Large-scale spectral clustering using diffusion coordinates on landmark-based bipartite graphs

Guangliang Chen and Khiem Pham

TextGraphs-2018 Workshop, New Orleans, LA

June 6, 2018

Outline

- Introduction + background
- Our scalable approach
- Experiments
- Conclusions

What is spectral clustering?

A family of clustering algorithms that utilize the **spectral decomposition of a similarity matrix** constructed on the given data $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$:

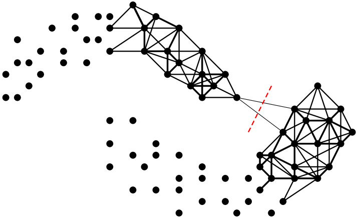
$$\mathbf{W} = (w_{ij}) \in \mathbb{R}^{n \times n}, \quad w_{ij} = \begin{cases} \kappa(\mathbf{x}_i, \mathbf{x}_j), & \text{if } i \neq j \\ 0, & \text{if } i = j. \end{cases}$$

Here, $\kappa(\cdot, \cdot)$ is a similarity function, such as

- an **indicator function** (whether two points are "sufficiently close"),
- the **Gaussian radial basis function (RBF)**, and
- the **cosine similarity**.

A (very convenient) graph cut point of view

$\mathbf{W}$ (as a weight matrix) defines a weighted graph on the given data.



Accordingly, clustering = finding an optimal cut (under some criterion): e.g., RatioCut, NCut, MinMaxCut.

Some graph terminology:

–**Degree matrix** $\mathbf{D} = \mathrm{diag}(\mathbf{W1})$, with $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.

–**Graph Laplacian** $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and its normalized versions:

$$\mathbf{L}_{\mathrm{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \underbrace{\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}}_{\widetilde{\mathbf{W}}\ \text{(symmetric)}}$$

$$\mathbf{L}_{\mathrm{rw}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \underbrace{\mathbf{D}^{-1}\mathbf{W}}_{\mathbf{P}\ \text{(row stochastic)}}$$

Different spectral clustering algorithms

...use different kinds of spectral embedding for $k$means clustering:

- The **Ng-Jordan-Weiss (NJW)** algorithm (NIPS'01): $\widetilde{\mathbf{U}} \in \mathbb{R}^{n\times k}$, top k eigenvectors of $\widetilde{\mathbf{W}}$:

$$\widetilde{\mathbf{W}} \approx \widetilde{\mathbf{U}}_{n\times k}\mathbf{\Lambda}_{k\times k}\widetilde{\mathbf{U}}^T_{n\times k}, \qquad \text{where} \quad \mathbf{\Lambda} = \mathrm{diag}(\lambda_1 \geq \cdots \geq \lambda_k)$$

- **Normalized Cut (NCut)** by Shi and Malik (PAMI'00): $\mathbf{U} = \mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{U}}$, top k eigenvectors of $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$

- **Diffusion Maps (DM$^{(t)}$)** by Coifman et al. (PNAS'05): $\mathbf{U}^{(t)} = \mathbf{U}\mathbf{\Lambda}^t = \mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{U}}\mathbf{\Lambda}^t$, diffusion coordinates in t time steps

Computational challenges

Spectral clustering has achieved superior results in many applications (such as image segmentation, documents clustering, social network partitioning), but requires significant computational power due to the matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$:

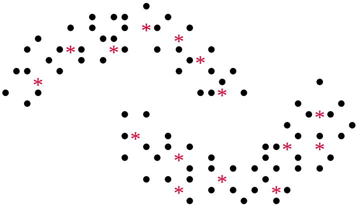
- Extensive memory requirement: $\mathcal{O}(n^2)$
- High computational cost: $\mathcal{O}(n^3) \leftarrow$ spectral decomposition

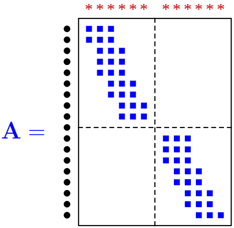
Consequently, there has been an urgent need to develop **fast**, **approximate** spectral clustering algorithms that are **scalable to large data**.

