Clustering Phenomena in Large Networks and Reducible Matrices.

An oral exam presented in partial fulfillment of the requirements of the PhD in mathematics at the University of Georgia

Daniel Mckenzie

6th June 2016

Table of Contents

The Research Questions

Question 1

Given $A \in \mathbb{R}^{n \times n}$, symmetric and non-negative, does there exist a **permutation matrix** *P such that PAP^T is block diagonal? That is:*

$$PAP^T = B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 \\ 0 & B_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix} \tag{1}$$

If such a P exists, can we find it?

The Research Questions

Question 1

Given $A \in \mathbb{R}^{n \times n}$, symmetric and non-negative, does there exist a **permutation matrix** *P such that PAP^T is block diagonal? That is:*

$$PAP^T = B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 \\ 0 & B_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix} \tag{1}$$

If such a P exists, can we find it?

Answer 1

Yes and yes. Exist $\mathcal{O}(n)$, graph-based methods.

The Research Questions

Question 2

Given $A \in \mathbb{R}^{n\times n}$, symmetric and non-negative, can we find a permutation matrix P such that

$$PAP^T = B + E$$

with B block diagonal and symmetric as in (1) and E is a small (symmetric) perturbation? ($||E|| << ||B||$)

The Research Questions

Question 2

Given $A \in \mathbb{R}^{n \times n}$, symmetric and non-negative, can we find a permutation matrix P such that

$$PAP^T = B + E$$

with B block diagonal and symmetric as in (1) and E is a small (symmetric) perturbation? ($||E|| << ||B||$)

Answer 2

Yes, but existing methods are slow, typically $\mathcal{O}(n^3)$.

The Research Questions

Question 3

Suppose $A \in \mathbb{R}^{n\times n}$ is no longer symmetric. Can we find a permutation matrix P such that

$$PAP^T = B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1,r} \\ 0 & B_{22} & \cdots & B_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix},$$

Or $PAP^T = B + E$ *with* $||E|| << ||B||$

The Research Questions

Question 4

What if A is no longer square? Say $A \in \mathbb{R}^{m \times n}$ with $m < n$, can we find permutation matrices $P \in \mathbb{R}^{m \times m}$ and $Q \in \mathbb{R}^{n \times n}$ such that $PAQ = B + E$ with

$$B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_{rr} & 0 \end{bmatrix}$$

and $||E|| << ||B||$.

The Research Questions

Question 5

Why should we care?

Motivating Example 1: Large Networks

- Prevalent in many areas of science, social science and engineering ([1]).

[1] Lovasz, "Very large graphs".

Motivating Example 1: Large Networks

- Prevalent in many areas of science, social science and engineering ([1]).
- Suppose graph G has vertices V representing users of a social network (eg. Facebook, LinkedIn) and edges between connected users.

[1] Lovasz, "Very large graphs".

Motivating Example 1: Large Networks

- Prevalent in many areas of science, social science and engineering ([1]).
- Suppose graph G has vertices V representing users of a social network (eg. Facebook, LinkedIn) and edges between connected users.
- Sets of nodes with high interconnectivity (communities/ clusters) could represent friendship groups/ co-workers.

[1] Lovasz, "Very large graphs".

Motivating Example 1: Large Networks

- Prevalent in many areas of science, social science and engineering ([1]).
- Suppose graph G has vertices V representing users of a social network (eg. Facebook, LinkedIn) and edges between connected users.
- Sets of nodes with high interconnectivity (communities/ clusters) could represent friendship groups/ co-workers.
- By identifying communities can suggest new connections to users.

[1] Lovasz, "Very large graphs".

Motivating Example 1: Large Networks

- Prevalent in many areas of science, social science and engineering ([1]).
- Suppose graph G has vertices V representing users of a social network (eg. Facebook, LinkedIn) and edges between connected users.
- Sets of nodes with high interconnectivity (communities/ clusters) could represent friendship groups/ co-workers.
- By identifying communities can suggest new connections to users.
- Some networks are **directed** (eg. Twitter, Citation Networks) making community detection more subtle.

[1] Lovasz, "Very large graphs".

Motivating Example 1: Large Networks

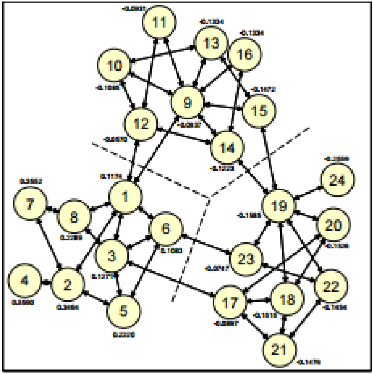


Figure: Graph with vertices ordered into clusters. (from Bertrand and Moonen, "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks")

Motivating Example 1: Large Networks

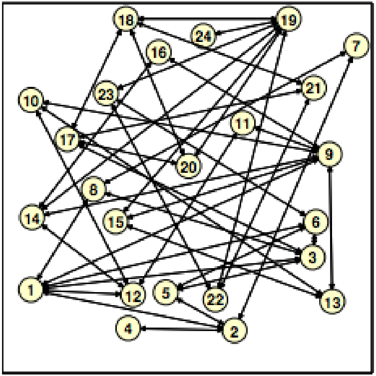


Figure: Same graph with vertices randomly positioned. (from Bertrand and Moonen, "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks")

Motivating Example 2: Unsupervised Learning[2,3]

- Suppose given a large, high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$.

[2] Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm".

[3] Von Luxburg, "A Tutorial on Spectral Clustering".

Motivating Example 2: Unsupervised Learning[2][3]

- Suppose given a large, high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$.
- Want to sort into k clusters $C_1, \ldots, C_k$ of similar data points.

[2] Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm".

[3] Von Luxburg, "A Tutorial on Spectral Clustering".

Motivating Example 2: Unsupervised Learning[2,3]

- Suppose given a large, high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$.
- Want to sort into k clusters $C_1, \ldots, C_k$ of similar data points.
- Many standard approaches (eg. k-means) are only capable of detecting clusters with convex polyhedral boundaries.

[2] Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm".

[3] Von Luxburg, "A Tutorial on Spectral Clustering".

Motivating Example 2: Unsupervised Learning[2,3]

- Suppose given a large, high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$.
- Want to sort into k clusters $C_1, \ldots, C_k$ of similar data points.
- Many standard approaches (eg. k-means) are only capable of detecting clusters with convex polyhedral boundaries.
- Another approach: Form a graph G with vertices $\{1, \ldots, n\}$ and an edge between i and j iff $d(\mathbf{x}_i, \mathbf{x}_j) < \epsilon$

[2] Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm".

[3] Von Luxburg, "A Tutorial on Spectral Clustering".

Motivating Example 2: Unsupervised Learning[2][3]

- Suppose given a large, high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$.
- Want to sort into k clusters $C_1, \ldots, C_k$ of similar data points.
- Many standard approaches (eg. k-means) are only capable of detecting clusters with convex polyhedral boundaries.
- Another approach: Form a graph G with vertices $\{1, \ldots, n\}$ and an edge between i and j iff $d(\mathbf{x}_i, \mathbf{x}_j) < \epsilon$
- Find clusters in G, reducing this problem to previous one.

[2] Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm".

[3] Von Luxburg, "A Tutorial on Spectral Clustering".

Motivating Example 2: Unsupervised Learning [4]

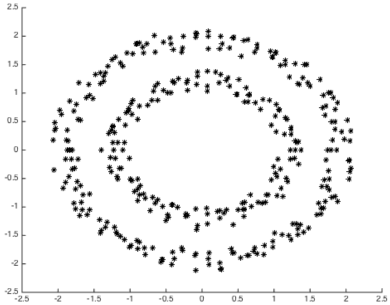


Figure: An artificial data set which appears to have two distinct clusters

[4]see (Jain, "Data clustering: 50 years beyond K-means")

Motivating Example 2: Unsupervised Learning

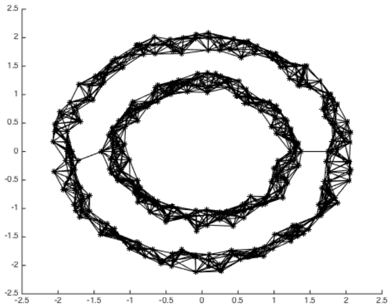
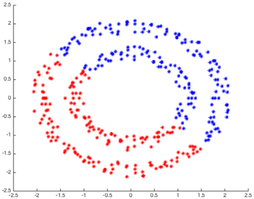
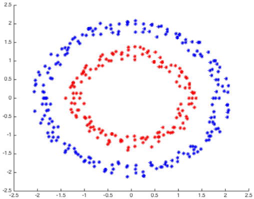


Figure: unweighted graph constructed by connecting data points which are close

Motivating Example 2: Unsupervised Learning



(a) k-means clustering



(b) Graph-based clustering

Figure: Comparison of clustering methods.

Motivating Example 2: Unsupervised Learning

We will relate detecting clusters in graphs to ‘almost block-diagonalizing’ symmetric matrices by permutation matrices (question 2) in section 2

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.
- Would like to find which words occur frequently together in documents, and which documents are similar.

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.
- Would like to find which words occur frequently together in documents, and which documents are similar.
- We can:

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.
- Would like to find which words occur frequently together in documents, and which documents are similar.
- We can:
- Choose n keywords.

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.
- Would like to find which words occur frequently together in documents, and which documents are similar.
- We can:
- Choose n keywords.
- Assign i-th document a vector $\mathbf{x}_i \in \mathbb{R}^n$ where $x_{ij} = \#$ times j-th word occurs in i-th document.

