CS-GY 9223 I: Lecture 10
Krylov methods, spectral clustering, spectral graph theory.

NYU Tandon School of Engineering, Prof. Christopher Musco

$2A^T(Ax-b) = \nabla \|Ax-b\|_2^2$

Three classes of methods.

- Direct Methods:

  Exact computation of A^{-1}: Gaussian elimination $O(n^3)$ → $n \times n$ matrix

  QR Algorithm: $O(n^3)$ for SVD eigendecomposition

- Iterative Methods:

  Power Method: Top singular vectors.

  $\|Ax-b\|_2^2$: Gradient Descent:

  Main Cost: Matrix-Vector product w/ A. → $O(nd)$

- Randomized Methods:

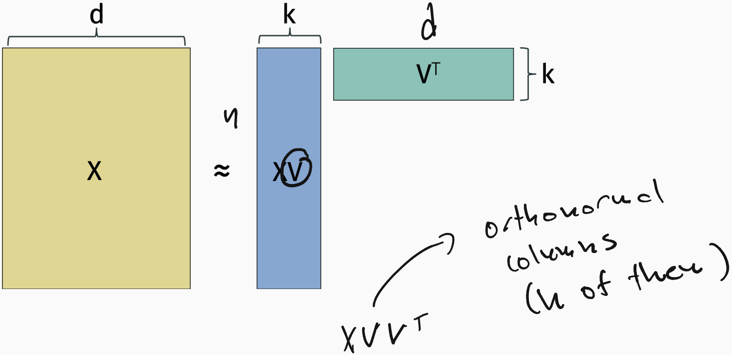
  Methods based on JL.

  SGD, SCD

$nnz(A)$ = "number of nonzeros in A"

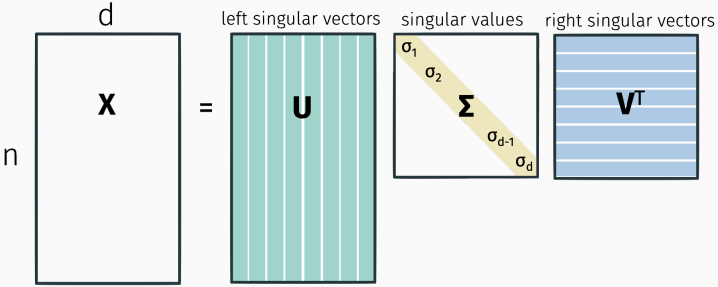
In general computing Ax takes $O(nnz(A))$ time $\leq O(nd)$

Write $\mathbf{X}$ as a rank k factorization by projecting onto the subspace spanned by an orthanormal matrix $\mathbf{V} \in \mathbb{R}^{d \times k}$



One-stop shop for computing optimal low-rank approximations.

Any matrix **X** can be written:



Where $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$.

Given a subspace $\mathcal{V}$ spanned by the k columns in $\mathbf{V}$,

$$\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{V}^T\|_F^2$$

We want to find the best $\mathbf{V} \in \mathbb{R}^{d\times k}$:

$$\min_{\text{orthonormal } \mathbf{V}\in\mathbb{R}^{d\times k}} \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 \tag{1}$$

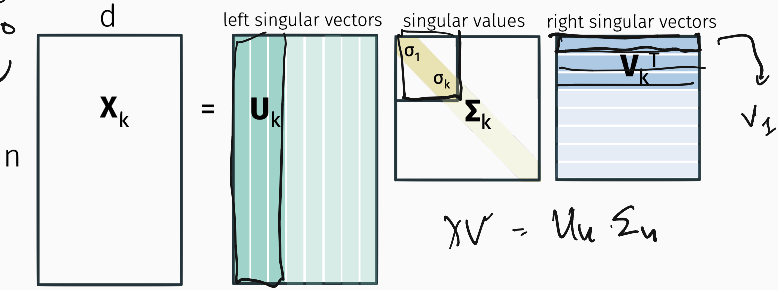
Note that $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$ for all orthonormal $\mathbf{V}$ (since $\mathbf{V}\mathbf{V}^T$ is a projection). Equivalent form:

$$\max_{\text{orthonormal } \mathbf{V}\in\mathbb{R}^{d\times k}} \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\mathbf{V}\|_F^2 \tag{2}$$

SINGULAR VALUE DECOMPOSITION

Can read off optimal low-rank approximations from the SVD:

$O(nd^2)$ time to compute



$$\mathbf{X}_k = \mathbf{U}_k\mathbf{U}_k^T\mathbf{X} = \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T.$$

$$\mathbf{V}_k = \underset{\text{orthonormal } \mathbf{V}\in\mathbb{R}^{d\times k}}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \underset{\text{orthonormal } \mathbf{V}\in\mathbb{R}^{d\times k}}{\arg\max} \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$$

Goal: Find some $\mathbf{z} \approx \mathbf{v}_1$.

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ with SVD $\mathbf{U\Sigma V}$.

