = \frac{x \cdot y}{||x|| \cdot ||y||}$$



Nearest Landmarks

- The largest r entries in each row are kept. The rest are set to zero.
- Makes the affinity matrix sparse, speeding up computations
- Makes clustering more robust to noise





Data Clustering

- $L1$ row normalization, then $\sqrt{L1}$ column normalization on A
- Find the top k left singular vectors $(u_1 ... u_k)$
- k-means outputs cluster assignments on the data

Landmark Clustering - new method

- Cluster landmarks based on the top k right singular vectors $(v_1 ... v_k)$
- Use k-NN to classify the original data

Experiments

- 20 Seeds
- Cosine Similarity
- Compare Landmark Selection Method and Clustering Method
  - $p = 500$, $r = 6$
- Parameter Sensitivity
  - Number of Landmarks (p)
  - Number of Nearest Landmarks (r)

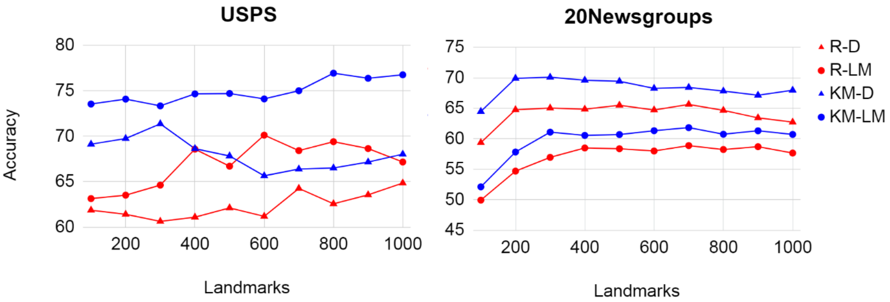
Results

Accuracy (%)					
Dataset	Random LM Selection		k-means LM Selection		NJW
	Data Clustering	Landmark Clustering	Data Clustering	Landmark Clustering	
20Newsgroups	65.51	58.37	**69.42**	60.69	63.36
Reuters	25.37	27.50	27.38	**31.21**	25.68
TDT2	59.85	64.34	59.45	**65.69**	44.38
USPS	62.12	66.70	67.83	**74.70**	67.74
Pendigits	78.81	78.76	77.94	**81.59**	73.75
MNIST	63.32	59.41	**69.43**	65.10	–

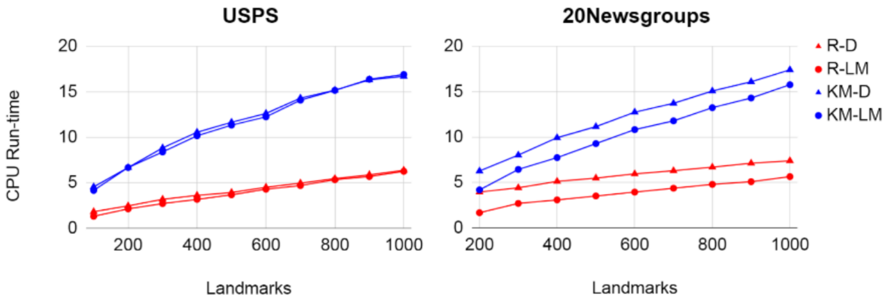
CPU Run-time (s)					
Dataset	Random LM Selection		k-means LM Selection		NJW
	Data Clustering	Landmark Clustering	Data Clustering	Landmark Clustering	
20Newsgroups	5.95	**3.78**	12.75	11.16	150.96
Reuters	7.38	**6.61**	451.88	444.28	52.31
TDT2	12.12	**11.67**	1912.68	1862.29	49.46
USPS	3.93	**3.56**	11.65	11.76	55.46
Pendigits	2.70	**2.25**	3.76	3.63	95.13
MNIST	31.05	**27.62**	584.06	619.06	–