Motivating Example 3: Natural Language Processing

- Suppose we have m documents we need to analyse.
- Would like to find which words occur frequently together in documents, and which documents are similar.
- We can:
- Choose n keywords.
- Assign i-th document a vector $\mathbf{x}_i \in \mathbb{R}^n$ where $x_{ij} = \#$ times j-th word occurs in i-th document.
- Form matrix $A \in \mathbb{R}^{n \times m}$ with columns $\mathbf{x}_i$.

Motivating Example 3: Natural Language Processing[5]

- Find permutation matrices P and Q such that $PAQ = B + E$ with $||E|| << ||B||$ and:

$$B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_{rr} & 0 \end{bmatrix}$$

[5] see: Dhillon, "Co-clustering documents and words using Bipartite spectral graph partitioning".

Motivating Example 3: Natural Language Processing[5]

- Find permutation matrices P and Q such that $PAQ = B + E$ with $||E|| << ||B||$ and:

$$B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_{rr} & 0 \end{bmatrix}$$

- Rows represent words. Hence rows of B_{ii} represent words which frequently occur together.

[5] see: Dhillon, "Co-clustering documents and words using Bipartite spectral graph partitioning".

Motivating Example 3: Natural Language Processing[5]

- Find permutation matrices P and Q such that $PAQ = B + E$ with $||E|| << ||B||$ and:

$$B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_{rr} & 0 \end{bmatrix}$$

- Rows represent words. Hence rows of B_{ii} represent words which frequently occur together.
- Columns represent documents. Hence columns of B_{ii} represent documents with similar patterns of occurrences of words.

[5] see: Dhillon, "Co-clustering documents and words using Bipartite spectral graph partitioning".

Motivating Example 3: Natural Language Processing[5]

- Find permutation matrices P and Q such that $PAQ = B + E$ with $||E|| << ||B||$ and:

$$B = \begin{bmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_{rr} & 0 \end{bmatrix}$$

- Rows represent words. Hence rows of B_{ii} represent words which frequently occur together.
- Columns represent documents. Hence columns of B_{ii} represent documents with similar patterns of occurrences of words.
- Have solved two problems simultaneously (co-clustering).

[5] see: Dhillon, "Co-clustering documents and words using Bipartite spectral graph partitioning".

Table of Contents

Reducible matrices

Definition

A matrix A is said to be **reducible** *if there exists a permutation matrix P such that:*

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} \quad (2)$$

and **irreducible** *if no such P exists. We say PAP^T is* **reduced**.

Reducible Matrices

Definition

If there exists a $k > 1$ and permutation matrix P such that:

$$PAP^{\top} = B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1,r} \\ 0 & B_{22} & \cdots & B_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix}, \qquad (3)$$

where each B_{ii} is a square matrix and is either irreducible or 0*, we say A is* **k-reducible** *and B is* **k-reduced**.

Reducible Matrices

Remark

If A is symmetric and reducible it can be **block diagonalized** *by a permutation matrix:*

$$PAP^T = \begin{bmatrix} B_{11} & 0 & \cdots & 0 \\ 0 & B_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix},$$

with each B_{ii} square and symmetric.

Almost Reducible Matrices

Definition

A is **almost reducible**[6] *if there exists a permutation P such that*

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \tag{4}$$

with $||B_{21}|| << ||B_{11}||$ *and* $||B_{21}|| << ||B_{22}||$.

[6] Bolt and Santitissadeekorn, *Applied and Computational Measurable Dynamics*.

Almost Reducible Matrices

Definition

A is **almost reducible**[6] *if there exists a permutation P such that*

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \quad (4)$$

with $||B_{21}|| << ||B_{11}||$ and $||B_{21}|| << ||B_{22}||$.
Equivalently:

$$PAP^T = \tilde{B} + E$$

with $\tilde{B}$ reduced and $||E|| << ||\tilde{B}||$.

[6] Bolt and Santitissadeekorn, *Applied and Computational Measurable Dynamics*.

Almost Reducible Matrices

Definition

A is **almost reducible**[6] *if there exists a permutation P such that*

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \tag{4}$$

with $||B_{21}|| << ||B_{11}||$ and $||B_{21}|| << ||B_{22}||$.
Equivalently:

$$PAP^T = \tilde{B} + E$$

with $\tilde{B}$ reduced and $||E|| << ||\tilde{B}||$.
Equivalently:

$$A = \tilde{A} + D$$

with $\tilde{A}$ reducible and $||D|| << ||\tilde{A}||$.

[6] Bollt and Santitissadeekorn, *Applied and Computational Measurable Dynamics*.

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced**

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced** *Can extend this straightforwardly to notion of* **k-almost-reducibility**

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced** *Can extend this straightforwardly to notion of* **k-almost-reducibility**

Suppose A is symmetric and k-almost-reducible:

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced** *Can extend this straightforwardly to notion of* **k-almost-reducibility**

Suppose A is symmetric and k-almost-reducible:

- Then $PAP^T = \tilde{B} + E$ with $\tilde{B}$ and E symmetric, $\tilde{B}$ k-reduced and $||E|| << ||\tilde{B}||$.

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced** *Can extend this straightforwardly to notion of* **k-almost-reducibility**

Suppose A is symmetric and k-almost-reducible:

- Then $PAP^T = \tilde{B} + E$ with $\tilde{B}$ and E symmetric, $\tilde{B}$ k-reduced and $||E|| << ||\tilde{B}||$.
- Equivalently $A = \tilde{A} + D$ with $\tilde{A}$ and D symmetric, $\tilde{A}$ k-reducible and $||D|| << ||\tilde{A}||$.

Almost Reducible Matrices

Definition

(cont.) We say PAP^T is **almost reduced** *Can extend this straightforwardly to notion of* **k-almost-reducibility**

Suppose A is symmetric and k-almost-reducible:

- Then $PAP^T = \tilde{B} + E$ with $\tilde{B}$ and E symmetric, $\tilde{B}$ k-reduced and $||E|| << ||\tilde{B}||$.
- Equivalently $A = \tilde{A} + D$ with $\tilde{A}$ and D symmetric, $\tilde{A}$ k-reducible and $||D|| << ||\tilde{A}||$.
- Thus A is a perturbation of a symmetric k-reducible matrix by a ‘small’ symmetric matrix.

Graph Theory[7]

Definition

A **graph** *G is a set of vertices V together with a subset $E \subset V \times V$ of edges. G is:*

- **Weighted** *if every edge $e \in E$ has a weight $a_e \in \mathbb{R}_{>0}$*
- **Directed** *if we distinguish between an edge from u to v or from v to u.*

[7] Bollobas, *Graph theory: an introductory course.*

Graph Theory[7]

Definition

A **graph** *G is a set of vertices V together with a subset $E \subset V \times V$ of edges. G is:*

- **Weighted** *if every edge $e \in E$ has a weight $a_e \in \mathbb{R}_{>0}$*
- **Directed** *if we distinguish between an edge from u to v or from v to u.*

Remark

Some Conventions:

- *Henceforth shall assume all graphs weighted and undirected.*
- *If $|V| = n < \infty$ we shall identify V with $\{1, 2, \ldots, n\}$.*

[7] Bollobas, *Graph theory: an introductory course.*

Graph Theory

Definition

The **degree** *of a vertex i is $d_i = \sum_{i \in e} a_e$.*

Graph Theory

Definition

The **degree** *of a vertex i is $d_i = \sum_{i\in e} a_e$.*

Definition

A **path** *from i to j is a collection of edges $\{i_0, i_1\}, \{i_1, i_2\}, \ldots, \{i_{k-1}, i_k\}$ with $i = i_0$ and $i_k = j$.*

Graph Theory

Definition

*The **degree** of a vertex i is $d_i = \sum_{i \in e} a_e$.*

Definition

*A **path** from i to j is a collection of edges $\{i_0, i_1\}, \{i_1, i_2\}, \ldots, \{i_{k-1}, i_k\}$ with $i = i_0$ and $i_k = j$.*

Definition

*G is **connected** if given any $i, j \in V$ there exists a path between them. Otherwise, G is **disconnected**.*

Clusters in Graphs

Definition

A **k-clustering** *of G is a partition $\pi = \{C_1, \ldots, C_k\}$ of V into k subsets with 'many' edges between vertices in C_a for any a, and 'few' edges between vertices in C_a and vertices in C_b for $a \neq b$.*

[8] Nascimento and De Carvalho, "Spectral methods for graph clustering - A survey".

Clusters in Graphs

Definition

A **k-clustering** *of G is a partition $\pi = \{C_1, \ldots, C_k\}$ of V into k subsets with 'many' edges between vertices in C_a for any a, and 'few' edges between vertices in C_a and vertices in C_b for $a \neq b$.*

We shall measure quality of a clustering π using ratiocut:

[8] Nascimento and De Carvalho, "Spectral methods for graph clustering - A survey".

Clusters in Graphs

Definition

A **k-clustering** *of G is a partition $\pi = \{C_1, \ldots, C_k\}$ of V into k subsets with 'many' edges between vertices in C_a for any a, and 'few' edges between vertices in C_a and vertices in C_b for $a \neq b$.*

We shall measure quality of a clustering π using ratiocut:

Definition

$$Rcut(\pi) = \frac{1}{2}\sum_{a=1}^{k}\frac{W(C_a, \bar{C}_a)}{|C_a|} \quad \text{where } W(C_a, \bar{C}_a) = \sum_{i \in C_a, j \notin C_a} a_{ij}$$

[8] Nascimento and De Carvalho, "Spectral methods for graph clustering - A survey".

