ETH zürich

usi

Spectral clustering using a multilevel approach

Kalman Szenes

Supervisors: Dimosthenis Pasadakis and Prof. Olaf Schenk

Motivation

- Clustering is one of the main techniques in exploratory data analysis

Motivation

- Clustering is one of the main techniques in exploratory data analysis
- Clustering problem can be rewritten as a graph partitioning problem

Motivation

- Clustering is one of the main techniques in exploratory data analysis
- Clustering problem can be rewritten as a graph partitioning problem
    - Solving balanced metric is **NP-Hard**

Motivation

- Clustering is one of the main techniques in exploratory data analysis
- Clustering problem can be rewritten as a graph partitioning problem
  - Solving balanced metric is **NP-Hard**
  - Relaxed problem can be solved using spectral methods

Outline

1. Graphs
2. Graclus Framework
3. Clustering Algorithms
4. Results

Outline

1. Graphs

2. Graclus Framework

3. Clustering Algorithms

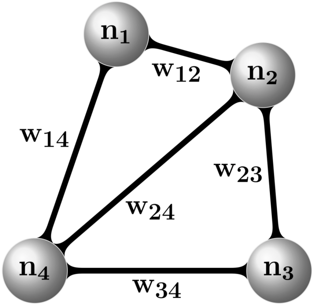
4. Results

Graph Representation[1]

For a graph $\mathcal{G}(\mathcal{V}$, E, W)

- Adjacency: **W** $\in R^{n*n}$
- Degree: **D** $\in R^{n*n}$
- Graph Laplacian: **L** $\in R^{n*n}$

$$\mathbf{W} = \begin{pmatrix} 0 & w_{12} & 0 & w_{14} \\ w_{12} & 0 & w_{23} & w_{24} \\ 0 & w_{23} & 0 & w_{34} \\ w_{14} & w_{24} & w_{34} & 0 \end{pmatrix}, \qquad d_{ii} = \begin{pmatrix} \sum_j w_{1j} \\ \sum_j w_{2j} \\ \sum_j w_{3j} \\ \sum_j w_{4j} \end{pmatrix}$$



[1] Pasadakis, Alappat, Schenk, and Wellein 2021.

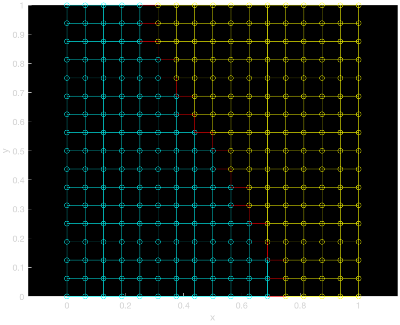
Partition Metric[2]

For subsets $\pi_1, \ldots, \pi_k$

- $cut(\pi, \bar{\pi}) = \sum_{i \in \pi, j \in \bar{\pi}} w_{ij}$
- $vol(\pi) = \sum_{i \in \pi} d_{ii}$

Minimize normalized cut:

$$NCut(\pi_1, \ldots, \pi_k) = \sum_{i=1}^{k} \frac{cut(\pi_i, \bar{\pi}_i)}{vol(\pi_i)}$$



A bi-partitioned graph

[2]Pasadakis, Alappat, Schenk, and Wellein 2021.

Outline

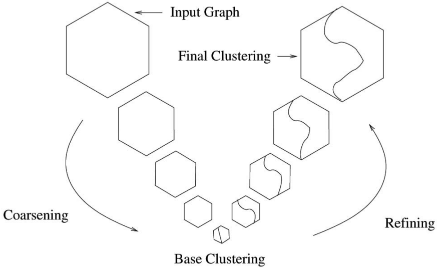
1. Graphs

2. Graclus Framework

3. Clustering Algorithms

4. Results

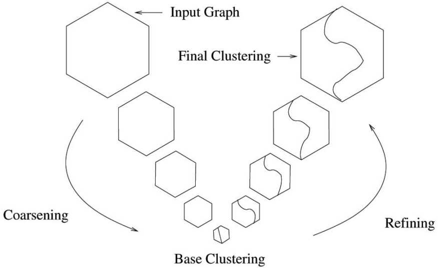
Coarsening and Refinement[3]



[3] Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

Coarsening



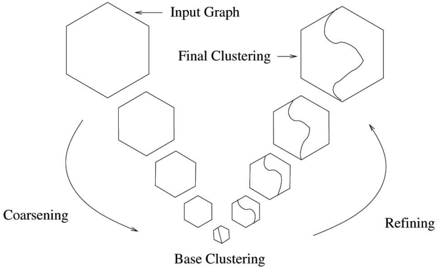
[3] Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

Coarsening

- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed



[3] Dhillon, Guan, and Kulis 2007.