Parameter Sensitivity

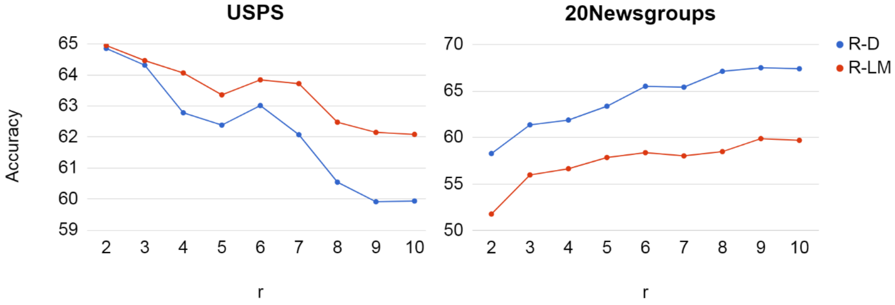
Varying the Number of Landmarks - Accuracy



Varying the Number of Landmarks - CPU Run-time



Varying the Number of Nearest Landmarks - Accuracy



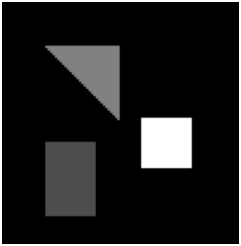
Conclusions

- LSC techniques can improve the speed and accuracy over NJW
- Random landmark selection is very efficient
- Landmark clustering is often more accurate
- Accuracy can be sensitive to the parameters

Spectral Clustering for Image Segmentation

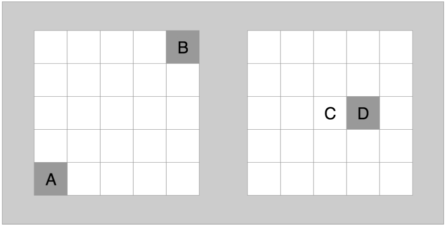
Image Segmentation:

Given an image, partition it into different regions for different objects.



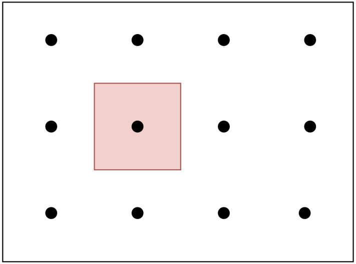
Original Spectral Clustering

- Input data: $m \times n$ pixels
- Similarity measure: location and intensity



New Methods of Image Segmentation by LSC

- NJW: $W \in \mathcal{R}^{(mn)\times(mn)}$
- A grid of representative pixels are landmarks
- Only consider the pixels close to each landmark



Example 1

Image Size: 115×71

NJW Result

time $= 28.02$

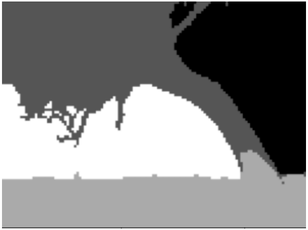
LSC Result

time $= 3.55$

Example 2

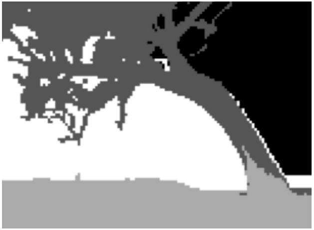
Image Size: 125×75

NJW Result



time $= 74.17$

LSC Result



time $= 6.85$

Landmark-based Bipartite Graph Spectral Clustering

Team 3

Team Member: Khiem Pham

Motivation

EVD of $n \times n$ matrix: $O(n^3)$ time.
SVD of $n \times m$ matrix, $m \ll n$: $O(nm^2 + m^3)$ time, **linear** in n.