Clusters in Graphs

Definition

A **k-clustering** *of G is a partition $\pi = \{C_1, \ldots, C_k\}$ of V into k subsets with 'many' edges between vertices in C_a for any a, and 'few' edges between vertices in C_a and vertices in C_b for $a \neq b$.*

We shall measure quality of a clustering π using ratiocut:

Definition

$$Rcut(\pi) = \frac{1}{2}\sum_{a=1}^{k}\frac{W(C_a, \bar{C}_a)}{|C_a|} \text{ where } W(C_a, \bar{C}_a) = \sum_{i\in C_a, j\notin C_a} a_{ij}$$

Although there are (many) other measures ([8]).

[8] Nascimento and De Carvalho, "Spectral methods for graph clustering - A survey".

Clusters in Graphs

We say π is a good clustering if $\text{Rcut}(\pi)$ is 'small'.

Clusters in Graphs

We say π is a good clustering if Rcut(π) is 'small'.

Definition (The Clustering Problem)

Given a graph G, find π such that:

$$Rcut(\pi) = \min\{Rcut(\sigma) : \ \sigma \text{ a partition of } V\} \tag{5}$$

Clusters in Graphs

We say π is a good clustering if $\mathrm{Rcut}(\pi)$ is 'small'.

Definition (The Clustering Problem)

Given a graph G, find π such that:

$$Rcut(\pi) = \min\{Rcut(\sigma) : \ \sigma \text{ a partition of } V\} \tag{5}$$

Remark

An ideal clustering would be k connected components, as this would have $Rcut = 0$. Thus finding clusters is a 'perturbation' of the problem of finding connected components.

Graphs and Associated Matrices

Definition

Given a graph G, its **Adjacency Matrix** *A is a non-negative $n \times n$ matrix defined by:*

$$A_{ij} = \begin{cases} a_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

Graphs and Associated Matrices

Definition

Given a graph G, its **Adjacency Matrix** *A is a non-negative $n \times n$ matrix defined by:*

$$A_{ij} = \begin{cases} a_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

A is **symmetric** if G is **undirected**, **binary** if G is **unweighted**.

Graphs and Associated Matrices

Definition

Given a graph G, its **Adjacency Matrix** *A is a non-negative $n \times n$ matrix defined by:*

$$A_{ij} = \begin{cases} a_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

A is **symmetric** if G is **undirected**, **binary** if G is **unweighted**.

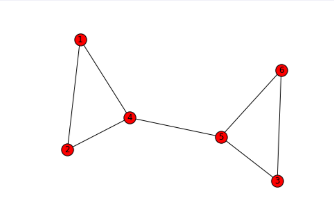
Remark

Given any permutation matrix P, $B = PAP^T$ is adjacency matrix of same graph G, just with labels of vertices permuted.

Graphs and Associated Matrices

Example

Consider the following graph G:



Which has adjacency matrix:

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

Graphs and Associated Matrices

Example

G clearly has two clusters. Moreover if P is the permutation matrix swapping the third and fourth rows (so P^T swaps the third and fourth columns) then:

$$PAP^T = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Which is 2-almost-reduced. (But note there are other P such that PAP^T is 2-almost-reduced.)

Graphs and Associated Matrices

Theorem

G has k connected components iff A is k reducible.
G has a good k clustering iff $A = \tilde{A} + E$ is k-almost-reducible and $||E||$ is small .

Graphs and Associated Matrices

Theorem

G has k connected components iff A is k reducible.
G has a good k clustering iff $A = \tilde{A} + E$ is k-almost-reducible and $||E||$ is small .

Given P can determine clustering π, and conversely given π can determine a P , but not uniquely.

Graphs and Associated Matrices

Theorem

G has k connected components iff A is k reducible.
G has a good k clustering iff $A = \tilde{A} + E$ is k-almost-reducible and $||E||$ is small .

Given P can determine clustering π, and conversely given π can determine a P , but not uniquely.

Remark

Can associate a graph G to any symmetric, non-negative matrix A, thus question 1 and 2 are equivalent to graph-theoretic problems.

Graphs and Associated Matrices

Definition

Let A be the adjacency matrix of an undirected graph G.
Let $D = diag(d_1, \ldots, d_n)$.
The (unnormalized) **Graph Laplacian** *is defined as:*

$$L = D - A$$

Graphs and Associated Matrices

Definition

Let A be the adjacency matrix of an undirected graph G.
Let $D = diag(d_1, \ldots, d_n)$.
The (unnormalized) **Graph Laplacian** *is defined as:*

$$L = D - A$$

Note that $d_i = \sum_j a_{ij}$ and L can be defined for any symmetric non-negative matrix A.

Graphs and Associated Matrices

Proposition

For any undirected graph G / symmetric non-negative matrix A:

Graphs and Associated Matrices

Proposition

For any undirected graph G / symmetric non-negative matrix A:

1. $\forall \mathbf{x} \in \mathbb{R}^n, \ \mathbf{x}^\top L\mathbf{x} = \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2.$

Graphs and Associated Matrices

Proposition

For any undirected graph G / symmetric non-negative matrix A:

1. $\forall \mathbf{x} \in \mathbb{R}^n,\ \mathbf{x}^\top L\mathbf{x} = \dfrac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2.$
2. *L is symmetric and positive semi-definite.*

Graphs and Associated Matrices

Proposition

For any undirected graph G / symmetric non-negative matrix A:

1. $\forall \mathbf{x} \in \mathbb{R}^n,\ \mathbf{x}^\top L\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2.$
2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is $\lambda_1 = 0$ with $\mathbf{1}$ as an eigenvector.*

Graphs and Associated Matrices

Proposition

For any undirected graph G / symmetric non-negative matrix A:

1. $\forall \mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top L\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2$.
2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is $\lambda_1 = 0$ with $\mathbf{1}$ as an eigenvector.*
4. *$mult(0) = \#$ connected components of G or $\#$blocks in reduced form of A*

Graphs and Associated Matrices

Proposition

For any undirected graph G/ symmetric non-negative matrix A:

1. $\forall \mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top L \mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_i - x_j)^2$.
2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is $\lambda_1 = 0$ with $\mathbf{1}$ as an eigenvector.*
4. *mult$(0) = \#$ connected components of G or $\#$blocks in reduced form of A*

Proof.

(See Chung, *Spectral graph theory*) □

Graphs and Associated Matrices

Convention: λ_1 will always denote the *smallest* eigenvalue of a matrix.

Graphs and Associated Matrices

Convention: λ_1 will always denote the *smallest* eigenvalue of a matrix.

Theorem

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and non-negative. Let $L = D - A$, then PAP^T is k-reduced iff PLP^T is k-reduced and PAP^T is k-almost-reduced iff PLP^T is k-almost-reduced.

Graphs and Associated Matrices

Convention: λ_1 will always denote the *smallest* eigenvalue of a matrix.

Theorem

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and non-negative. Let $L = D - A$, then PAP^T is k-reduced iff PLP^T is k-reduced and PAP^T is k-almost-reduced iff PLP^T is k-almost-reduced.

Proof.

1. Observe that PDP^T is diagonal for any permutation matrix P.

Graphs and Associated Matrices

Convention: λ_1 will always denote the *smallest* eigenvalue of a matrix.

Theorem

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and non-negative. Let $L = D - A$, then PAP^T is k-reduced iff PLP^T is k-reduced and PAP^T is k-almost-reduced iff PLP^T is k-almost-reduced.

Proof.

1. Observe that PDP^T is diagonal for any permutation matrix P.
2. Hence $PLP^T = PDP^T - PAP^T$ is k-reduced iff A is k-reduced.

Graphs and Associated Matrices

Convention: λ_1 will always denote the *smallest* eigenvalue of a matrix.

Theorem

Let $A \in \mathbb{R}^{n \times n}$ be symmetric and non-negative. Let $L = D - A$, then PAP^T is k-reduced iff PLP^T is k-reduced and PAP^T is k-almost-reduced iff PLP^T is k-almost-reduced.

Proof.

1. Observe that PDP^T is diagonal for any permutation matrix P.
2. Hence $PLP^T = PDP^T - PAP^T$ is k-reduced iff A is k-reduced.
3. Almost reduced case is similar.

□

Summary

- A is k-reducible iff G has k connected components, A is k-almost reducible iff G has a good k-clustering.

Summary

- A is k-reducible iff G has k connected components, A is k-almost reducible iff G has a good k-clustering.
- Finding a k-clustering π for G equivalent to finding P such that PAP^T is k-almost reduced.

Summary

- A is k-reducible iff G has k connected components, A is k-almost reducible iff G has a good k-clustering.
- Finding a k-clustering π for G equivalent to finding P such that PAP^T is k-almost reduced.
- Easier in practice to work with L than A, and A is (almost-) reduced iff L is (almost-) reduced.

Table of Contents

Indicator Vectors

Definition

Suppose $|V| = n$*. For any* $C \subset V$*, define the* **indicator vector** $\mathbf{1}_C \in \mathbb{R}^n$ *as:*

$$(\mathbf{1}_C)_i = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases}$$

Indicator Vectors

Definition

Suppose $|V| = n$. For any $C \subset V$, define the **indicator vector** $\mathbf{1}_C \in \mathbb{R}^n$ *as:*

$$(\mathbf{1}_C)_i = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases}$$

Theorem

Suppose G has k connected components, with vertex sets $C_1, \ldots, C_k$. Then $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ form a basis for W_0, the 0-eigenspace/kernel of L

Indicator Vectors

Definition

Suppose $|V| = n$. For any $C \subset V$, define the **indicator vector** $\mathbf{1}_C \in \mathbb{R}^n$ *as:*

$$(\mathbf{1}_C)_i = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases}$$

Theorem

Suppose G has k connected components, with vertex sets $C_1, \ldots, C_k$. Then $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ form a basis for W_0, the 0-eigenspace/kernel of L

Proof.