Landmark-based scalable methods

Most existing scalable methods use a small landmark set $\mathbf{y}_1, \ldots, \mathbf{y}_m \in \mathbb{R}^d$, selected from the **given data** $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ (e.g., uniformly at random or via k-means), to first construct a (sparse) similarity matrix between them:

$$\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times m}, \quad a_{ij} = \kappa(\mathbf{x}_i, \mathbf{y}_j)$$





Landmark-based Bipartite Diffusion Maps (LBDM)

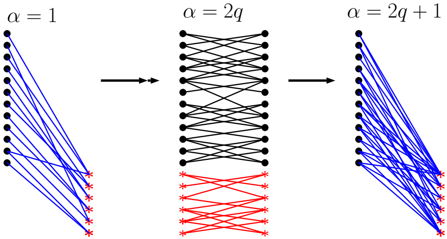
Afterwards, different methods use the similarity matrix $\mathbf{A}$ in different ways:

- **cSPEC** (Wang et al., 2009): Regards $\mathbf{A}$ as a **column-sampled** version of $\mathbf{W}$ and uses linear algebra to estimate eigenvectors of $\mathbf{W}$
- **KASP** (Yan, Huang and Jordan, 2009): Uses **vector quantization** technique (k-means) to aggressively reduce the given data to a collection of centroids (landmarks) and applies spectral clustering to group them
- **LSC** (Cai and Chen, 2015): Obtains the matrix $\mathbf{A}$ from a **sparse coding** perspective with the landmarks as a dictionary and then applies spectral clustering to the rows of $\mathbf{A}$ (after performing certain row and column normalizations).

Overview of our approach

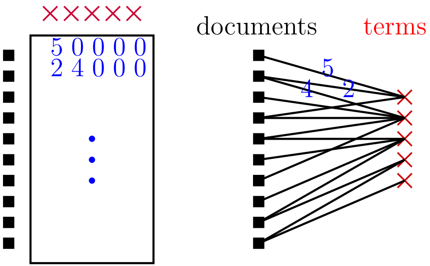
We follow the direction of landmark-based spectral clustering, but

- use the landmarks to form a bipartite graph,
- and then run a **random walk** on the graph.



Motivation

Dhillon (2001) proposed a **bipartite graph** model for the setting of documents data, with the goal to **co-cluster documents and terms**.



Frequency matrix (under bag of words model)

In principle, they apply NCut to the mixture of documents and terms with the weight matrix

$$\mathbf{W} = \begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}.$$

They derived an efficient procedure for computing the eigenvectors of $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ directly from $\mathbf{A}$.

Landmark-based Bipartite Diffusion Maps (LBDM)

Theorem (Dhillon'01). Let

$$\mathbf{D}_1 = \mathrm{diag}(\mathbf{A}\mathbf{1}), \quad \mathbf{D}_2 = \mathrm{diag}(\mathbf{A}^T\mathbf{1}), \quad \mathbf{D} = \mathrm{diag}(\mathbf{W}\mathbf{1}) = \begin{pmatrix} \mathbf{D}_1 & \\ & \mathbf{D}_2 \end{pmatrix},$$

and

$$\widetilde{\mathbf{A}} = \mathbf{D}_1^{-1/2}\mathbf{A}\mathbf{D}_2^{-1/2}.$$

Then for each pair of left and right singular vectors $\widetilde{\mathbf{A}}\widetilde{\mathbf{v}}_2 = \sigma\widetilde{\mathbf{v}}_1$,

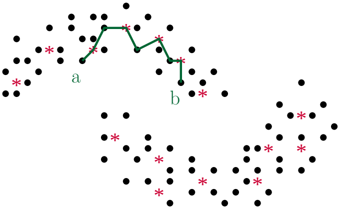
$$\mathbf{v} = \begin{pmatrix} \mathbf{D}_1^{-1/2} & \\ & \mathbf{D}_2^{-1/2} \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{v}}_1 \\ \widetilde{\mathbf{v}}_2 \end{pmatrix} = \mathbf{D}^{-1/2}\,\widetilde{\mathbf{v}}$$

is an eigenvector of $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ (corresponding to eigenvalue σ).