"number of non-zeros in X"

Power method:

- Choose $\mathbf{z}^{(0)}$ randomly. E.g. $\mathbf{z}_0 \sim \mathcal{N}(0, 1)$.
- For $i = 1, \ldots, T$
  - $\mathbf{z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{z}^{(i-1)})$
  - $n_i = \|\mathbf{z}^{(i)}\|_2$
  - $\mathbf{z}^{(i)} = \mathbf{z}^{(i)}/n_i$

  Return $\mathbf{z}_T$

nnz(X) time

nnz(X)

+ d

Theorem (Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be parameter capturing the "gap" between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log(d/\epsilon)}{\gamma}\right)$ steps, we have:

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 \leq \epsilon.$$

Total runtime: $O(T \cdot \mathrm{nnz}(\mathbf{X})) \leq O(T \cdot nd)$

cost per iterations

svds

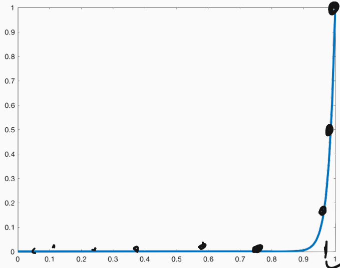
eigs

q = # of iterations

random gaussian

$$z^{(q)} = c \cdot \underline{(X^TX)^q} \cdot \underline{g}$$

$g = c_1 v_1 + c_2 v_2 + \ldots c_d v_d$



σ_2/σ_1

$\left(\frac{\sigma_2}{\sigma_1}\right)^{2q}$

$$\underline{z}^{(q)} = c \cdot \left[c_1 \cdot \frac{\sigma_1^{2q}}{\sigma_1^{2q}} v_1 + c_2 \cdot \frac{\sigma_2^{2q}}{\sigma_1^{2q}} v_2 + \ldots + c_d \cdot \frac{\sigma_d^{2q}}{\sigma_1^{2q}} v_d \right]$$

$X^T(X(X^T(Xg)))$

$$\mathbf{z}^{(q)} = c \cdot \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$$

Along the way we computed:

$$\mathcal{K}_q = \left[\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right) \cdot \mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)^2 \cdot \mathbf{g}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}\right]$$

$\mathcal{K}$ is called the Krylov subspace of degree q.

Idea behind Krlyov methods: Don't throw away everything before $\left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$. What you're using when you run `svds` or `eigs` in MATLAB or Python.

Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method: $\mathcal{K} =$ d, q $\to O(dq^2)$

- Let $\mathbf{Q} \in \mathbb{R}^{d\times q}$ be an orthonormal span for the vectors in $\mathcal{K}_q$
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$. $\to$ need to compute SVD Q^TX^TXQ
  - Find best vector in the Krylov subspace, instead of just using last vector.
  - Can be done in $O\left(\text{nnz}(\mathbf{X}) \cdot q + dq^2\right)$ time.

$\min_{v} \|X - Xvv^T\|_F^2 \to$ optimal rank 1 approximation

d

Require SVD.

LANCZOS METHOD ANALYSIS

Claim 1: For any degree q polynomial p, we can write $p(\mathbf{X}^T\mathbf{X}) \cdot \mathbf{g}$ as $\mathbf{Q}\mathbf{w}$ for some $\mathbf{w}$.

$$\left[c_1 (X^T X) + c_2 (X^T X)^2 + \dots + c_q (X^T X)^q\right] g$$
$$= p(X^T X) g$$

Claim 2:

$$\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2 = \min_{\text{degree } q \text{ polynomial} p} \|\mathbf{X} - \mathbf{X}\mathbf{v}_p\mathbf{v}_p^T\|_F^2$$

where $\mathbf{v}_p = p(\mathbf{X}^T\mathbf{X}) \cdot \mathbf{g}$.

Claim 3:

$$p(X^T X) \cdot g = c \cdot \left[c_1 \cdot \underbrace{p(\sigma_1^2)}_{p(\sigma_1^2)} \mathbf{v}_1 + c_2 \cdot \underbrace{p(\sigma_2^2)}_{p(\sigma_1^2)} \mathbf{v}_2 + \ldots + c_n \cdot p(\sigma_n^2) \mathbf{v}_n\right]$$

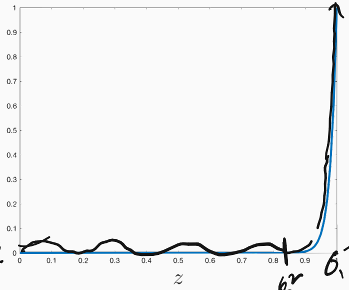
$$(X^T X)^q g = c \cdot \left[c_1 \sigma_1^{2q} v_1 + \dots c_d \sigma_d^{2q} v_d\right]$$

Claim: There is an $O\left(\sqrt{q \log \frac{1}{\epsilon}}\right)$ degree polynomial $\hat{p}$ approximating $\mathbf{x}^q$ up to error ϵ on $[0, \sigma_1^2]$.

