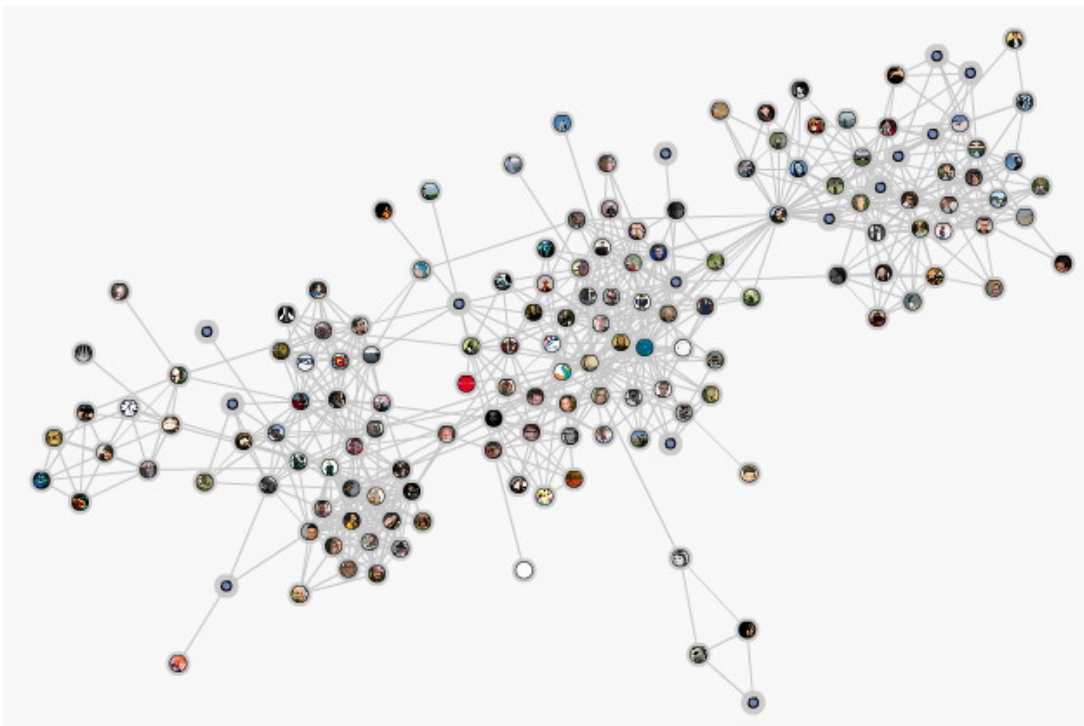
Matrices and Graphs – Spectral Techniques

PPT by Brandon Fain

Outline

- Motivating Problem: Community Detection
- Spectral Clustering and Conductance
- Graph Laplacian

Motivating Problem: Community Detection



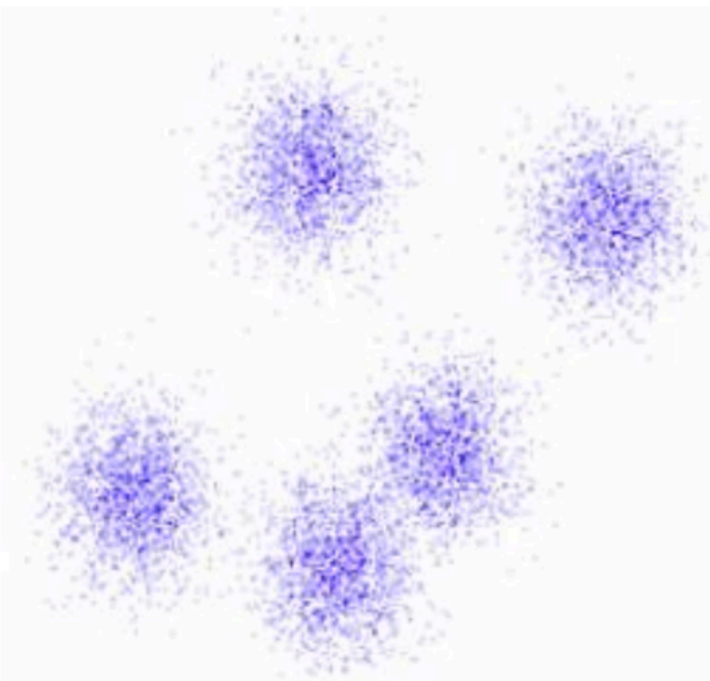
Given a social network, how do you find the strongly connected communities?

Corollary question: How would you suggest friends to a user?