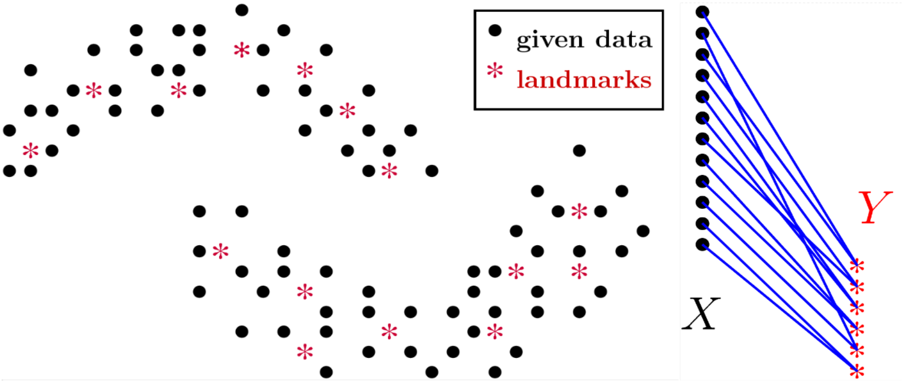
Team 1: avoid forming affinity matrix

Team 2: dictionary learning + sparse coding feature

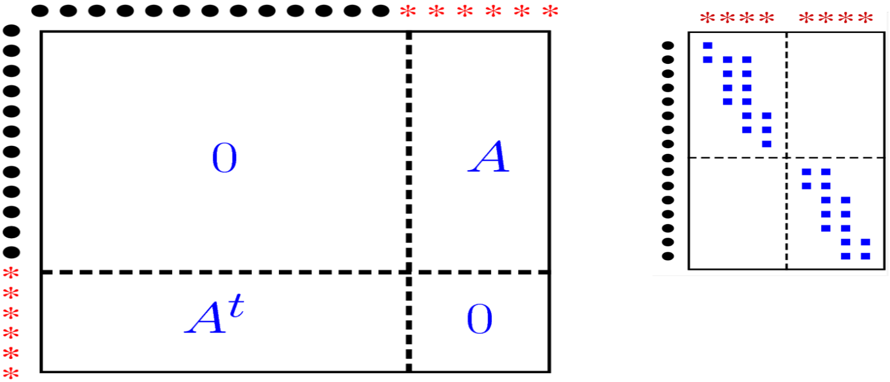
A more "native" approach?

Bipartite Graph

- Pick representative landmarks



- Form affinity matrix between landmarks and datapoints



Proposition

$A \in R^{n*m}$: affinity matrix between n data points and m landmarks D_1 (D_2): diagonal matrices of row (column) sums of A.

Then the eigenvectors of $P = \begin{pmatrix} D_1^{-1} & \\ & D_2^{-1} \end{pmatrix} \begin{pmatrix} & A \\ A^t & \end{pmatrix}$ are:

$$V = \begin{pmatrix} D_1^{-1/2}\widetilde{V}_1 \\ D_2^{-1/2}\widetilde{V}_2 \end{pmatrix}$$

where $\widetilde{V}_1$ and $\widetilde{V}_2$ are left and right singular vectors of:

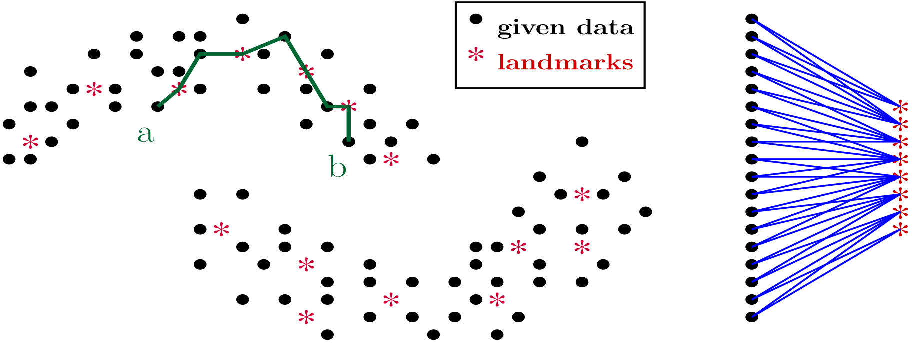
$$\widetilde{A} = D_1^{-1/2} A D_2^{-1/2} \in R^{n \times m}$$