(See Von Luxburg, "A Tutorial on Spectral Clustering", pg. 4) □

Spectral Methods for Connectivity

Given $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ can easily determine $C_1, \ldots, C_k$, so does finding first k eigenvectors of L provide another solution to question 1?

Spectral Methods for Connectivity

Given $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ can easily determine $C_1, \ldots, C_k$, so does finding first k eigenvectors of L provide another solution to question 1?

Problem

If $\dim(W_0) > 1$ *it has infinitely many orthonormal bases, hence with probability* 0 *will an eigenvector finding routine return eigenvectors* $\{\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}\}$.

Spectral Methods for Connectivity

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.

Spectral Methods for Connectivity

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n \times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.
- *Suppose* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *is any other orthogonal basis for* W_0.

Spectral Methods for Connectivity

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n \times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.
- *Suppose* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *is any other orthogonal basis for* W_0.
- *Let* $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k] \in \mathbb{R}^{n \times k}$. *Let* $\mathbf{w}^i \in \mathbb{R}^k$ *denote* i*-th* **row** *of* X.

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n \times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.
- *Suppose* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *is any other orthogonal basis for* W_0.
- *Let* $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k] \in \mathbb{R}^{n \times k}$. *Let* $\mathbf{w}^i \in \mathbb{R}^k$ *denote* i*-th* **row** *of* X.
- $X = VU$ *for some* $k \times k$ *invertible matrix* U, *so* $\mathbf{w}^i = \mathbf{e}_a U$ *iff* $i \in C_a$.

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.
- *Suppose* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *is any other orthogonal basis for* W_0.
- *Let* $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{w}^i \in \mathbb{R}^k$ *denote* i*-th* **row** *of* X.
- $X = VU$ *for some* $k \times k$ *invertible matrix* U, *so* $\mathbf{w}^i = \mathbf{e}_a U$ *iff* $i \in C_a$.
- *So, group* $\mathbf{w}^i$ *into* k *groups* $R_1, \ldots R_k$ *such that all vectors in same group are equal.*

Solution

- *Let* $V = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{r}^i \in \mathbb{R}^k$ *denotes the* i*-th* **row** *of* V.
- *if* $\mathbf{e}_a \in \mathbb{R}^k$ *has* 1 *in* a*-th position,* 0 *elsewhere, claim that* $\mathbf{r}^i = \mathbf{e}_a$ *iff* $i \in C_a$.
- *Suppose* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *is any other orthogonal basis for* W_0.
- *Let* $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k] \in \mathbb{R}^{n\times k}$. *Let* $\mathbf{w}^i \in \mathbb{R}^k$ *denote* i*-th* **row** *of* X.
- $X = VU$ *for some* $k \times k$ *invertible matrix* U, *so* $\mathbf{w}^i = \mathbf{e}_a U$ *iff* $i \in C_a$.
- *So, group* $\mathbf{w}^i$ *into* k *groups* $R_1, \ldots R_k$ *such that all vectors in same group are equal.*
- *Let* $C_a = \{i : \mathbf{w}^i \in R_a\}$.

Spectral Method for Clustering

Would like to extend this spectral approach to the problem of detecting clusters.

Spectral Method for Clustering

Would like to extend this spectral approach to the problem of detecting clusters.
Will first offer some justification for this extension

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.

---

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.
- Because $\tilde{L}$ has eigenvalue 0 with multiplicity k, L has k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ close to zero.[9]

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.
- Because $\tilde{L}$ has eigenvalue 0 with multiplicity k, L has k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ close to zero.[9]
- Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be first k eigenvectors of L, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$ eigenvectors of $\tilde{L}$ corresponding to 0.

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.
- Because $\tilde{L}$ has eigenvalue 0 with multiplicity k, L has k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ close to zero.[9]
- Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be first k eigenvectors of L, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$ eigenvectors of $\tilde{L}$ corresponding to 0.
- Then $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is close to $\text{span}\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$.[10]

---

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.
- Because $\tilde{L}$ has eigenvalue 0 with multiplicity k, L has k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ close to zero.[9]
- Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be first k eigenvectors of L, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$ eigenvectors of $\tilde{L}$ corresponding to 0.
- Then $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is close to $\text{span}\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$.[10]
- Hence letting $\mathbf{w}^1, \ldots, \mathbf{w}^n$ denote **rows** of $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ expect to find k groups $R_1 \ldots, R_k$ of $\mathbf{w}^i$ which are similar.

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification of Spectral Method for Clustering 1:

Finding clusters is a 'small perturbation' of finding connected components.

- If G has a good k-clustering then $L = \tilde{L} + E$ where $\tilde{L}$ is k reducible, $||E|| << ||\tilde{L}||$ and E symmetric.
- Because $\tilde{L}$ has eigenvalue 0 with multiplicity k, L has k eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ close to zero.[9]
- Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ be first k eigenvectors of L, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$ eigenvectors of $\tilde{L}$ corresponding to 0.
- Then $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is close to $\text{span}\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k\}$.[10]
- Hence letting $\mathbf{w}^1, \ldots, \mathbf{w}^n$ denote **rows** of $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ expect to find k groups $R_1 \ldots, R_k$ of $\mathbf{w}^i$ which are similar.
- Thus expect $C_1, \ldots, C_k$ where $C_a = \{i : \ \mathbf{w}^i \in R_a\}$ to be a good clustering of G.

[9] M.-J. Lai, "Introduction to Numerical Analysis", Theorem 2.8.

[10] Bhatia, *Matrix analysis*, Theorem VII.3.1.

Justification for Spectral Method for Clustering 2:

Recall our earlier definition of the clustering problem:

Justification for Spectral Method for Clustering 2:

Recall our earlier definition of the clustering problem:

Definition (The Clustering Problem)

Given a graph G, find π such that:

$$Rcut(\pi) = \min\{Rcut(\sigma) : \ \sigma \text{ a partition of } V\}$$

Justification for Spectral Method for Clustering 2:

Recall our earlier definition of the clustering problem:

Definition (The Clustering Problem)

Given a graph G, find π such that:

$$Rcut(\pi) = \min\{Rcut(\sigma) : \ \sigma \text{ a partition of } V\}$$

This problem is NP-complete, can be solved exactly in $\mathcal{O}(n!)$ time.

Justification for Spectral Method for Clustering 2:

Recall our earlier definition of the clustering problem:

Definition (The Clustering Problem)

Given a graph G, find π such that:

$$Rcut(\pi) = \min\{Rcut(\sigma) : \ \sigma \text{ a partition of } V\}$$

This problem is NP-complete, can be solved exactly in $\mathcal{O}(n!)$ time. Thus, we consider an approximation to this problem, which is easier to solve.

Justification for Spectral Method for Clustering 2:

- Given any partition $\pi = \{C_1, \ldots, C_k\}$ of V, let $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ denote the indicator vectors.

Justification for Spectral Method for Clustering 2:

- Given any partition $\pi = \{C_1, \ldots, C_k\}$ of V, let $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ denote the indicator vectors.
- Define the **Indicator matrix** of π as $X^{(\pi)} = [\frac{1}{|C_1|}\mathbf{1}_{C_1}, \ldots, \frac{1}{|C_k|}\mathbf{1}_{C_k}] \in \mathbb{R}^{n \times k}$

Justification for Spectral Method for Clustering 2:

- Given any partition $\pi = \{C_1, \ldots, C_k\}$ of V, let $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ denote the indicator vectors.
- Define the **Indicator matrix** of π as $X^{(\pi)} = [\frac{1}{|C_1|}\mathbf{1}_{C_1}, \ldots, \frac{1}{|C_k|}\mathbf{1}_{C_k}] \in \mathbb{R}^{n \times k}$

Theorem

$Rcut(\pi) = trace((X^{(\pi)})^T L X^{(\pi)})$

Justification for Spectral Method for Clustering 2:

- Given any partition $\pi = \{C_1, \ldots, C_k\}$ of V, let $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ denote the indicator vectors.
- Define the **Indicator matrix** of π as $X^{(\pi)} = [\frac{1}{|C_1|}\mathbf{1}_{C_1}, \ldots, \frac{1}{|C_k|}\mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$

Theorem

$Rcut(\pi) = trace((X^{(\pi)})^T L X^{(\pi)})$

Proof.

(See Chan, Schlag, and Zien, "Spectral K -way ratio-cut partitioning and clustering", Theorem 1) $\square$

Justification for Spectral Method for Clustering 2:

- Given any partition $\pi = \{C_1, \ldots, C_k\}$ of V, let $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ denote the indicator vectors.
- Define the **Indicator matrix** of π as $X^{(\pi)} = [\frac{1}{|C_1|}\mathbf{1}_{C_1}, \ldots, \frac{1}{|C_k|}\mathbf{1}_{C_k}] \in \mathbb{R}^{n\times k}$

Theorem

$Rcut(\pi) = trace((X^{(\pi)})^T L X^{(\pi)})$

Proof.