$p(z) = z^q$

z^q

$\pm\epsilon$

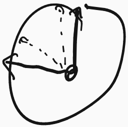


$\frac{p(\sigma_1^2)}{p(\sigma_1^2)} \to \approx 1$ $\quad \frac{p(\sigma_2^2)}{p(\sigma_1^2)} \to \approx q$ $\quad \dots \quad \frac{p(\sigma_d^2)}{p(\sigma_1)^2}$ close to zero

$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_{p*}\mathbf{v}_{p*}^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_{\hat{p}}\mathbf{v}_{\hat{p}}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_{x^q}\mathbf{v}_{x^q}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

Runtime: $O\left(\frac{\log(d/\epsilon)}{\sqrt{\gamma}} \cdot \mathrm{nnz}(\mathbf{X})\right)$ vs. $O\left(\frac{\log(d/\epsilon)}{\gamma} \cdot \mathrm{nnz}(\mathbf{X})\right)$

Convergence is slow when $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ is small. $\mathbf{z}^{(q)}$ has large components of both $\mathbf{v}_1$ and $\mathbf{v}_2$. But in this case:



$\sum \sigma_i^2 - \sigma_1^2$

$\sigma_i^2 = \|\mathbf{X} - \mathbf{X}\mathbf{v}_2\mathbf{v}_2^T\|_F^2$

$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 = \sum_{i \neq 1} \sigma_i^2 \approx \sum_{i \neq 2} \cancel{\sigma_i^2 \|\mathbf{X} - \mathbf{X}\mathbf{v}_2\mathbf{v}_2^T\|_F^2}$$

$\sum \sigma_i^2 - \sigma_2^2$

So we don't care! Either $\mathbf{v}_1$ or $\mathbf{v}_2$ give good rank-1 approximations.

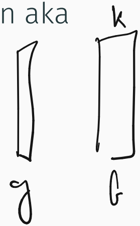
Claim: To achieve

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1 + \epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

we need $O\left(\frac{\log(d/\epsilon)}{\epsilon}\right)$ power method iterations or $O\left(\frac{\log(d/\epsilon)}{\sqrt{\epsilon}}\right)$ Lanczos iterations.

$(X - Xv_1v_1^T) \xrightarrow{\text{Power Method}} \approx v_2$

- Block Power Method aka Simultaneous Iteration aka Subspace Iteration aka Orthogonal Iteration
- Block Krylov methods



- Let $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathcal{K}_q = \left[\mathbf{G}, (\mathbf{X}^T\mathbf{X}) \cdot \mathbf{G}, (\mathbf{X}^T\mathbf{X})^2 \cdot \mathbf{G}, \ldots, \underbrace{(\mathbf{X}^T\mathbf{X})^q \cdot \mathbf{G}}\right]$

Orthonormalize $\rightarrow \approx V_k$

Runtime: $O\left(\text{nnz}(\mathbf{X}) \cdot k \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ to obtain a nearly optimal low-rank approximation.

$[(X^TX)g \quad (X^TX)^2 g \quad \ldots \quad]$

$[G, \underline{(X^TX) \cdot G}]$ $\quad [G, (X^TX) \cdot \text{orth}(G)]$

What do you think a stochastic version of Krylov subspace method would look like?

$\to q_i q_i^T x$

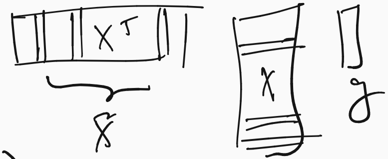
$\underbrace{2A^TAx}$ $- A^Tb$

Oja's Method

$$\mathcal{K}_q = \left[\mathbf{g}, (\mathbf{X}^T\mathbf{X})\cdot\mathbf{g}, (\mathbf{X}^T\mathbf{X})^2\cdot\mathbf{g}, \ldots, (\mathbf{X}^T\mathbf{X})^q\cdot\mathbf{g}\right]$$

$\underbrace{X^TX}\,g$

$O(nd)$ $O(nnz(X))$



$\tilde{X}^T\tilde{X}g$

$X^TXg \approx x_i x_i^T g$

$\downarrow$

one row from X

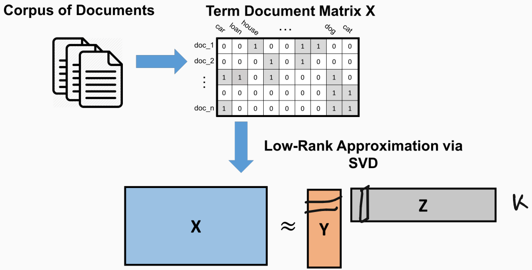
ENTITY EMBEDDINGS

$\boxed{A}$ y $A^T A$ $A^{-1} x$

Ax

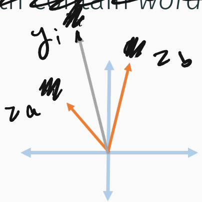
Applications of (partial) singular value decomposition:

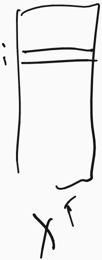
- Low-rank approximation (data compression)
- Denoising, in-painting, matrix completion
- Semantic embeddings

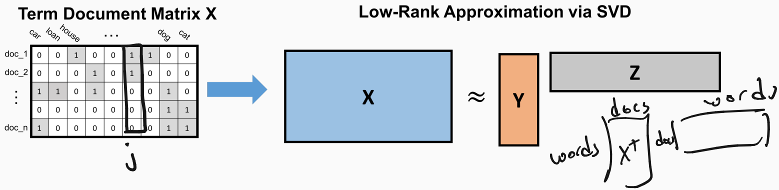


- $\langle \vec{y}_i, \vec{z}_a \rangle \approx 1$ when doc_i contains $word_a$.
- If doc_i ~~and doc_j both contain $word_a$~~, $\langle \vec{y}_i, \vec{z}_a \rangle \approx \langle \vec{y}_i, \vec{z}_b \rangle = 1$.