which can be computed in $O(nm^2 + m^3) time$

Landmark-based Bipartite Graph Spectral Clustering

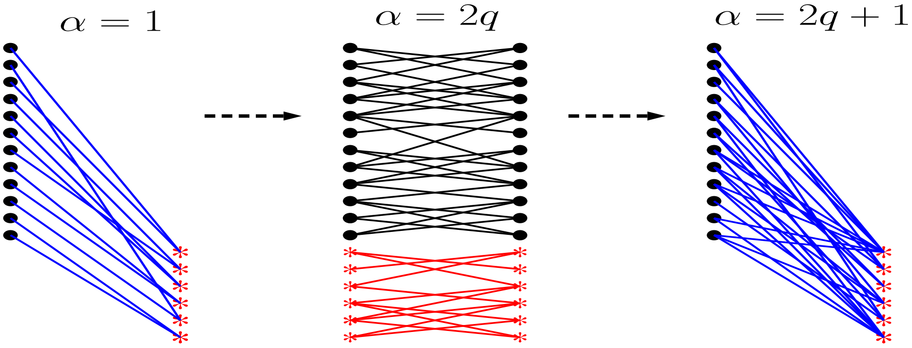
Diffusion Map

- Generate random walks on bipartite graph.
- "Enhance" global affinity of far-away data points.



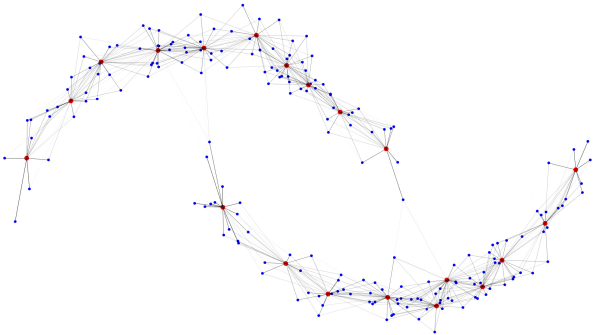
Landmark-based Bipartite Graph Spectral Clustering

- For odd time step, **co-clustering**
- For even time step, **direct clustering** or **landmark clustering** (with extension)



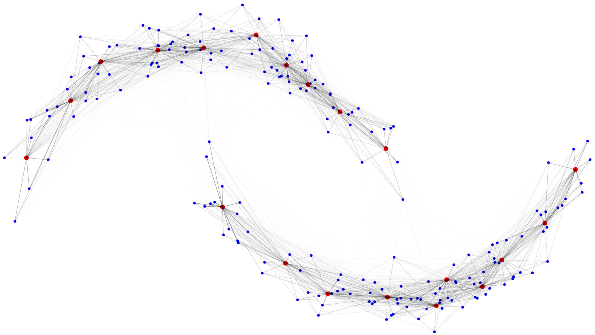
Landmark-based Bipartite Graph Spectral Clustering

t=1, data points <-> landmarks



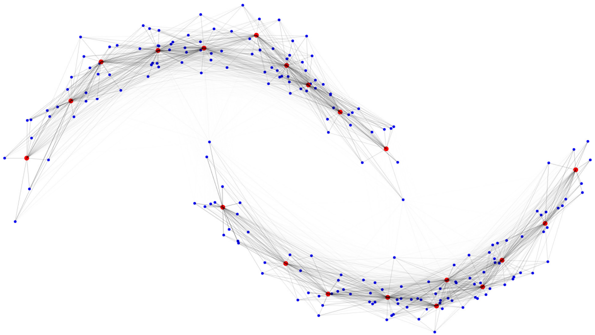
Landmark-based Bipartite Graph Spectral Clustering

t=5, data points <-> landmarks



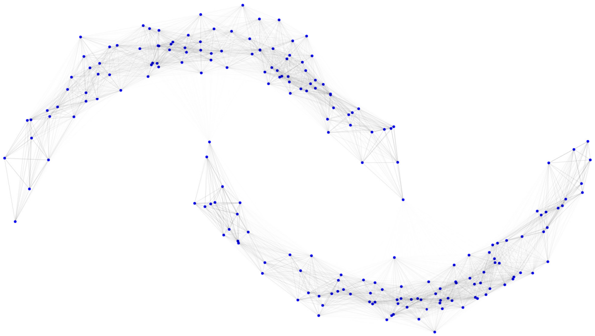
Landmark-based Bipartite Graph Spectral Clustering