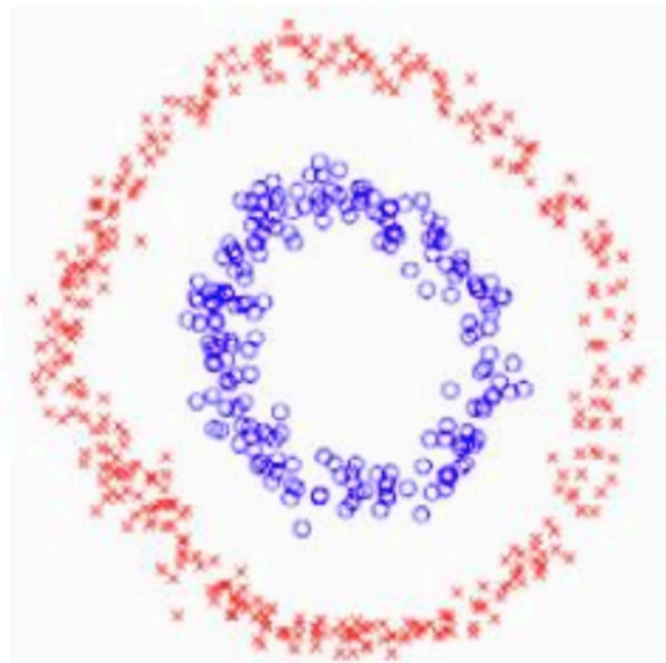
Motivating Community Detection

- But the question is much broader than this: it's about **clustering**, the fundamental task in unsupervised machine learning.
- Stated loosely, clustering is about creating a *partition* of the data so that points within a partition are more *similar* to one another than points outside of the partition.
- But what is the right notion of "similar?" For geometric data, perhaps it is geometric distance or compactness. What about for a graph?

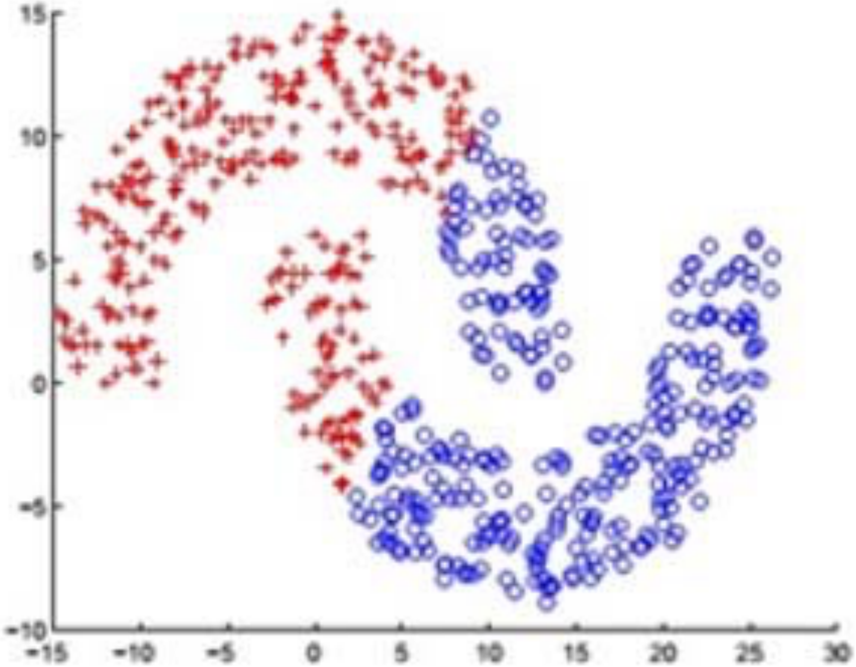
- Compactness, e.g., k-means, mixture models
- Connectivity, e.g., spectral clustering