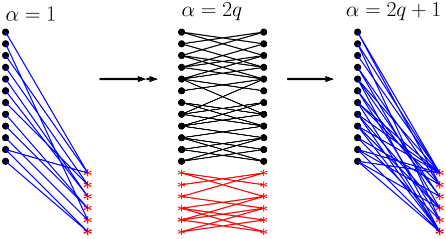
Our approach

We extend the bipartite graph model by Dhillon (2001) in two ways:

(1) We adapt it for landmark-based clustering by using instead the given data and a landmark set as its two parts.



(2) We run a **random walk** on the bipartite graph to gather global information about the data set.



Two important remarks

(1) The diffusion coordinates at any time step α can be computed efficiently from $\mathbf{A}$ too:

$$[\cdots \mid \sigma^\alpha \mathbf{v} \mid \cdots]$$

(2) Depending on whether α is odd or even, there are three ways to cluster the given data:

- α even (two disjoint subgraphs): **direct clustering**, or **landmark clustering + NN classification** (faster)
- α odd (still a bipartite graph): **co-clustering** first but removing the landmarks later

Alg. 1 Landmark-based Bipartite Diffusion Maps (LBDM)

Input:

- Data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
- similarity function κ
- $\#$ clusters k
- $\#$ diffusion steps α
- landmark selection method
- $\#$ landmark points m
- $\#$ nearest landmark points s
- clustering method (direct, landmark, or co-clustering)

Output: Clusters $C_1, \ldots, C_k$

Landmark-based Bipartite Diffusion Maps (LBDM)

Steps:

1. Select m landmark points $\{\mathbf{y}_j\}$ by the given method.
2. Compute the s-sparse adjacency matrix $\mathbf{A} = (a_{ij}),\ a_{ij} = \kappa(\mathbf{x}_i, \mathbf{y}_j)$ between each given data point $\mathbf{x}_i$ and the s nearest landmarks $\mathbf{y}_j$.
3. Normalize $\mathbf{A}$ by using its row and column sums $\widetilde{\mathbf{A}} = \mathbf{D}_1^{-1/2}\mathbf{A}\mathbf{D}_2^{-1/2}$ and then calculate the diffusion coordinates for the bipartite graph (through the rank-k SVD of $\widetilde{\mathbf{A}}$):

$$[\cdots \mid \sigma^{\alpha}\mathbf{D}^{-1/2}\widetilde{\mathbf{v}} \mid \cdots]$$

4. Use the indicated clustering method to cluster the given data.

Complexity analysis

Total running time is $\mathcal{O}(nm(d+s)+nk(s+k))$, obtained as follows:

- Landmark sampling: $\mathcal{O}(nmd)$ (k-means), or $\mathcal{O}(m)$ (uniform)
- Constructing $\mathbf{A}$: $\mathcal{O}(nm(d+s))$
- Computing $\widetilde{\mathbf{A}}$: $\mathcal{O}(ns)$
- Rank-k SVD of $\widetilde{\mathbf{A}}$: $\mathcal{O}(nsk)$
- Diffusion coordinates: $\mathcal{O}((n+m)k)$
- Final k-means: $\mathcal{O}(nk^2)$ (direct), or $\mathcal{O}(mk^2+ns)$ (landmark), or $\mathcal{O}((n+m)k^2)$ (co-clustering)

Experiments: methods and setup

- To be compared with: **plain NCut, KASP, LSC, cSPEC, Dhillon**
- **Gaussian similarity** $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$
- k**-means sampling** (same landmark points for all)
- **Parameters**: $m = 500$ (for all scalable methods), $s = 5$ (for LSC, Dhillon, and LBDM), $\alpha = 1, 2$ (only for LBDM)
- Evaluation metrics: **clustering accuracy** and **CPU time** (averaged over 50 trials for each method)

Experiments: benchmark data sets

Data	n	d	k
usps	9,298	256	10
pendigits	10,992	16	10
letter	20,000	16	26
protein	24,387	357	3
shuttle	58,000	9	7
mnist	70,000	784	10

Experiments: clustering accuracy (%)