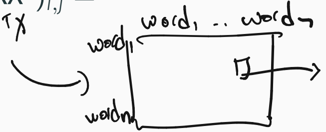
If doc_i contains word a and word b:







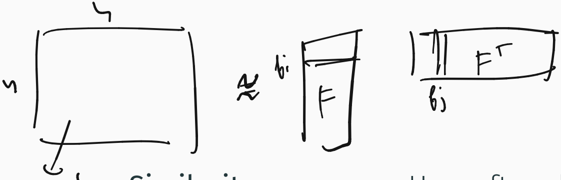
- The columns $\vec{z}_1, \vec{z}_2, \ldots$ give representations of words, with $\vec{z}_i$ and $\vec{z}_j$ tending to have high dot product if $word_i$ and $word_j$ appear in many of the same documents.
- **Z** corresponds to the top k right singular vectors: the eigenvectors of $\cancel{\mathbf{X}\mathbf{X}^T}$ X^TX. Intuitively, what is $\cancel{\mathbf{X}\mathbf{X}^T}$ X^TX?
- $(\cancel{\mathbf{X}\mathbf{X}^T})_{i,j} =$ X^TX



$\to (X^TX)_{ij} =$ # of documents both word i and j appear in

EXAMPLE: WORD EMBEDDING

Not obvious how to convert a word into a feature vector that captures the meaning of that word. Approach suggested by LSA: build a $d \times d$ symmetric "similarity matrix" $\mathbf{M}$ between words, and factorize: $\mathbf{M} \approx \mathbf{F}\mathbf{F}^T$ for rank k $\mathbf{F}$.

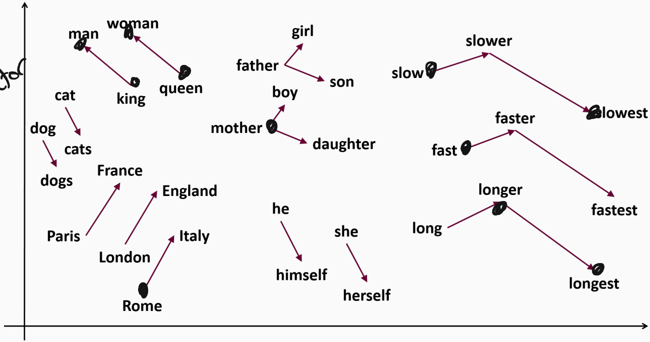


$\langle f_i, f_j \rangle \approx M_{ij}$

similarity between words

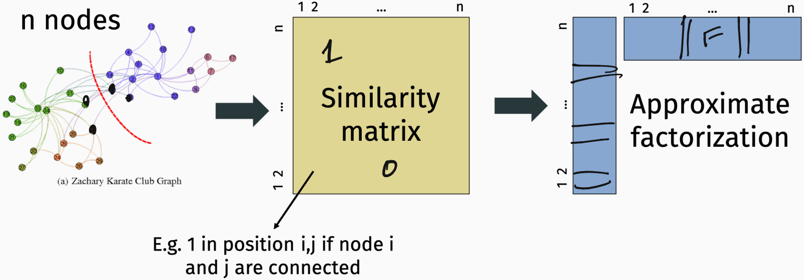
- **Similarity measures:** How often do $word_i, word_j$ appear in the same sentence, in the same window of w words, in similar positions of documents in different languages?
- Replacing $\mathbf{X}\mathbf{X}^T$ with these different metrics (sometimes appropriately transformed) leads to popular word embedding algorithms: `word2vec`, `GloVe`, etc.

word → vector

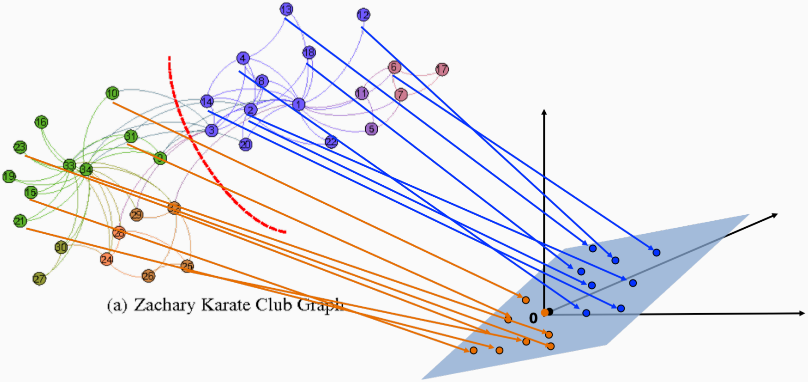


`word2vec` was originally described as a neural-network method, but Levy and Goldberg show that it is simply low-rank approximation of a specific similarity matrix. *Neural word embedding as implicit matrix factorization.*

Often data is represented as a graph and similarities can be obtained from that graph:

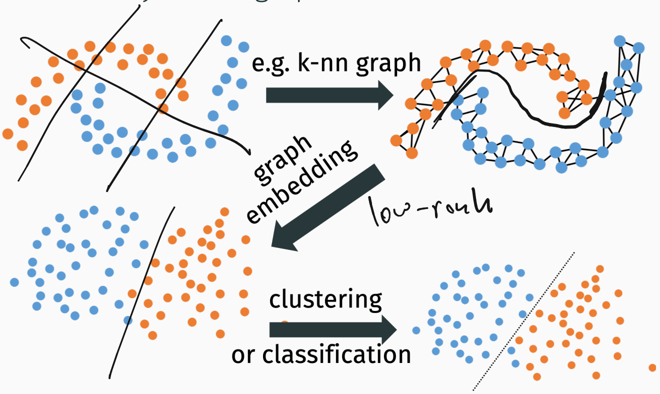


ENCODING GRAPH SIMILARITY



Social networks in 1970: “The network captures 34 members of a karate club, documenting links between pairs of members who interacted outside the club. During the study a conflict arose between the administrator ”John A” and instructor ”Mr. Hi” (pseudonyms), which led to the split of the club into two. Half of the members formed a new club around Mr. Hi; members from the other part found a new instructor or gave up karate. Based on collected data Zachary correctly assigned all but one member of the club to the groups they actually joined after the split.” – Wikipedia

Idea: Construct synthetic graph for data that is hard to cluster.



Spectral Clustering, Laplacian Eigenmaps, Locally linear embedding, Isomap, etc.

Spectral graph theory lets us formalize this heuristic idea.

Laplacians → matrix representation of a graph

Ajacency Matrix

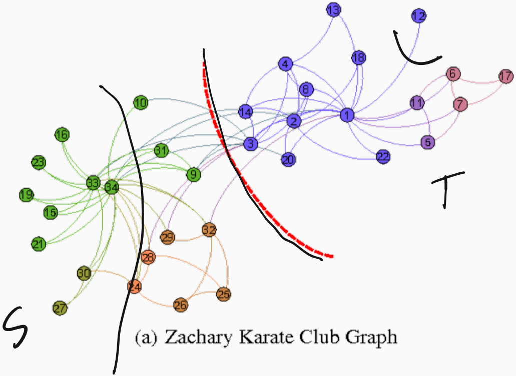
Powers of Ajacency matrix

Goal: Partition nodes along a cut that:

- Has few crossing edges: $|\{(u,v) \in E : u \in S, v \in T\}|$ is small.
- Separates large partitions: $\underline{|S|, |T|}$ are not too small.

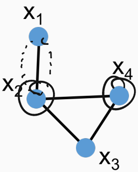
$|S| \approx |T|$

$\approx \frac{n}{2}$



(a) Zachary Karate Club Graph

For a graph with adjacency matrix **A** and degree matrix **D**, **L** = **D** − **A** is the graph Laplacian.



$$\overset{\mathbf{D}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}} - \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}} = \overset{\mathbf{L}}{\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}}$$

$\boxed{\mathbf{L} = \mathbf{B}^T\mathbf{B}}$ where B is the "edge-vertex incidence" matrix.

edges

1 .... 4 column for each node

$$\mathbf{B} = \begin{bmatrix} 1 & -1 & & \\ & 1 & & -1 \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \\ e_4^T \end{bmatrix} = \boxed{L}$$

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

$$\boxed{e_1 e_1^T} + e_2 e_2^T + e_3 e_3^T + e_4 e_4^T = B^T B$$

$x^T L x = x^T B^T B x = \|Bx\|_2^2$

Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

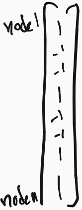
- $\mathbf{L}$ is positive semidefinite: $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x}$.
- $\mathbf{L} = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T$ where $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is $\mathbf{B}$'s SVD. Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{L}$.

  $L = V\Sigma \underbrace{U^T U}_{I} \Sigma V^T = V\Sigma^2 V^T$

- For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$ with $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$:

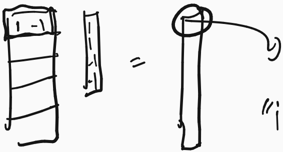
  $T : V \setminus S$

  - $\mathbf{c}^T\mathbf{L}\mathbf{c} = \sum_{(i,j)\in E}(\mathbf{c}(i) - \mathbf{c}(j))^2 = 4 \cdot cut(S, T)$.



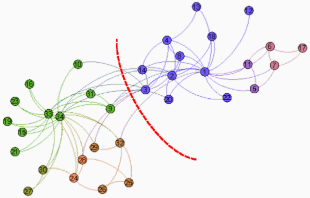
$c^T B^T B c = \|Bc\|_2^2$

\# of edges between S and T



0 if edge is "inside" S or T

2 otherwise



(a) Zachary Karate Club Graph

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$ with $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$:

want small → small cut

- $\mathbf{c}^T L \mathbf{c} = 4 \cdot cut(S, T)$.
- $\mathbf{c}^T \mathbf{1} = |T| - |S|$.

$|c^T 1| = \big| |T| - |S| \big|$

Want to minimize both $\mathbf{c}^T L \mathbf{c}$ (cut size) and $\mathbf{c}^T \mathbf{1}$ (imbalance).