(See Chan, Schlag, and Zien, "Spectral K -way ratio-cut partitioning and clustering", Theorem 1) □

Justification for Spectral Method for Clustering 2:

Thus one can relax the Clustering Problem to:

Justification for Spectral Method for Clustering 2:

Thus one can relax the Clustering Problem to:

Definition (Relaxation of Clustering Problem)

Find X^* such that:

$$X^* = argmin\{trace(X^T LX) : \ X \in \mathbb{R}^{n \times k}\} \tag{6}$$

Justification for Spectral Method for Clustering 2:

Thus one can relax the Clustering Problem to:

Definition (Relaxation of Clustering Problem)

Find X^* such that:

$$X^* = argmin\{trace(X^T LX) : \ X \in \mathbb{R}^{n \times k}\} \tag{6}$$

Solution to this problem can be found analytically:

Justification for Spectral Method for Clustering 2:

Thus one can relax the Clustering Problem to:

Definition (Relaxation of Clustering Problem)

Find X^* such that:

$$X^* = argmin\{trace(X^T LX) : \ X \in \mathbb{R}^{n\times k}\} \tag{6}$$

Solution to this problem can be found analytically:

Theorem

If $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are eigenvectors corresponding to k smallest eigenvalues of L, then $X^* = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ is a solution to 6.

Justification for Spectral Method for Clustering 2:

Thus one can relax the Clustering Problem to:

Definition (Relaxation of Clustering Problem)

Find X^* such that:

$$X^* = argmin\{trace(X^T LX) : \ X \in \mathbb{R}^{n\times k}\} \tag{6}$$

Solution to this problem can be found analytically:

Theorem

If $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are eigenvectors corresponding to k smallest eigenvalues of L, then $X^* = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ is a solution to 6.

Proof.

Follows from Courant-Fischer-Weyl minmax principle / Rayleigh quotient (Bhatia, *Matrix analysis*, Cor. III.1.2) □

Justification for Spectral Method for Clustering 2:

- Solution to relaxed problem should be good approximation to solution to original problem.

Justification for Spectral Method for Clustering 2:

- Solution to relaxed problem should be good approximation to solution to original problem.
- First k eigenvectors solve relaxed graph clustering problem.

Justification for Spectral Method for Clustering 2:

- Solution to relaxed problem should be good approximation to solution to original problem.
- First k eigenvectors solve relaxed graph clustering problem.
- Hence can think of them as approximations to indicator vectors of optimal clustering.

The Spectral Clustering Algorithm

Algorithm 1 Spectral Algorithm for clustering/ almost reducibility[11]

Given a non-negative, symmetric matrix A:

1. Compute eigenvalues of L, $\lambda_1, \ldots, \lambda_m$ for $1 < m << n$.
2. If $\lambda_1, \ldots, \lambda_k < \epsilon$ and $\lambda_{k+1} >> \epsilon$ then A is k-almost-reducible.
3. Compute eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$. Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$.
4. Let $\mathbf{w}^1, \ldots, \mathbf{w}^n$ denote **rows** of X. Sort these into k groups $R_1, \ldots, R_k$ using any linear time clustering algorithm (e.g. k-means).
5. Let $C_a = \{i : \mathbf{w}^i \in R_a\}$ for $a = 1, \ldots, k$.

[11] See: Ng, Jordan, and Weiss, "On Spectral Clustering: Analysis and Algorithm", and references therein.

The Spectral Clustering Algorithm

We caution that:

The Spectral Clustering Algorithm

We caution that:

Remark

- *The Spectral method is a heuristic approach to the Graph Clustering problem*

The Spectral Clustering Algorithm

We caution that:

Remark

- *The Spectral method is a heuristic approach to the Graph Clustering problem*
- *There exist (pathological) graphs on which Spectral approach will miss the best clustering (See the 'cockroach graphs' of Guattery and Miller, "On the performance of spectral graph partitioning methods")*

The Spectral Clustering Algorithm

We caution that:

Remark

- *The Spectral method is a heuristic approach to the Graph Clustering problem*
- *There exist (pathological) graphs on which Spectral approach will miss the best clustering (See the 'cockroach graphs' of Guattery and Miller, "On the performance of spectral graph partitioning methods")*
- *Observed to work well in practice and be robust to small perturbations.*

Implementation and Experimental Results

- Several ways to determine k, most based on detecting a 'jump' from λ_k to λ_{k+1}.

Implementation and Experimental Results

- Several ways to determine k, most based on detecting a 'jump' from λ_k to λ_{k+1}.
  In our implementation choose k such that:

$$\frac{|\lambda_{k+1}|}{|\lambda_k|} = \max_{i=2,\ldots,10}\left\{\frac{\lambda_{i+1}}{\lambda_i}\right\}$$

Implementation and Experimental Results

- Several ways to determine k, most based on detecting a 'jump' from λ_k to λ_{k+1}.
  In our implementation choose k such that:

$$\frac{|\lambda_{k+1}|}{|\lambda_k|} = \max_{i=2,\ldots,10}\left\{\frac{\lambda_{i+1}}{\lambda_i}\right\}$$

- k-means used to find $R_1, \ldots, R_k$

Implementation and Experimental Results

- Several ways to determine k, most based on detecting a 'jump' from λ_k to λ_{k+1}.
  In our implementation choose k such that:

$$\frac{|\lambda_{k+1}|}{|\lambda_k|} = \max_{i=2,\ldots,10}\left\{\frac{\lambda_{i+1}}{\lambda_i}\right\}$$

- k-means used to find $R_1, \ldots, R_k$
- Code written in MATLAB.

Implementation and Experimental Results

- Several ways to determine k, most based on detecting a 'jump' from λ_k to λ_{k+1}.
  In our implementation choose k such that:

$$\frac{|\lambda_{k+1}|}{|\lambda_k|} = \max_{i=2,\ldots,10}\left\{\frac{\lambda_{i+1}}{\lambda_i}\right\}$$

- k-means used to find $R_1, \ldots, R_k$
- Code written in MATLAB.
- Tested on a variety of natural and artificial data sets.

Experimental Results: Artificial Data Set

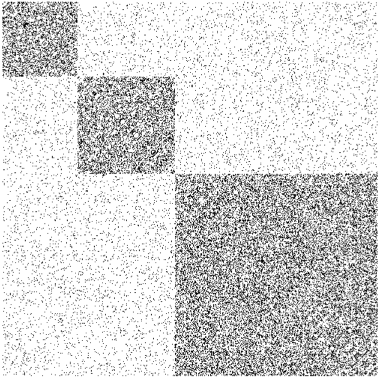


Figure: Original Matrix A

Experimental Results: Artificial Data Set

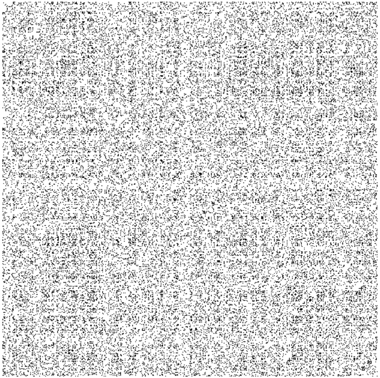


Figure: QAQ^T for a random permutation matrix Q

Experimental Results: Artificial Data Set

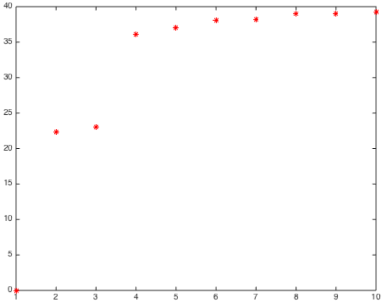


Figure: First 10 eigenvalues of L

Experimental Results: Artificial Data Set

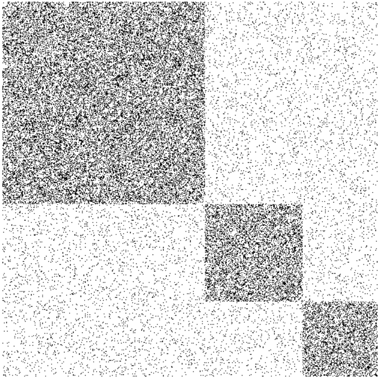


Figure: PAP^T for P found using spectral method

Experimental Results: Facebook Data Set[12]

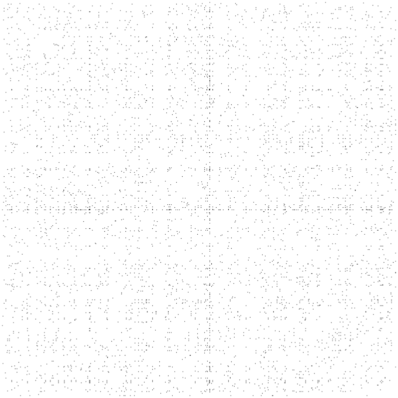


Figure: A for a graph G consisting of anonymised Facebook users and their friendship connections

[12]Leskovec and Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*

Experimental Results: Facebook Data Set[13]

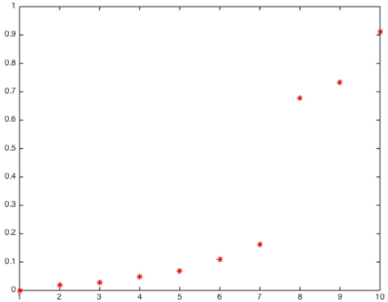


Figure: First 10 eigenvalues of L

[13] Leskovec and Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*.

Experimental Results: Facebook Data Set[14]

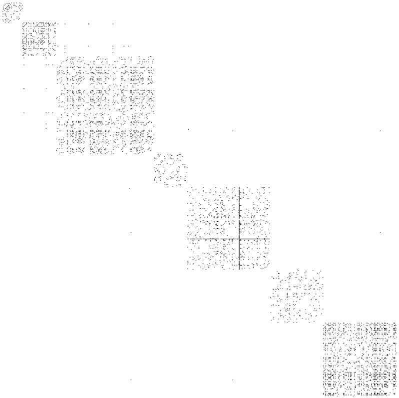


Figure: PAP^T for P found using spectral method.

[14] Leskovec and Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*.

Generalizations

One can generalize the Spectral algorithm to:

- Matrices A which are not non-negative, by working with $A_{new} = A + |\min_{i,j} a_{ij}|$.