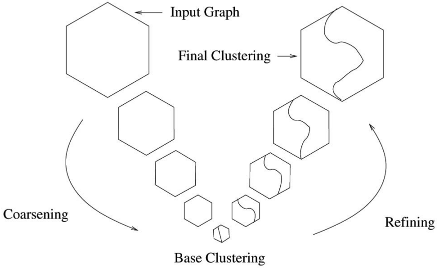
Coarsening and Refinement[3]

Coarsening

- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed

- Obtain successively smaller graphs $\{G, G_{+1}, ..., G_m\}$



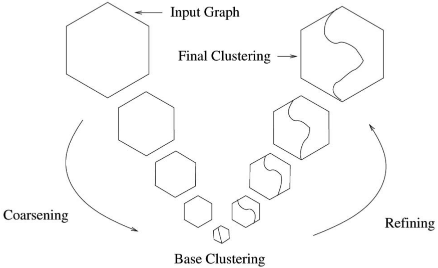
[3]Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

Coarsening

- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed
- Obtain successively smaller graphs $\{G, G_{+1}, ..., G_m\}$
- Stop coarsening when $|\mathcal{V}_m| < c$



[3]Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

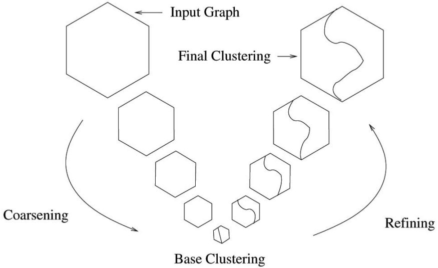
Coarsening

- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed

- Obtain successively smaller graphs $\{G, G_{+1}, ..., G_m\}$
- Stop coarsening when $|\mathcal{V}_m| < c$

Refining



[3] Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

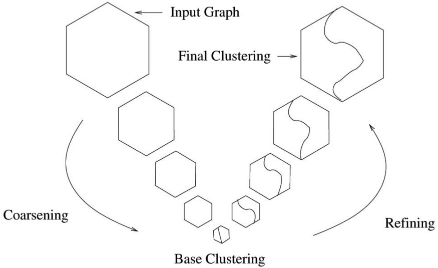
Coarsening

- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed
- Obtain successively smaller graphs $\{G, G_{+1}, ..., G_m\}$
- Stop coarsening when $|\mathcal{V}_m| < c$

Refining

- Finer graphs inherit coarser partitioning



[3] Dhillon, Guan, and Kulis 2007.

Coarsening and Refinement[3]

Coarsening

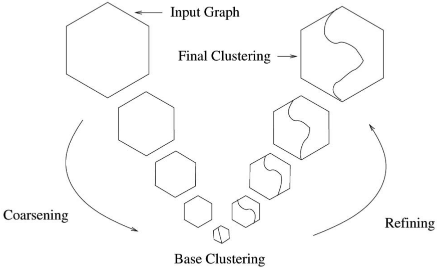
- Visit each vertex and merge with neighbor forming supernode

  $\Rightarrow$ Node degree and edge weights are summed

- Obtain successively smaller graphs $\{G, G_{+1}, ..., G_m\}$
- Stop coarsening when $|\mathcal{V}_m| < c$

Refining

- Finer graphs inherit coarser partitioning
- Refine partitioning using Weighted Kernel K-Means algorithm



[3] Dhillon, Guan, and Kulis 2007.

Weighted Kernel K-Means[4]

- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:

- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} ||x_i - \mu_c||^2$

[4] Dhillon, Guan, and Kulis 2007.

Weighted Kernel K-Means[4]

- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:
  $\Rightarrow$ only works on **linearly separable**
  data

- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} ||x_i - \mu_c||^2$

[4] Dhillon, Guan, and Kulis 2007.

Weighted Kernel K-Means[4]

- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:
  $\Rightarrow$ only works on **linearly separable** data

- **Solution** Project to higher dimension (KKM)

- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} ||\phi(x_i) - \mu_c||^2$

[4] Dhillon, Guan, and Kulis 2007.