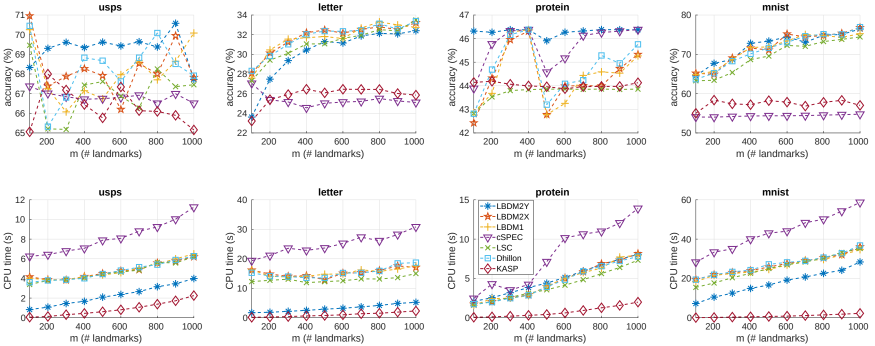
Dataset	Ncut	KASP	LSC	cSPEC	Dhillon	LBDM$^{(1)}$	$\_^{(2,X)}$	$\_^{(2,Y)}$
usps	66.21	67.25	66.86	66.89	68.21	67.80	68.10	**69.45**
pendigits	69.73	68.45	**77.93**	67.93	73.20	72.95	74.70	73.22
letter	24.93	26.19	31.51	24.98	32.06	32.13	**32.21**	31.28
protein	43.68	43.85	43.85	44.84	43.35	43.55	43.16	**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**

Experiments: CPU time (in seconds)

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

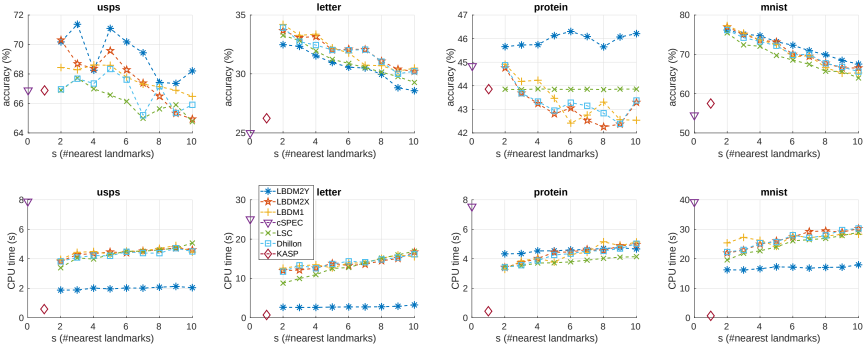
Experiments: parameter study (m)

$s = 5$ fixed (top: clustering accuracy, bottom: CPU time)



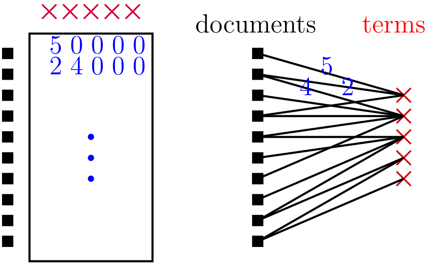
Experiments: parameter study (s)

$m = 500$ fixed (top: accuracy, bottom: CPU time)



LBDM for document-term bipartite graphs

We can apply LBDM directly to documents data by simply treating terms as “landmarks”.



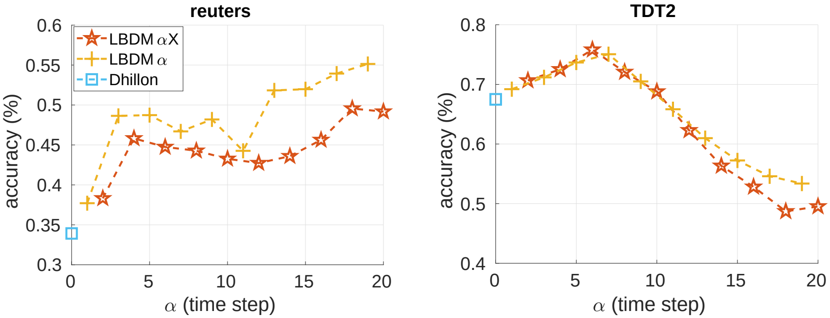
LBDM extends Dhillon’s method by running a random walk on the document-term bipartite graph to construct **document-document**, term-term, and document-term graphs (at different time steps).