t=9, data points <-> landmarks



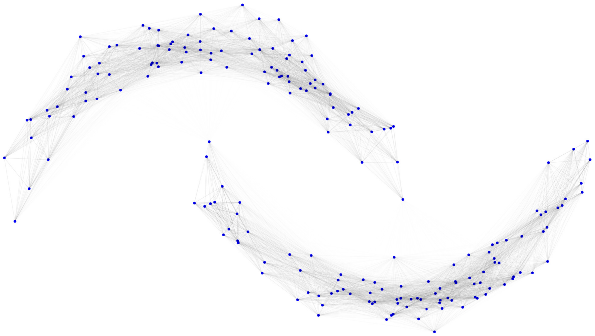
Landmark-based Bipartite Graph Spectral Clustering

t=2, data points <-> data points



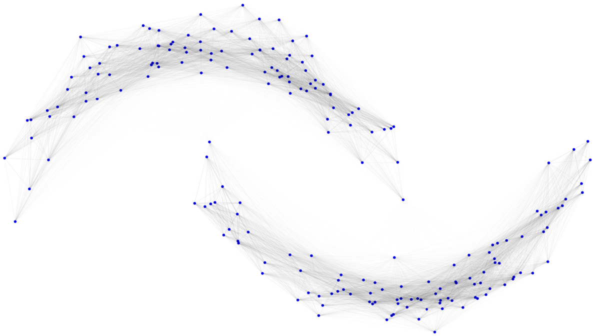
Landmark-based Bipartite Graph Spectral Clustering

t=6, data points <-> data points



Landmark-based Bipartite Graph Spectral Clustering

t=10, data points <-> data points



Experiment Results (accuracy)

LBDM$^{(1)}$: diffusion map, co-clustering, time step $= 1$
LBDM$^{(2,X)}$: diffusion map, direct clustering, time step $= 2$
LBDM$^{(2,Y)}$: diffusion map, landmark clustering, time step $= 2$

Dataset	Ncut	KASP	LSC	cSPEC	Dhillon	LBDM$^{(1)}$	$\_^{(2,X)}$	$\_^{(2,Y)}$
usps	66.21	67.25	66.86	66.89	68.21	67.80	68.10	**69.45**
pendigits	69.73	68.45	**77.93**	67.93	73.20	72.95	74.70	73.22
letter	24.93	26.19	31.51	24.98	32.06	32.13	**32.21**	31.28
protein	43.68	43.85	43.85	44.84	43.35	43.55	43.16	**45.88**
shuttle		74.52	39.71	**82.78**	74.24	74.26	74.38	74.49
mnist		57.99	70.28	54.50	72.15	72.43	72.37	**73.29**

Experiment Results (Time)

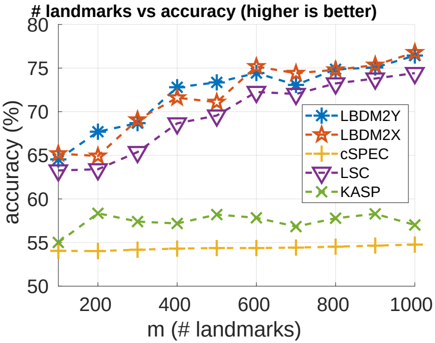
LBDM$^{(1)}$: diffusion map, co-clustering, time step $= 1$
LBDM$^{(2,X)}$: diffusion map, direct clustering, time step $= 2$
LBDM$^{(2,Y)}$: diffusion map, landmark clustering, time step $= 2$

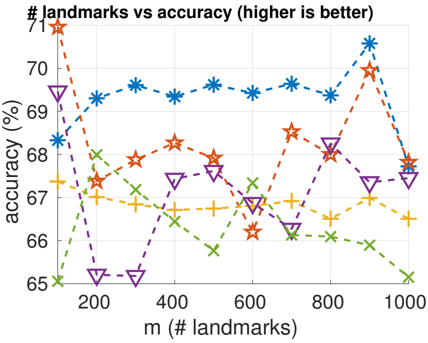
Dataset	Ncut	(k-means)	KASP	LSC	cSPEC	Dhillon	LBDM$^{(1)}$	–$^{(2,X)}$	–$^{(2,Y)}$
usps	131.78	7.46 +	0.61	4.44	7.89	4.45	4.39	4.17	1.95
pendigits	246.08	3.13 +	0.55	3.08	5.26	3.14	2.91	3.08	1.65
letter	1180.70	5.30 +	0.77	12.24	25.07	13.51	14.96	12.87	2.78
protein	2024.54	27.04 +	0.41	3.55	7.54	3.93	4.04	3.93	4.40
shuttle		23.89 +	1.23	8.49	61.68	12.35	15.09	12.15	5.88
mnist		299.74 +	0.63	25.07	39.26	27.17	25.69	25.83	16.67