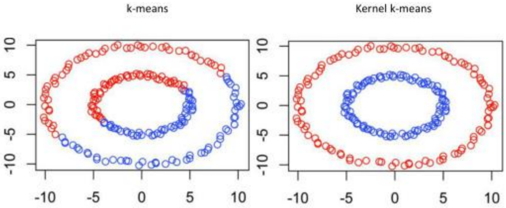
Weighted Kernel K-Means[4]

- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:
  $\Rightarrow$ only works on **linearly separable** data

- **Solution** Project to higher dimension (KKM)

- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} ||\phi(x_i) - \mu_c||^2$



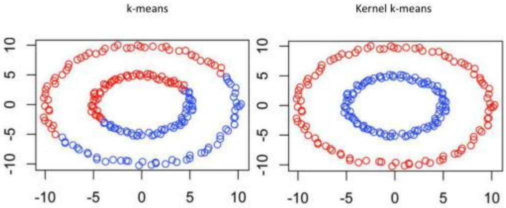
[4]Dhillon, Guan, and Kulis 2007.

Weighted Kernel K-Means[4]

- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:
  $\Rightarrow$ only works on **linearly separable** data

- **Solution** Project to higher dimension (KKM)
- Add weights to vertices to obtain Weighted Kernel K-means (WKKM) objective:
- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} w_i ||\phi(x_i) - \mu_c||^2$



[4] Dhillon, Guan, and Kulis 2007.

Weighted Kernel K-Means[4]

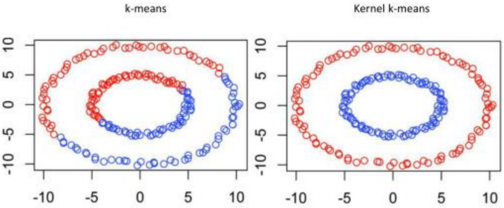
- K-means simple clustering algorithm.

  Minimize sum of distances to centroids:
  $\Rightarrow$ only works on **linearly separable** data

- **Solution** Project to higher dimension (KKM)
- Add weights to vertices to obtain Weighted Kernel K-means (WKKM) objective:
- $\sum_{c=1}^{k} \sum_{x_i \in \pi_c} w_i ||\phi(x_i) - \mu_c||^2$

Node Weight	Kernel
Degree of node	$\sigma D^{-1} + D^{-1} W D^{-1}$

Table: Weights and Kernel for minimising NCut



[4]Dhillon, Guan, and Kulis 2007.

Outline

Direct Mutliway Spectral Clustering[7]

- **Goal**: Partition graph into k parts
- Compute k smallest eigenvectors of graph Laplacian

  $\Rightarrow U \in R^{n*k}$ containing eigenvectors
- Use k-means algorithm to cluster eigenvectors

[6] Inoue, Li, and Kurata 2010.
[6] Pasadakis, Alappat, Schenk, and Wellein 2021.
[7] von Luxburg 2007.

Direct Mutliway Spectral Clustering[7]

- **Goal**: Partition graph into k parts
- Compute k smallest eigenvectors of graph Laplacian

  $\Rightarrow U \in R^{n*k}$ containing eigenvectors
- Use k-means algorithm to cluster eigenvectors

Three Laplacians to study

- Normalized Laplacian: $L_{rw} = I - D^{-1}W$
- β-Laplacian [5]: $L_{\beta} = I - D^{-\beta}W \quad \beta \in [1,2]$
- p-Spectral [6]

[6] Inoue, Li, and Kurata 2010.
[6] Pasadakis, Alappat, Schenk, and Wellein 2021.
[7] von Luxburg 2007.

p-Spectral[8]

- Smallest eigenvector computation:

$$\min_{U \in R^{n*k}} Tr(U^T L U) \text{ subject to } U^T U = I$$

$$\text{with } Tr(U^T L U) = \sum_{l}^{k} \sum_{ij}^{n} \frac{w_{ij}(u_i^l - u_j^l)^2}{2||u^l||_2^2}$$

[8] Pasadakis, Alappat, Schenk, and Wellein 2021.

p-Spectral[8]

- Smallest eigenvector computation:

$$\min_{U \in R^{n*k}} Tr(U^T L U) \text{ subject to } U^T U = I$$

$$\text{with } Tr(U^T L U) = \sum_{l}^{k} \sum_{ij}^{n} \frac{w_{ij}(u_i^l - u_j^l)^2}{2||u^l||_2^2}$$