Experiments: documents data

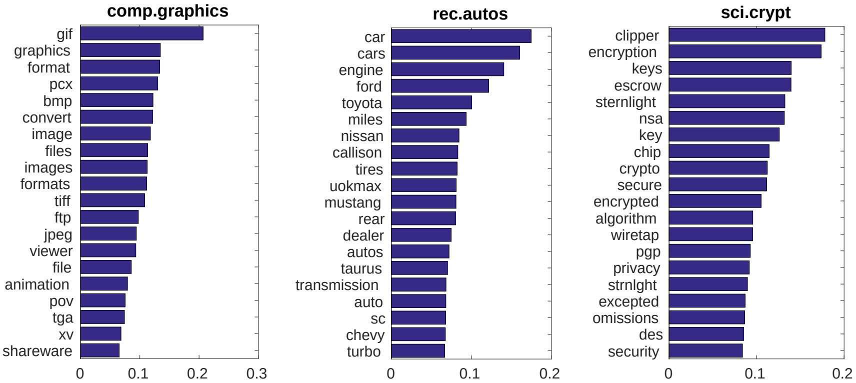
Some documents data used in our experiments:

- **Reuters-21578**: We choose the largest 30 categories with a total of 8,067 documents and 18,933 words.
- **TDT2**: We choose the largest 30 categories with a total of 9,392 documents and 34,090 words.
- **20newsgroups**: There are 18,774 documents (partitioned into 20 newsgroups) and 61,118 words.

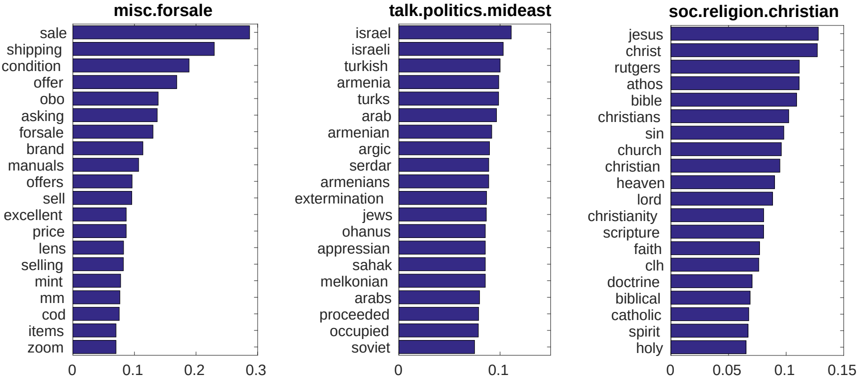
Experiments: accuracy (for different α)



Experiments: topic identification for 20news



Landmark-based Bipartite Diffusion Maps (LBDM)



A unified view

These scalable algorithms are all based on the matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, but may differ in three aspects (besides motivation).

Methods	Sparsity	Normalizations			Clustering
		row	column	order	
LBDM	$1 < s \ll m$	sqrt-L_1	sqrt-L_1	same time	all 3
LSC	$1 < s \ll m$	L_1	sqrt-L_1	row first	direct
Dhillon	$1 < s \ll m$	sqrt-L_1	sqrt-L_1	same time	co-clustering
cSPEC	$s = m$				direct
KASP	$s = 1$				landmark

Thank you for your attention!

We presented a new **landmark-based spectral clustering** method and also provided a **unified view**.

Our algorithm is **simple to implement**, **fast to run**, and **accurate**.

It can be applied for **documents grouping** and **topic identification**.

Acknowledgment. This work was motivated by a project with ***Verizon Wireless****, whose goal was to cluster their cell phone users based on daily website visits.*

Contact: `guangliang.chen@sjsu.edu,khiem.pham@sjsu.edu`