Courant–Fischer min-max principle

singular vector

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the ~~eigenvectors~~ of $\mathbf{L}$.

$$\boxed{\mathbf{v}_1 = \arg\max_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}}$$

$\|LVV^T\|_F^2$

$= \|L\|_F^2 - \|L - LVV^T\|_F^2$

$$\mathbf{v}_2 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_3 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \mathbf{v}_2} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\vdots$$

$$\mathbf{v}_n = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_n = \arg\min_{\|\mathbf{v}\|=1} \mathbf{v}^T\mathbf{L}\mathbf{v}$$

$$\mathbf{v}_{n-1} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v}\perp\mathbf{v}_n} \mathbf{v}^T\mathbf{L}\mathbf{v}$$

$$\mathbf{v}_{n-2} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v}\perp\mathbf{v}_n, \mathbf{v}_{n-1}} \mathbf{v}^T\mathbf{L}\mathbf{v}$$

$$\vdots$$

$$\mathbf{v}_1 = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v}\perp\mathbf{v}_n, \ldots, \mathbf{v}_2} \mathbf{v}^T\mathbf{L}\mathbf{v}$$

SMALLEST LAPLACIAN EIGENVECTOR

The smallest eigenvector/singular vector $\mathbf{v}_n$ satisfies:

$$\mathbf{v}_n = \frac{1}{\sqrt{n}} \cdot \mathbf{1} = \underset{v \in \mathbb{R}^n \text{ with } \|\mathbf{v}\|=1}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

$$v_n = \begin{bmatrix} \frac{1}{\sqrt{n}} \\ \vdots \\ \frac{1}{\sqrt{n}} \end{bmatrix}$$

with $\mathbf{v}_n^T L \mathbf{v}_n = 0$.

$$L v_n = \vec{0}$$

scaled all ones

By Courant-Fischer, $\mathbf{v}_{n-1}$ is given by:

$$\mathbf{v}_{n-1} = \underset{\|\mathbf{v}\|=1,\ \mathbf{v}_n^T\mathbf{v}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

If $\mathbf{v}_{n-1}$ were binary $\{-1, 1\}^n$ it would have:

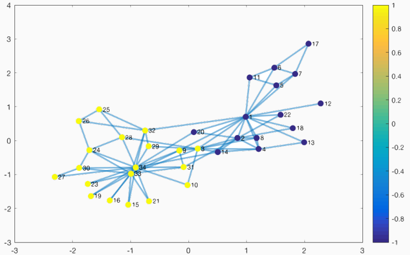
- $\mathbf{v}_{n-1}^T L \mathbf{v}_{n-1} = cut(S, T)$ as small as possible given that $\mathbf{v}_{n-1}^T \mathbf{1} = |T| - |S| = 0$.
- $\mathbf{v}_{n-1}$ would indicate the smallest perfectly balanced cut.

$\mathbf{v}_{n-1} \in \mathbb{R}^n$ is not generally binary, but still satisfies a 'relaxed' version of this property.

Find a good partition of the graph by computing

$$\mathbf{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\mathbf{v}\|=1,\ \mathbf{v}^T\mathbf{1}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

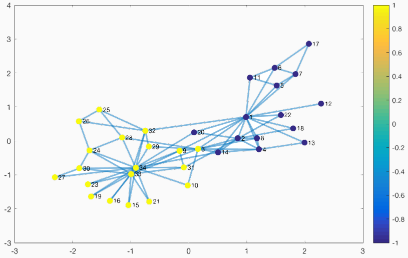
Set S to be all nodes with $\mathbf{v}_{n-1}(i) < 0$, and T to be all with $\mathbf{v}_{n-1}(i) \geq 0$.



Find a good partition of the graph by computing

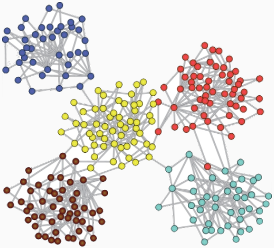
$$\mathbf{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\mathbf{v}\|=1,\ \mathbf{v}^T\mathbf{1}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

Set S to be all nodes with $\mathbf{v}_{n-1}(i) < 0$, and T to be all with $\mathbf{v}_{n-1}(i) \geq 0$.



The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\bar{L} = D^{-1/2} L D^{-1/2}$.

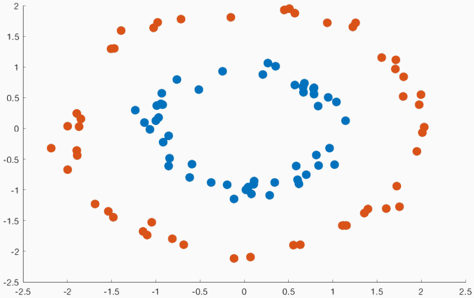
Important consideration: What to do when we want to split the graph into more than two parts?



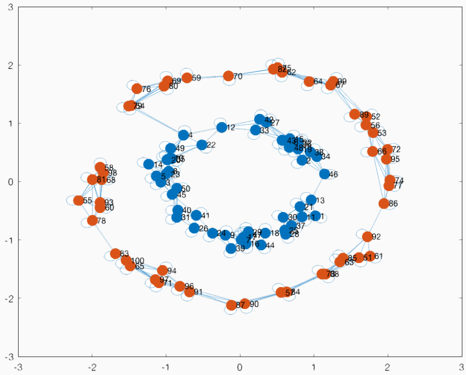
Spectral Clustering:

- Compute smallest k eigenvectors $\mathbf{v}_{n-1}, \ldots, \mathbf{v}_{n-k}$ of $\mathbf{L}$.
- Represent each node by its corresponding row in $\mathbf{V} \in \mathbb{R}^{n \times k}$ whose rows are $\mathbf{v}_{n-1}, \ldots \mathbf{v}_{n-k}$.
- Cluster these rows using k-means clustering (or really any clustering method).