Unconstrained p-norm optimization on Grassmann

$$\underset{U \in \mathcal{G}r(k,n)}{\text{minimize}} = \sum_{l}^{k} \sum_{ij}^{n} \frac{w_{ij}|u_i^l - u_j^l|^p}{2||u^l||_p^p} \quad p \in [1.1, 2]$$

$$\text{p-norm: } ||u||_p = \sqrt[p]{\sum_{i=1}^{n} |u_i|^p}$$

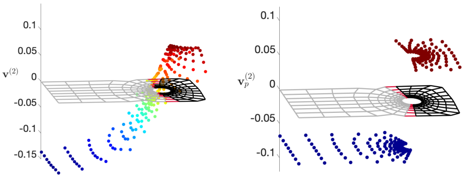
[8] Pasadakis, Alappat, Schenk, and Wellein 2021.

p-Spectral[8]

- Smallest eigenvector computation:

$$\min_{U \in R^{n*k}} Tr(U^T L U) \text{ subject to } U^T U = I$$

$$\text{with } Tr(U^T L U) = \sum_{l}^{k} \sum_{ij}^{n} \frac{w_{ij}(u_i^l - u_j^l)^2}{2||u^l||_2^2}$$



Fiedler vector for p=2 (left) p = 1.1 (right)

Unconstrained p-norm optimization on Grassmann

$$\underset{U \in \mathcal{G}r(k,n)}{\text{minimize}} = \sum_{l}^{k} \sum_{ij}^{n} \frac{w_{ij}|u_i^l - u_j^l|^p}{2||u^l||_p^p} \quad p \in [1.1, 2]$$

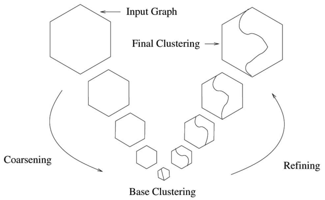
$$\text{p-norm: } ||u||_p = \sqrt[p]{\sum_{i=1}^{n} |u_i|^p}$$

[8] Pasadakis, Alappat, Schenk, and Wellein 2021.

Algorithm: Multilevel spectral clustering algorithm
Input: Adjacency matrix $W \in R^{n*n}$, number k of clusters to construct

1) **Coarsen** graph to produce successively smaller adjacency matrices $\{W, W_1, ..., W_m\}$
   - Merge vertices together, sum up node degree and edge weights
   - Stop once $|V_m| < 5k$
2) Run **base clustering**
   - Compute first k eigenvectors using (RW, Beta, p)-Laplacian
   - Cluster eigenvectors using k-means algorithm
3) **Refine** partition to initial graph
   - Finer graph inherits partitioning of lower level graph
   - Use as initial guess for WKKM

Output: Clusters $A_1, ..., A_k$



Outline

1. Graphs
2. Graclus Framework
3. Clustering Algorithms
4. Results

Metrics

Internal Metrics

- Normalized Cut (**minimize**):

$$\sum_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

External Metrics

- Accuracy (**maximize**)

Simulation Parameters

- Averaged over 10 runs
- Coarsen until $|\mathcal{V}_m| < 5k$
- K-means algorithm run 30 times
- Best beta chosen from 11 values $\in [1, 2]$
- p-Spectral method uses **normalized eigenvectors** as initial guess.
- Best p chosen from 8 values $\in [1.1, 2]$

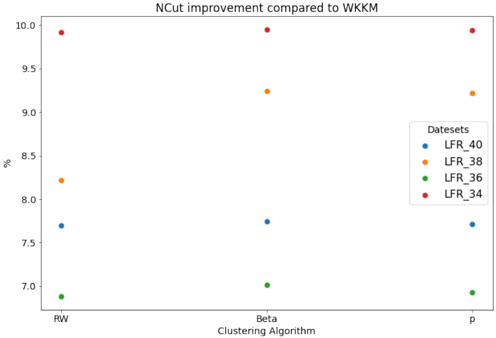
LFR Dataset

Stochastic block model with 10 clusters.

- Increasing **noise**

Coarse Level

Less than **50 vertices**



Original Graph

Metric	RW (%)	Beta (%)	p (%)
NCut	-1.71	-1.45	-1.48
Acc	+1.35	+1.33	+1.41
Runtime (s)	0.2	1.7	55.1

Table: Benchmark of LFR_40 dataset at finest level (**1000 vertices**)