Generalizations

One can generalize the Spectral algorithm to:

- Matrices A which are not non-negative, by working with $A_{new} = A + |\min_{i,j} a_{ij}|$.
- Matrices A which are non-symmetric, (See Malliaros and Vazirgiannis, "Clustering and community detection in directed networks: A survey") and references within.

Generalizations

One can generalize the Spectral algorithm to:

- Matrices A which are not non-negative, by working with $A_{new} = A + |\min_{i,j} a_{ij}|$.
- Matrices A which are non-symmetric, (See Malliaros and Vazirgiannis, "Clustering and community detection in directed networks: A survey") and references within.
- Matrices A which are non-square, (see Dhillon, "Co-clustering documents and words using Bipartite spectral graph partitioning")

Table of Contents

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.
- All existing implementations use standard eigenvector finding algorithms, eg. Lanscoz method.

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.
- All existing implementations use standard eigenvector finding algorithms, eg. Lanscoz method.
- This method is $\mathcal{O}(n^3)$[15], making it impossible to apply spectral clustering to very large data sets.

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.
- All existing implementations use standard eigenvector finding algorithms, eg. Lanscoz method.
- This method is $\mathcal{O}(n^3)$[15], making it impossible to apply spectral clustering to very large data sets.
- But *a priori* we know a lot about eigenvectors of L.

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.
- All existing implementations use standard eigenvector finding algorithms, eg. Lanscoz method.
- This method is $\mathcal{O}(n^3)$[15], making it impossible to apply spectral clustering to very large data sets.
- But *a priori* we know a lot about eigenvectors of L.
- Expect $L = \tilde{L} + E$, where $\tilde{L}$ is k-reduced (hence has k eigenvectors of the form $\mathbf{1}_{C_i}$) and E is a small symmetric perturbation.

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

Speeding up eigenvector computation

- Computational bottleneck of spectral clustering is the computation of eigenvectors of L.
- All existing implementations use standard eigenvector finding algorithms, eg. Lanscoz method.
- This method is $\mathcal{O}(n^3)$[15], making it impossible to apply spectral clustering to very large data sets.
- But *a priori* we know a lot about eigenvectors of L.
- Expect $L = \tilde{L} + E$, where $\tilde{L}$ is k-reduced (hence has k eigenvectors of the form $\mathbf{1}_{C_i}$) and E is a small symmetric perturbation.
- Can we use this additional information to develop an eigenvector finding routine adapted specifically to the case of finding first k eigenvectors of a (graph) Laplacian?

[15] Borm and Mehl, *Numerical Methods for Eigenvalue Problems*.

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).
- Assume that $|C_1|$ is the smallest amongst the $|C_i|$.

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).
- Assume that $|C_1|$ is the smallest amongst the $|C_i|$.
- As before, $\{\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}\}$ is a basis for W_0

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).
- Assume that $|C_1|$ is the smallest amongst the $|C_i|$.
- As before, $\{\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}\}$ is a basis for W_0
- Note that the $\mathbf{1}_{C_i}$ have **disjoint support** .

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).
- Assume that $|C_1|$ is the smallest amongst the $|C_i|$.
- As before, $\{\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}\}$ is a basis for W_0
- Note that the $\mathbf{1}_{C_i}$ have **disjoint support** .
- If $\mathbf{w} \in W_0$ and $\mathbf{w} \neq \mathbf{0}$ then $w = \sum_{i=1}^{k} \alpha_i \mathbf{1}_{C_i}$ with not all $\alpha_i = 0$.

A Compressed sensing approach

- Consider again case where G has k connected components $C_1, \ldots, C_k$ (equivalently A is k-reducible).
- Assume that $|C_1|$ is the smallest amongst the $|C_i|$.
- As before, $\{\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}\}$ is a basis for W_0
- Note that the $\mathbf{1}_{C_i}$ have **disjoint support** .
- If $\mathbf{w} \in W_0$ and $\mathbf{w} \neq \mathbf{0}$ then $w = \sum_{i=1}^{k} \alpha_i \mathbf{1}_{C_i}$ with not all $\alpha_i = 0$.

Idea: $\mathbf{1}_{C_1}$ has the fewest non-zero entries among all elements of $W_0 \setminus \{\mathbf{0}\}$, so we could find it as:

$$\min ||\mathbf{w}||_0 \text{ subject to } L\mathbf{w} = \mathbf{0} \tag{7}$$

A Compressed sensing approach

- Need to add a 'normalization' condition to (7): $w_1 = 1$

A Compressed sensing approach

- Need to add a 'normalization' condition to (7): $w_1 = 1$

Consider the problem:

Problem

Find $\mathbf{w}^* \in \mathbb{R}^n$ *such that:*

$$\mathbf{w}^* = argmin\{||\mathbf{w}||_0 \text{ subject to } L\mathbf{w} = \mathbf{0} \text{ and } w_1 = 1\} \tag{8}$$

A Compressed sensing approach

- Need to add a 'normalization' condition to (7): $w_1 = 1$

Consider the problem:

Problem

Find $\mathbf{w}^* \in \mathbb{R}^n$ *such that:*

$$\mathbf{w}^* = \mathit{argmin}\{||\mathbf{w}||_0 \text{ subject to } L\mathbf{w} = \mathbf{0} \text{ and } w_1 = 1\} \tag{8}$$

Note that

$$w_1 = 1 \Rightarrow \mathbf{w} = \begin{pmatrix} 1 \\ \hat{\mathbf{w}} \end{pmatrix} \text{ with } \hat{\mathbf{w}} \in \mathbb{R}^{n-1}$$

A Compressed sensing approach

- Need to add a 'normalization' condition to (7): $w_1 = 1$

Consider the problem:

Problem

Find $\mathbf{w}^* \in \mathbb{R}^n$ such that:

$$\mathbf{w}^* = \mathit{argmin}\{||\mathbf{w}||_0 \text{ subject to } L\mathbf{w} = \mathbf{0} \text{ and } w_1 = 1\} \tag{8}$$

Note that

$$w_1 = 1 \Rightarrow \mathbf{w} = \begin{pmatrix} 1 \\ \hat{\mathbf{w}} \end{pmatrix} \text{ with } \hat{\mathbf{w}} \in \mathbb{R}^{n-1}$$

So: $L\mathbf{w} = \mathbf{0}$ and $w_1 = 1 \Leftrightarrow L_{-1}\hat{\mathbf{w}} = -\ell_1$ where $L = [\ell_1, \ell_2, \ldots, \ell_n]$ and $L_{-1} = [\ell_2, \ldots, \ell_n]$

A Compressed sensing approach

Lemma

If L is k-reducible the solution then $\mathbf{w}^* = \mathbf{1}_{i^*}$ where $1 \in C_{i^*}$ is the unique solution to (8)

Proof.

- If $\mathbf{w} \in W_0$ then $\mathbf{w} = \sum_i \alpha_i \mathbf{1}_{C_i}$.

A Compressed sensing approach

Lemma

If L is k-reducible the solution then $\mathbf{w}^* = \mathbf{1}_{i^*}$ where $1 \in C_{i^*}$ is the unique solution to (8)

Proof.

- If $\mathbf{w} \in W_0$ then $\mathbf{w} = \sum_i \alpha_i \mathbf{1}_{C_i}$.
- If $w_1 = 1$ then $\alpha_{i^*} = 1$ and so: $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$

A Compressed sensing approach

Lemma

If L is k-reducible the solution then $\mathbf{w}^* = \mathbf{1}_{i^*}$ where $1 \in C_{i^*}$ is the unique solution to (8)

Proof.

- If $\mathbf{w} \in W_0$ then $\mathbf{w} = \sum_i \alpha_i \mathbf{1}_{C_i}$.
- If $w_1 = 1$ then $\alpha_{i^*} = 1$ and so: $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- $$||\mathbf{w}||_0 = \sum_{i:\alpha_i\neq 0} |C_i| = |C_{i^*}| + \sum_{i:i\neq i^*,\alpha_i\neq 0} |C_i|$$

A Compressed sensing approach

Lemma

If L is k-reducible the solution then $\mathbf{w}^* = \mathbf{1}_{i^*}$ where $1 \in C_{i^*}$ is the unique solution to (8)

Proof.

- If $\mathbf{w} \in W_0$ then $\mathbf{w} = \sum_i \alpha_i \mathbf{1}_{C_i}$.
- If $w_1 = 1$ then $\alpha_{i^*} = 1$ and so: $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- $$||\mathbf{w}||_0 = \sum_{i:\alpha_i \neq 0} |C_i| = |C_{i^*}| + \sum_{i:i\neq i^*, \alpha_i\neq 0} |C_i|$$
- This is clearly minimized when $\alpha_i = 0$ for all $i \neq i^*$.

A Compressed sensing approach

Lemma

If L is k-reducible the solution then $\mathbf{w}^* = \mathbf{1}_{i^*}$ where $1 \in C_{i^*}$ is the unique solution to (8)

Proof.

- If $\mathbf{w} \in W_0$ then $\mathbf{w} = \sum_i \alpha_i \mathbf{1}_{C_i}$.
- If $w_1 = 1$ then $\alpha_{i^*} = 1$ and so: $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- $$||\mathbf{w}||_0 = \sum_{i:\alpha_i \neq 0} |C_i| = |C_{i^*}| + \sum_{i:i\neq i^*, \alpha_i\neq 0} |C_i|$$
- This is clearly minimized when $\alpha_i = 0$ for all $i \neq i^*$.
- Hence $\mathbf{w} = \mathbf{1}_{C_{i^*}}$ is indeed the minimizer.