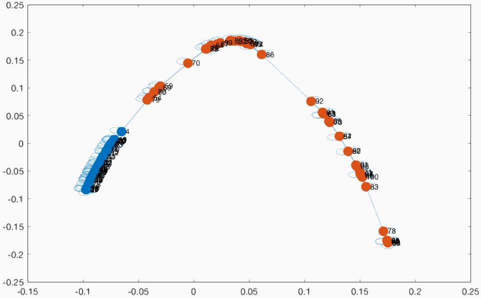
Original Data: (not linearly separable)



k-Nearest Neighbors Graph:



Embedding with eigenvectors $\mathbf{v}_{n-1}, \mathbf{v}_{n-2}$: (linearly separable)



So far: Showed that spectral clustering partitions a graph along a small cut between large pieces.

- No formal guarantee on the 'quality' of the partitioning.
- Would be difficult to analyze for general input graphs.

Common approach: Give a natural generative model for which produces random but realistic inputs and analyze how the algorithm performs on inputs drawn from this model.

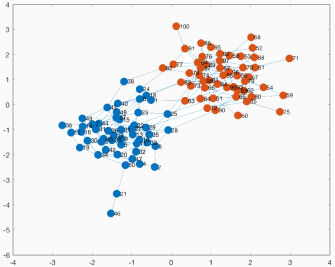
- Very common in algorithm design for data analysis/machine learning (can be used to justify ℓ_2 linear regression, k-means clustering, PCA, etc.)

Ideas for a generative model for graphs that would allow us to understand partitioning?

Stochastic Block Model (Planted Partition Model):

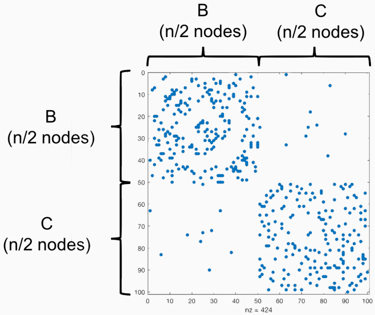
Let $G_n(p, q)$ be a distribution over graphs on n nodes, split equally into two groups B and C, each with $n/2$ nodes.

- Any two nodes in the same group are connected with probability p (including self-loops).
- Any two nodes in different groups are connected with prob. $q < p$.

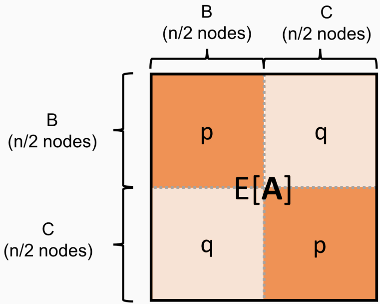


Let G be a stochastic block model graph drawn from $G_n(p, q)$.

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G. What is $\mathbb{E}[\mathbf{A}]$?

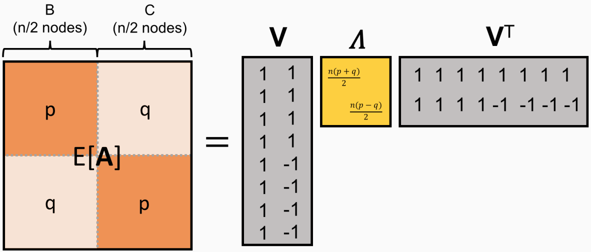


Letting G be a stochastic block model graph drawn from $G_n(p,q)$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ be its adjacency matrix. $(\mathbb{E}[\mathbf{A}])_{i,j} = p$ for i,j in same group, $(\mathbb{E}[\mathbf{A}])_{i,j} = q$ otherwise.



What are the eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{A}]$?

Letting G be a stochastic block model graph drawn from $G_n(p, q)$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix, what are the eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{A}]$?



- $\vec{v}_1 = \vec{1}$ with eigenvalue $\lambda_1 = \frac{(p+q)n}{2}$.
- $\vec{v}_2 = \chi_{B,C}$ with eigenvalue $\lambda_2 = \frac{(p-q)n}{2}$.
- $\chi_{B,C}(i) = 1$ if $i \in B$ and $\chi_{B,C}(i) = -1$ for $i \in C$.

If we compute $\vec{v}_2$ then we recover the communities B and C!

Letting G be a stochastic block model graph drawn from $G_n(p, q)$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix and $\mathbf{L}$ be its Laplacian, what are the eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{L}]$?

Upshot: The second small eigenvector of $\mathbb{E}[\mathbf{L}]$ is $\chi_{B,C}$ – the indicator vector for the cut between the communities.

- If the random graph G (equivilantly $\mathbf{A}$ and $\mathbf{L}$) were exactly equal to its expectation, partitioning using this eigenvector would exactly recover communities B and C.