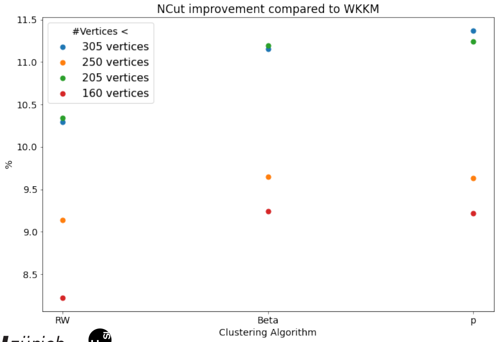
Gauss Dataset

Dataset containing k clusters with 400 points each, sampled from **Gaussian** distribution.
k $\in \{32, 41, 50, 61\}$

- Increasing **size**

Coarse Level

Less than **5k vertices**



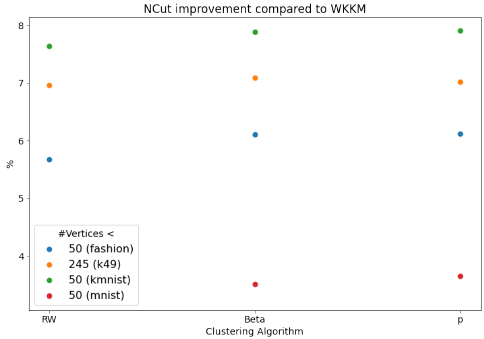
Original Graph

Metric	RW (%)	Beta (%)	p (%)
NCut	0.0	-0.79	+0.03
Acc	+0.06	+1.62	-0.61
Runtime (s)	1.0	7.2	219.7

Table: Benchmark of largest Gauss dataset at finest level (**24400 vertices**)

Real-World Data

Coarse Level



Original Graph

Metric	RW (%)	Beta (%)	p (%)
NCut	-3.58	-5.12	-4.66
Acc	-2.94	-1.38	-1.02
Runtime (s)	0.2	1.0	46.4

Table: Benchmark of Fashion dataset at finest level (**10000 vertices**)

Future Work

- Rewrite Graclus code
- Improve refinement algorithm
- Study the combination of β-Laplacian and p-Spectral

Thank you for your attention!

References

- Dhillon, Inderjit S., Yuqiang Guan, and Brian Kulis (Nov. 2007). "Weighted Graph Cuts without Eigenvectors A Multilevel Approach". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.11, pp. 1944–1957. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2007.1115.
- Inoue, Kentaro, Weijiang Li, and Hiroyuki Kurata (Sept. 2010). "Diffusion Model Based Spectral Clustering for Protein-Protein Interaction Networks". In: *PLoS ONE* 5.9. Ed. by Ingemar T. Ernberg, e12623. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0012623.
- Pasadakis, Dimosthenis, Christie Louis Alappat, Olaf Schenk, and Gerhard Wellein (Nov. 2021). "Multiway P-Spectral Graph Cuts on Grassmann Manifolds". In: *Machine Learning*. ISSN: 1573-0565. DOI: 10.1007/s10994-021-06108-1.
- von Luxburg, Ulrike (Nov. 2007). "A Tutorial on Spectral Clustering". In: *arXiv:0711.0189 [cs]*. arXiv: 0711.0189 [cs].

LFR Original

Metric	RW %	Beta %	p %	p+beta %
NCut	-1.71	-1.45	-1.48	-1.40
Acc	+1.35	+1.33	+1.41	+1.34
NMI	+1.40	+1.09	+1.21	+1.10

Table: Comparison of Fashion dataset at original graph

Gauss Original

Metric	RW (%)	Beta (%)	p (%)
NCut	0.0	-0.79	+0.03
Acc	+0.06	+1.62	-0.61
NMI	-0.02	+0.47	-0.18
Runtime (s)	1.0	7.2	219.7

Table: Comparison of Gauss_61_55_10NN dataset at original graph with 24400 vertices

Real-World Ncut Coarse

Dataset	RW %	Beta %	p %	p+beta %
Fashion	5.68	6.11	6.12	6.22
k49	6.96	7.09	7.02	7.11
kmnist	7.64	7.89	7.91	7.91
mnist	3.29	3.52	3.65	3.79

Table: Percentage of improvement in NCut when compared to WKKM

Real-World Metrics Original

Metric	RW %	Beta %	p %	p+beta %
NCut	3.58	5.12	4.66	4.01
Acc	-2.94	-1.38	-1.02	-0.93
NMI	-0.79	-0.04	+0.57	+0.08

Table: Comparison of Fashion dataset at original graph

Random Walk Laplacian

Diffusion Laplacian[9]

$$L = I - D^{-\beta}W$$

with beta taking values between 1 and 2.

[9] Inoue, Li, and Kurata 2010.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s): **First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date **Signature(s)**

Kalman Szenes

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.