Parameter Sensitivity

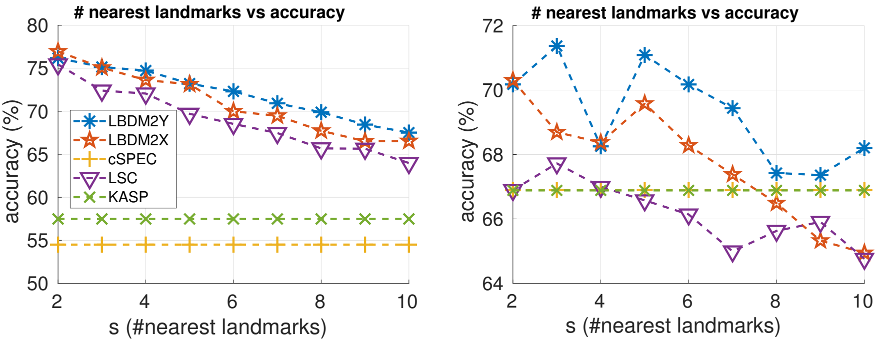
- Investigate the influence of each parameter on MNIST and USPS
- Baseline configuration:
  - # landmarks = 500.
  - # nearest neighbors = 5.
  - # random walk length/time step = 2.

- Varying number of landmarks

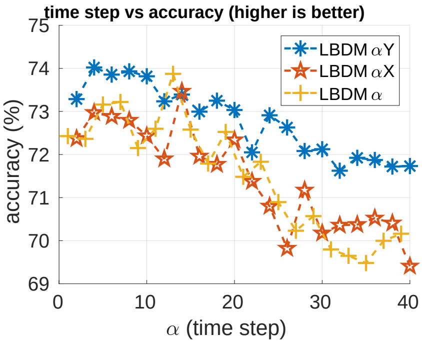


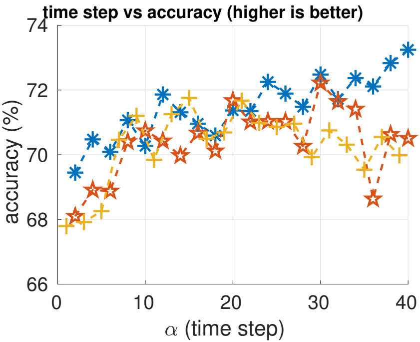


- Varying number of nearest landmark neighbors



- Varying time step





Biparite graph model of documents and words

- Applicable to text data.
- Each document is a bag-of-word (ignoring syntax)
- Documents are data points (to be clustered), words are landmarks (not artificial landmarks).