$\square$

A Compressed sensing approach

Moreover, we can replace the 0 "norm" in (8) by any p norm (for $1 \leq p < \infty$):

Problem

Find $\mathbf{w}^* \in \mathbb{R}^n$ *such that:*

$$\mathbf{w}^* = argmin\{||\mathbf{w}||_p \text{ subject to } L\mathbf{w} = \mathbf{0} \text{ and } w_1 = 1\} \qquad (9)$$

A Compressed sensing approach

Theorem

$\mathbf{w}^*$ *solves* (8) *if and only if it solves* (9).

A Compressed sensing approach

Theorem

$\mathbf{w}^*$ *solves* (8) *if and only if it solves* (9).

Proof.

- If $\mathbf{w} \in W_0$ and $w_1 = 1$ then $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$

A Compressed sensing approach

Theorem

$\mathbf{w}^*$ *solves* (8) *if and only if it solves* (9).

Proof.

- If $\mathbf{w} \in W_0$ and $w_1 = 1$ then $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- Because the $\mathbf{1}_{C_i}$'s have disjoint support:

$$||\mathbf{w}||_p^p = |C_{i^*}| + \sum_{i:i\neq i^*,\alpha_i\neq 0} |\alpha_i|^p |C_i|$$

A Compressed sensing approach

Theorem

$\mathbf{w}^*$ *solves* (8) *if and only if it solves* (9).

Proof.

- If $\mathbf{w} \in W_0$ and $w_1 = 1$ then $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- Because the $\mathbf{1}_{C_i}$'s have disjoint support:

$$||\mathbf{w}||_p^p = |C_{i^*}| + \sum_{i:i\neq i^*, \alpha_i \neq 0} |\alpha_i|^p |C_i|$$

- This is clearly minimized, for any p, by setting $\alpha_i = 0$ for all $i \neq i^*$.

A Compressed sensing approach

Theorem

$\mathbf{w}^*$ *solves* (8) *if and only if it solves* (9).

Proof.

- If $\mathbf{w} \in W_0$ and $w_1 = 1$ then $\mathbf{w} = \mathbf{1}_{C_{i^*}} + \sum_{i:i\neq i^*} \alpha_i \mathbf{1}_{C_i}$
- Because the $\mathbf{1}_{C_i}$'s have disjoint support:

$$||\mathbf{w}||_p^p = |C_{i^*}| + \sum_{i:i\neq i^*, \alpha_i \neq 0} |\alpha_i|^p |C_i|$$

- This is clearly minimized, for any p, by setting $\alpha_i = 0$ for all $i \neq i^*$.
- Hence $\mathbf{w}^* = \mathbf{1}_{C_{i^*}}$

Preliminary Results

Have implemented the $p = 2$ case in MATLAB, solves the connected components problem well.

Preliminary Results

Have implemented the $p = 2$ case in MATLAB, solves the connected components problem well.

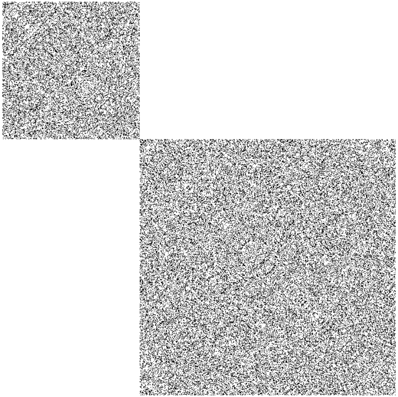


Figure: Original Matrix A

Preliminary Results

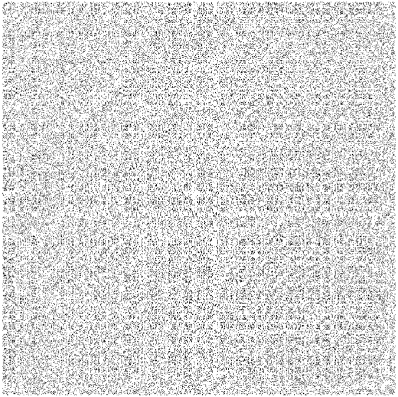


Figure: QAQ^T for a random permutation matrix Q

Preliminary Results

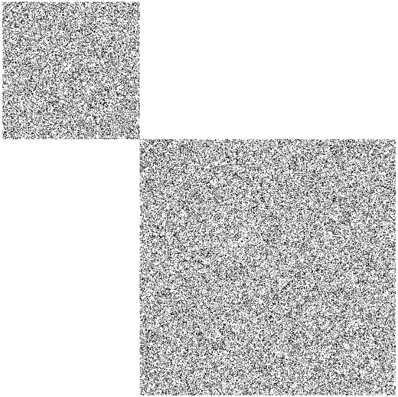


Figure: PAP^T for P found using Compressed Sensing method.

Preliminary Results

But does not extend well to the clustering problem.

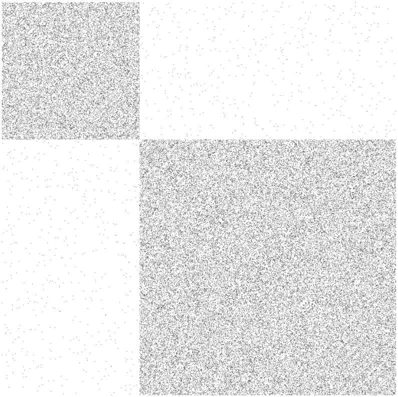


Figure: Original matrix A

Preliminary Results

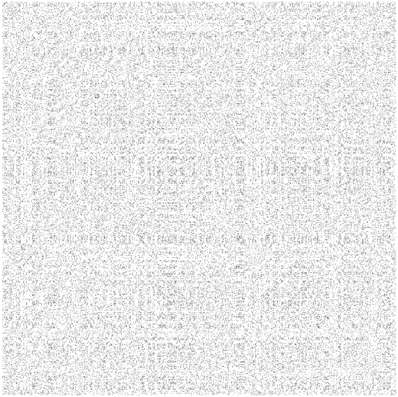


Figure: QAQ^T for a random permutation matrix Q

Preliminary Results

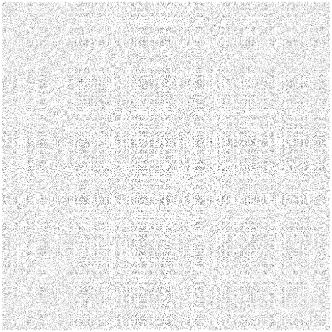


Figure: The ‘unscrambled’ matrix

Preliminary Results

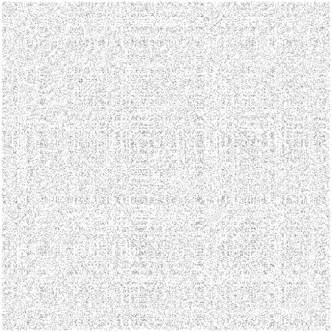


Figure: The 'unscrambled' matrix

Morally, this is because eigenvectors do not vary continuously when $\tilde{L}$ is perturbed to $L = \tilde{L} + E$.

Preliminary Results

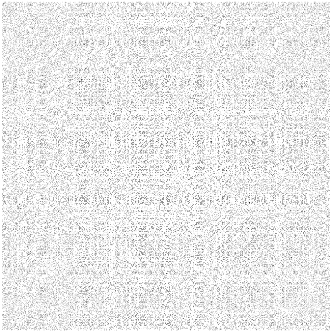


Figure: The ‘unscrambled’ matrix

Morally, this is because eigenvectors do not vary continuously when $\tilde{L}$ is perturbed to $L = \tilde{L} + E$.
The eigenspaces, however, are better behaved.

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.
- Let $\tilde{X} = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}]$, then $\tilde{L}\tilde{X} = 0$.

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.
- Let $\tilde{X} = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}]$, then $\tilde{L}\tilde{X} = 0$.
- Now let $L = \tilde{L} + E$ be an almost-k-reducible matrix.

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.
- Let $\tilde{X} = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}]$, then $\tilde{L}\tilde{X} = 0$.
- Now let $L = \tilde{L} + E$ be an almost-k-reducible matrix.
- Then $L\tilde{X} = \tilde{L}\tilde{X} + E\tilde{X} = E\tilde{X} \approx 0$.

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.
- Let $\tilde{X} = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}]$, then $\tilde{L}\tilde{X} = 0$.
- Now let $L = \tilde{L} + E$ be an almost-k-reducible matrix.
- Then $L\tilde{X} = \tilde{L}\tilde{X} + E\tilde{X} = E\tilde{X} \approx 0$.
- Moreover, all row sums of $\tilde{X}$ equal 1. (in fact all rows have precisely one non-zero element).

(Another) Compressed Sensing Approach

- Suppose again that $\tilde{L}$ is k-reducible.
- Let $\tilde{X} = [\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}]$, then $\tilde{L}\tilde{X} = 0$.
- Now let $L = \tilde{L} + E$ be an almost-k-reducible matrix.
- Then $L\tilde{X} = \tilde{L}\tilde{X} + E\tilde{X} = E\tilde{X} \approx 0$.
- Moreover, all row sums of $\tilde{X}$ equal 1. (in fact all rows have precisely one non-zero element).

Problem

For $X \in \mathbb{R}^{n\times k}$ let $\mathbf{r}_X \in \mathbb{R}^n$ denote vector of row sums of X. Find X^* such that:

$$X^* = \mathit{argmin}\{||LX|| \text{ subject to } \mathbf{r}_X = \mathbf{1}\}$$

(Another) Compressed Sensing Approach

Hopefully X^* will provide a good approximation to $\tilde{X}$.