How do we show that a matrix (e.g., $\mathbf{A}$) is close to its expectation? **Matrix concentration inequalities.**

- Analogous to scalar concentration inequalities like Markovs, Chebyshevs, Bernsteins.
- Random matrix theory is a very recent and cutting edge subfield of mathematics that is being actively applied in computer science, statistics, and machine learning.

Matrix Concentration Inequality: If $p \geq O\left(\frac{\log^4 n}{n}\right)$, then with high probability

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

where $\|\cdot\|_2$ is the matrix spectral norm (operator norm).

For $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\|\mathbf{X}\|_2 = \max_{z \in \mathbb{R}^d : \|z\|_2 = 1} \|\mathbf{X}z\|_2$.

Exercise: Show that $\|\mathbf{X}\|_2$ is equal to the largest singular value of $\mathbf{X}$. For symmetric $\mathbf{X}$ (like $\mathbf{A} - \mathbb{E}[\mathbf{A}]$) show that it is equal to the magnitude of the largest magnitude eigenvalue.

For the stochastic block model application, we want to show that the second eigenvectors of $\mathbf{A}$ and $\mathbb{E}[\mathbf{A}]$ are close. How does this relate to their difference in spectral norm?

Davis-Kahan Eigenvector Perturbation Theorem: Suppose $\mathbf{A}, \overline{\mathbf{A}} \in \mathbb{R}^{d \times d}$ are symmetric with $\|\mathbf{A} - \overline{\mathbf{A}}\|_2 \leq \epsilon$ and eigenvectors $v_1, v_2, \ldots, v_d$ and $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_d$. Letting $\theta(v_i, \bar{v}_i)$ denote the angle between v_i and $\bar{v}_i$, for all i:

$$\sin[\theta(v_i, \bar{v}_i)] \leq \frac{\epsilon}{\min_{j \neq i} |\lambda_i - \lambda_j|}$$

where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of $\overline{\mathbf{A}}$.

The error gets larger if there are eigenvalues with similar magnitudes.

EIGENVECTOR PERTURBATION

$$\overset{\mathbf{A}}{\begin{bmatrix} 1+\varepsilon & 0 \\ 0 & 1 \end{bmatrix}} - \overset{\bar{\mathbf{A}}}{\begin{bmatrix} 1 & 0 \\ 0 & 1+\varepsilon \end{bmatrix}} = \overset{\mathbf{A}-\bar{\mathbf{A}}}{\begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix}}$$

APPLICATION TO STOCHASTIC BLOCK MODEL

Claim 1 (Matrix Concentration): For $p \geq O\left(\frac{\log^4 n}{n}\right)$,

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Claim 2 (Davis-Kahan): For $p \geq O\left(\frac{\log^4 n}{n}\right)$,

$$\sin\theta(v_2, \bar{v}_2) \leq \frac{O(\sqrt{pn})}{\min_{j \neq i} |\lambda_i - \lambda_j|} \leq \frac{O(\sqrt{pn})}{(p-q)n/2} == O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right)$$

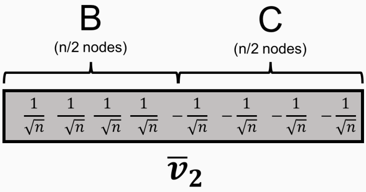
Recall: $\mathbb{E}[\mathbf{A}]$, has eigenvalues $\lambda_1 = \frac{(p+q)n}{2}$, $\lambda_2 = \frac{(p-q)n}{2}$, $\lambda_i = 0$ for $i \geq 3$.

$$\min_{j \neq i} |\lambda_i - \lambda_j| = \min\left(qn, \frac{(p-q)n}{2}\right).$$

Typically, $\frac{(p-q)n}{2}$ will be the minimum of these two gaps.

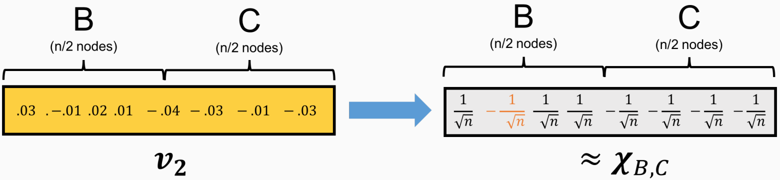
So Far: $\sin\theta(v_2, \bar{v}_2) \leq O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right)$. What does this give us?

- Can show that this implies $\|v_2 - \bar{v}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$ (exercise).
- $\bar{v}_2$ is $\frac{1}{\sqrt{n}}\chi_{B,C}$: the community indicator vector.



- Every i where $v_2(i), \bar{v}_2(i)$ differ in sign contributes $\geq \frac{1}{n}$ to $\|v_2 - \bar{v}_2\|_2^2$.
- So they differ in sign in at most $O\left(\frac{p}{(p-q)^2}\right)$ positions.

Upshot: If G is a stochastic block model graph with adjacency matrix $\mathbf{A}$, if we compute its second large eigenvector v_2 and assign nodes to communities according to the sign pattern of this vector, we will correctly assign all but $O\left(\frac{p}{(p-q)^2}\right)$ nodes.



- Why does the error increase as q gets close to p?
- Even when $p - q = O(1/\sqrt{n})$, assign all but an $O(n)$ fraction of nodes correctly. E.g., assign 99% of nodes correctly.