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

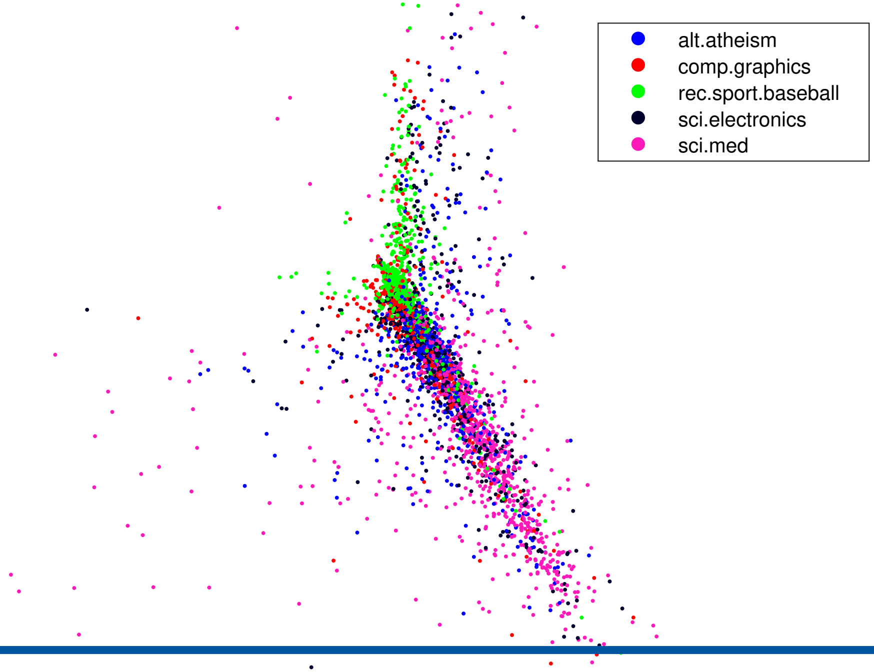
Landmark-based Bipartite Graph Spectral Clustering

- Recall: eigenvectors are embeddings of data points and landmarks
- Get embeddings of both documents and words
- Great for dimensionality reduction and visualization (similar to Laplacian Eigenmap[1])

[1] Belkin, Mikhail, and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation." Neural computation 15, no. 6 (2003): 1373-1396.

Problem

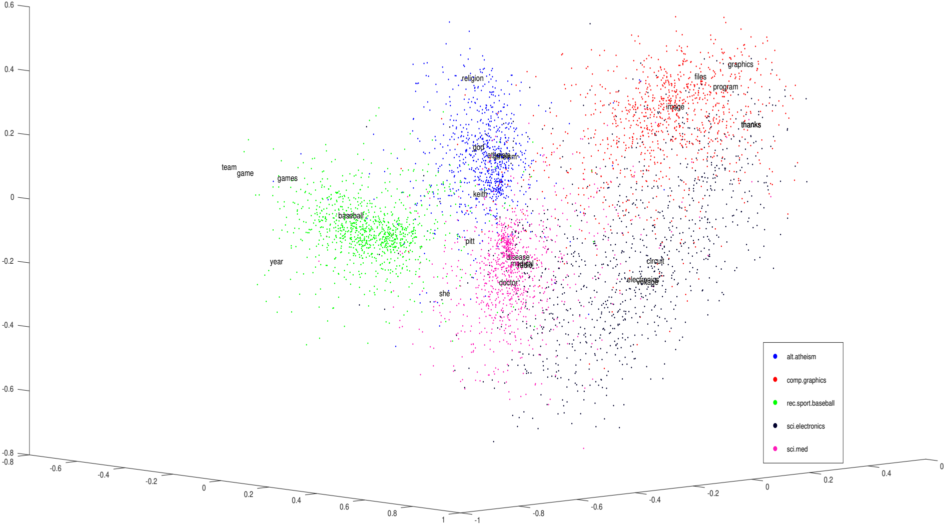
- 20 news accuracy: 26.09%
- due to sparse matrix, many low degree words, several low degree documents
- can remove low degree nodes in graph, but lose information
- ?


alt.atheism
comp.graphics
rec.sport.baseball
sci.electronics
sci.med

Solution

- Based on recent works on degree-corrected stochastic block model, "inflate" degree of node:[2]
  - $\widetilde{D}_1 = D_1 + \tau_1 I$
  - $\widetilde{D}_2 = D_2 + \tau_2 I$
  - $\widetilde{A} = \widetilde{D}_1^{-1/2} A \widetilde{D}_2^{-1/2}$
- Accuracy: 63.94%

[2] Rohe, Karl, and Bin Yu. "Co-clustering for directed graphs; the stochastic co-blockmodel and a spectral algorithm." stat 1050 (2012): 10.



religion
god
atheists
atheism
keith
team
game
games
baseball
year
pitt
she
disease
medical
radio
doctor
graphics
files
program
image
thanks
circuit
electronics
voltage
alt.atheism
comp.graphics
rec.sport.baseball
sci.electronics
sci.med
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4