Table of Contents

Application 1: Decoupling systems of ODE's

- Suppose we have n functions of time: $x_1, \ldots, x_n$ whose evolution is given by:

$$\dot{x}_i = \sum_j a_{ij} x_j$$

Application 1: Decoupling systems of ODE's

- Suppose we have n functions of time: $x_1, \ldots, x_n$ whose evolution is given by:

$$\dot{x}_i = \sum_j a_{ij} x_j$$

- Can represent this as $\dot{\mathbf{x}} = A\mathbf{x}$ where $A \in \mathbb{R}^{n \times n}$.

Application 1: Decoupling systems of ODE's

- Suppose we have n functions of time: $x_1, \ldots, x_n$ whose evolution is given by:

$$\dot{x}_i = \sum_j a_{ij} x_j$$

- Can represent this as $\dot{\mathbf{x}} = A\mathbf{x}$ where $A \in \mathbb{R}^{n \times n}$.
- Suppose

$$PAP^T = B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1,k} \\ 0 & B_{22} & \cdots & B_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{kk} \end{bmatrix},$$

Application 1: Decoupling systems of ODE's

- Let $P\mathbf{x} = \mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)^T$ and consider equivalent system $\dot{\mathbf{y}} = B\mathbf{y}$

Application 1: Decoupling systems of ODE's

- Let $P\mathbf{x} = \mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)^T$ and consider equivalent system $\dot{\mathbf{y}} = B\mathbf{y}$
- Now need only solve smaller systems

$$\begin{aligned}
\dot{\mathbf{y}}_k &= B_{kk}\mathbf{y}_k \\
\dot{\mathbf{y}}_{k-1} &= B_{k-1,k-1}\mathbf{y}_{k-1} + B_{k-1,k}\mathbf{y}_k \\
&\vdots \\
\dot{\mathbf{y}}_1 &= B_{11}\mathbf{y}_1 + B_{12}\mathbf{y}_2 + \ldots + B_{1r}\mathbf{y}_k
\end{aligned}$$

Application 1: Decoupling systems of ODE's

- If A is sparse it likely is reducible. Will such a pre-processing step as above speed up numerical solution of this system?

Application 1: Decoupling systems of ODE's

- If A is sparse it likely is reducible. Will such a pre-processing step as above speed up numerical solution of this system?
- Even if $PAP^T = B + E$ with $||E|| << B)$ and B k-reduced, might we still find useful approximate solutions like this by ignoring E?

Application 2: Feature recognition in images

- Suppose we have a collection $\{\mathbf{y}_1, \ldots, \mathbf{y}_n]$ of images, each $\mathbf{y}_i \in \mathbb{R}^m$ $(n >> m)$.

[16] Ming-jun Lai. *Nonconvex and Non-Lipschitz Differentiable Minimization for Sparse Solution of Underdetermined Linear Systems*. 2016.

Application 2: Feature recognition in images

- Suppose we have a collection $\{\mathbf{y}_1, \ldots, \mathbf{y}_n]$ of images, each $\mathbf{y}_i \in \mathbb{R}^m$ $(n >> m)$.
- Suppose the $\mathbf{y}_i$ are images of k different people's faces, in a variety of angles and conditions.

[16] Ming-jun Lai. *Nonconvex and Non-Lipschitz Differentiable Minimization for Sparse Solution of Underdetermined Linear Systems*. 2016.

Application 2: Feature recognition in images

- Suppose we have a collection $\{\mathbf{y}_1, \ldots, \mathbf{y}_n]$ of images, each $\mathbf{y}_i \in \mathbb{R}^m$ ($n >> m$).
- Suppose the $\mathbf{y}_i$ are images of k different people's faces, in a variety of angles and conditions.
- Wish to sort the $\{\mathbf{y}_i\}$ into k groups, each corresponding to a single person.
- **Idea:**[16] each $\mathbf{y}_i$ should be well approximated by a linear combination of a (few) other images of same face.

[16] Ming-jun Lai. *Nonconvex and Non-Lipschitz Differentiable Minimization for Sparse Solution of Underdetermined Linear Systems*. 2016.

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i\hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

to have a sparse solution.

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i\hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

  to have a sparse solution.
- Expand $\hat{\mathbf{x}}_i \in \mathbb{R}^{n-1}$ to $\mathbf{x}_i \in \mathbb{R}^n$ by:

$$x_{i,j} = \begin{cases} \hat{x}_{i,j} & \text{if } j < i \\ 1 & \text{if } j = i \\ \hat{x}_{i,j-1} \text{ if } j > i \end{cases}$$

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i \hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

  to have a sparse solution.
- Expand $\hat{\mathbf{x}}_i \in \mathbb{R}^{n-1}$ to $\mathbf{x}_i \in \mathbb{R}^n$ by:

$$x_{i,j} = \begin{cases} \hat{x}_{i,j} & \text{if } j < i \\ 1 & \text{if } j = i \\ \hat{x}_{i,j-1} \text{ if } j > i & \end{cases}$$

- Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times n}$.

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i\hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

  to have a sparse solution.
- Expand $\hat{\mathbf{x}}_i \in \mathbb{R}^{n-1}$ to $\mathbf{x}_i \in \mathbb{R}^n$ by:

$$x_{i,j} = \begin{cases} \hat{x}_{i,j} & \text{if } j < i \\ 1 & \text{if } j = i \\ \hat{x}_{i,j-1} \text{ if } j > i \end{cases}$$

- Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times n}$.
- Suppose can find permutation P such that $PXP^T = B + E$ with B block diagonal and $||E|| << ||B||$.

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i \hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

  to have a sparse solution.
- Expand $\hat{\mathbf{x}}_i \in \mathbb{R}^{n-1}$ to $\mathbf{x}_i \in \mathbb{R}^n$ by:

$$x_{i,j} = \begin{cases} \hat{x}_{i,j} & \text{if } j < i \\ 1 & \text{if } j = i \\ \hat{x}_{i,j-1} \text{ if } j > i \end{cases}$$

- Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times n}$.
- Suppose can find permutation P such that $PXP^T = B + E$ with B block diagonal and $||E|| << ||B||$.
- Then columns permuted together into same block by P correspond to images that are strongly correlated.

Application 2: Feature recognition in images

- If $\hat{Y}_i = [\mathbf{y}_1, \ldots, \hat{\mathbf{y}}_i, \ldots, \mathbf{y}_n]$ then expect:

$$\min ||\hat{\mathbf{x}}_i||_0 \text{ subject to } ||Y_i\hat{\mathbf{x}}_i - \mathbf{y}_i||_2 < \epsilon$$

  to have a sparse solution.
- Expand $\hat{\mathbf{x}}_i \in \mathbb{R}^{n-1}$ to $\mathbf{x}_i \in \mathbb{R}^n$ by:

$$x_{i,j} = \begin{cases} \hat{x}_{i,j} & \text{if } j < i \\ 1 & \text{if } j = i \\ \hat{x}_{i,j-1} \text{ if } j > i \end{cases}$$

- Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times n}$.
- Suppose can find permutation P such that $PXP^T = B + E$ with B block diagonal and $||E|| << ||B||$.
- Then columns permuted together into same block by P correspond to images that are strongly correlated.
- Hence blocks of B should correspond to images of same face.

The End

Thank you for listening!

The End

Thank you for listening!

Any questions, comments or suggestions?

References I

Bertrand, Alexander and Marc Moonen. "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks". In: *Signal Processing* 93.5 (2013), pp. 1106–1117.

Bhatia, Rajendra. *Matrix analysis*. Vol. 169. Springer Science & Business Media, 2013.

Bollobas, Bela. *Graph theory: an introductory course*. Vol. 63. Springer Science & Business Media, 2012.

Bollt, Erik M and Naratip Santitissadeekorn. *Applied and Computational Measurable Dynamics*. Vol. 18. SIAM, 2013.

Borm, Steffen and Christian Mehl. *Numerical Methods for Eigenvalue Problems*. De Gruyter, 2012.

References II

Chan, Pak K., Martine D. F. Schlag, and Jason Y. Zien. "Spectral K -way ratio-cut partitioning and clustering". In: *Proceedings of the 30th international on Design automation conference - DAC '93* 13.9 (1993), pp. 749–754.

Chung, Fan RK. *Spectral graph theory*. Vol. 92. American Mathematical Soc., 1997.

Dhillon, Inderjit s. "Co-clustering documents and words using Bipartite spectral graph partitioning". In: *Proc of 7th ACM SIGKDD Conf* (2001), pp. 269–274.

Guattery, Stephen and G.L. Miller. "On the performance of spectral graph partitioning methods". In: *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms* (1995), pp. 233–242.

Jain, Anil K. "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666.

References III

Lai, Ming-Jun. "Introduction to Numerical Analysis". 1991.

Lai, Ming-jun. *Nonconvex and Non-Lipschitz Differentiable Minimization for Sparse Solution of Underdetermined Linear Systems*. 2016.

Leskovec, Jure and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014.

Lovasz, Laszlo. "Very large graphs". In: *Current Developments in Mathematics* 67867 (2009), p. 63.

Malliaros, Fragkiskos D. and Michalis Vazirgiannis. "Clustering and community detection in directed networks: A survey". In: *Physics Reports* 533.4 (2013), pp. 95–142.

References IV

Nascimento, Maria C V and Andre C P L F De Carvalho. "Spectral methods for graph clustering - A survey". In: *European Journal of Operational Research* 211.2 (2011), pp. 221–231.

Ng, Andrew Y, Michael I Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and Algorithm". In: *Adv. Neural Inf. Process. Syst.* (2001), pp. 849–856.

Von Luxburg, Ulrike. "A Tutorial on Spectral Clustering". In: March (2007), pp. 1–32.