Concluding Remarks

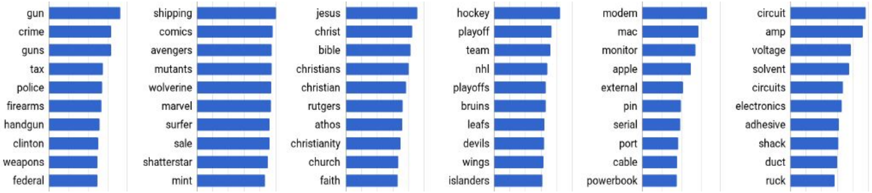
Text Cluster Interpretation

Singular Value Decomposition: Take the first basis vector of each cluster

Frequencies Ranking: Rank all words based on total frequency inside each cluster

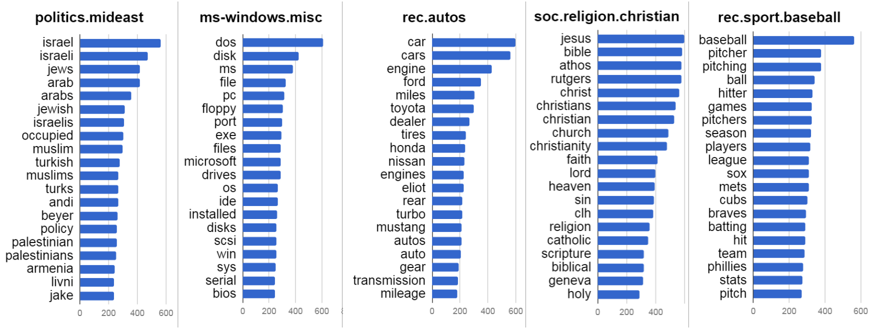
Text Cluster Interpretation

- After clustering, we use rank 1 singular value decomposition to obtain the first basis vector of each cluster.
- The top entries in each first basis vector represent important words in that cluster.



Text Cluster Interpretation

Rank all words based on the total frequency inside each cluster



Team Comparisons

	1. Cosine		2. Landmark		3. Bipartite	
Dataset	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
USPS	67.5	(1.1)	**74.7**	(11.8)	69.5	(9.4)
Pendigits	73.6	(3.4)	**81.6**	(3.6)	74.7	(6.2)
MNIST	52.6	(36.2)	69.4	(584.1)	**73.3**	(316.4)
TDT2	51.2	(25.3)	64.3	(11.7)	**70.8**	(38.1)
Reuters	24.6	(5.9)	27.5	(6.6)	**38.3**	(36.6)

Conclusion

- We worked on three ideas for scalable spectral clustering methods
- They are often faster and more accurate than older spectral clustering algorithms
- Next: Clustering data provided by Verizon

Future Work

- More Evaluation Metrics
  - F_1 score
- Recursive Partitioning
  - Finds a hierarchical structure
  - Useful for determining the number of clusters
- Clustering Browsing History with Demographic Data
  - Categorical data

Acknowledgements

- We would like to thank Prof. Guangliang Chen for his guidance and supervision with this project and Prof. Slobodan Simic for helping to organize this project
- Thanks to Verizon for their generous sponsorship

References

[1] A.Y. Ng, M. I. Jordan, Y. Weiss *"On Spectral Clustering: Analysis and an Algorithm", NIPS Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, pp: 849-856* MIT Press Cambridge, MA, USA, Dec 2001

[2] U. Von Luxburg *"A tutorial on spectral clustering", Statistics and Computing, 17(4):pp 395-416,*2007

[3] Zelnik-Manor, Lihi, P. Perona. *"Self-tuning spectral clustering."* Advances in neural information processing systems. 2005

[4] G. Chen, *"Scalable spectral clustering with cosine similarity."* To appear in the Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China. 2018

[5] J. Fitch et al., *"Adaptive Spectral Clustering for High-Dimensional Sparse Count Data"* Dept. Math., San Jose State Univ., San Jose, CA, 2017

[6] D. Cai, X. Chen, *"Large Scale Spectral Clustering Via Landmark-Based Sparse Representation"* IEEE Trans. Cybernetics, Vol 45 Issue 8, August 2015

Questions?