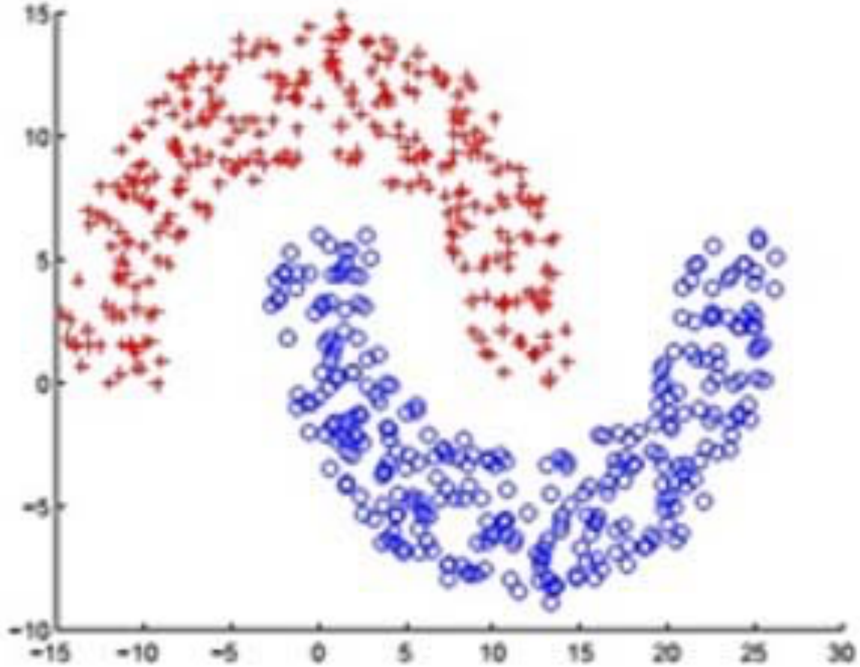
Compactness



Connectivity



(a) K-means



(b) Spectral Clustering

Community Detection – Spectral Clustering

- In fact, as you can probably see, there are other applications where you might like to cluster based on "community structure" in a graph: it captures the idea of similarity by connectivity!

- In such applications, one typically:
  - Given a similarity measure S(), draw a graph by placing an edge between data points x and y with $S(x,y) < t$, for some threshold t.
  - Use **spectral clustering** to partition the graph based on its connectivity (i.e., ignore everything else about the data).

Community Detection - Spectral Clustering

- Questions:
  - How do we measure the quality of a cluster?
  - How do we compute such a clustering?

- This class and next, we'll try to answer these questions.

- Today, we'll focus on the first question, and reviewing some tools we will need to answer the second next week.

Outline

- ~~Motivating Problem: Community Detection~~

- Spectral Clustering and Conductance

- Graph Laplacian

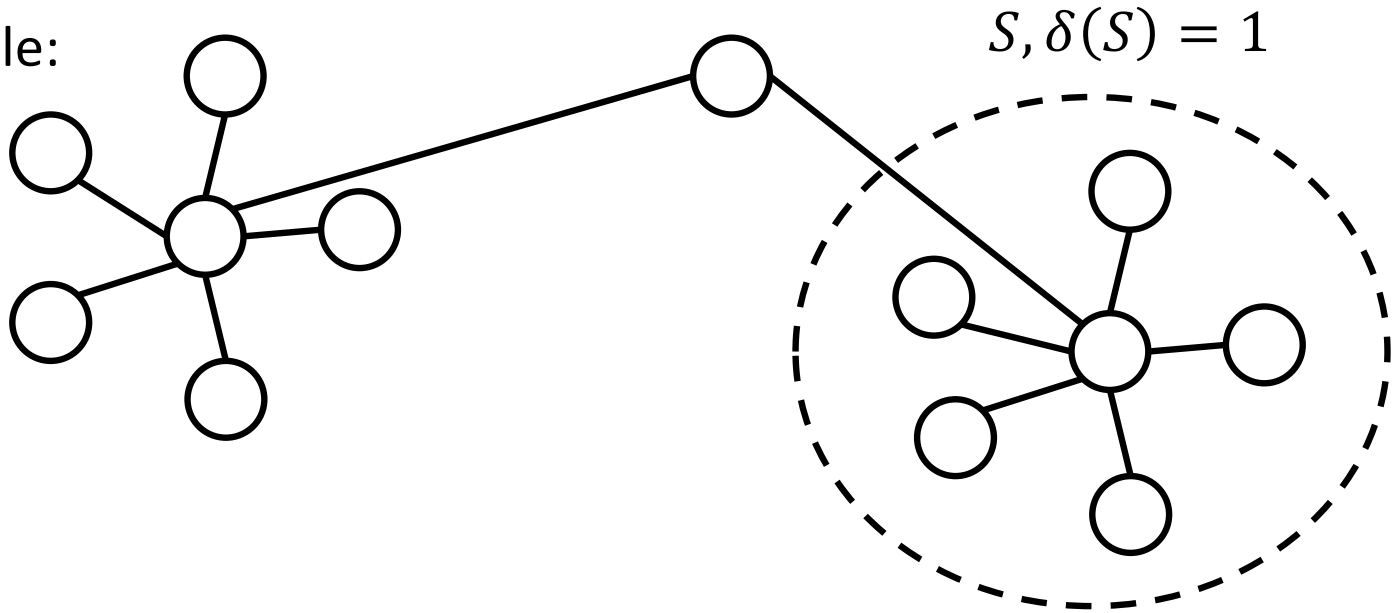
Spectral Clustering – Measuring Quality

- In the simplest case, suppose we just want to generate a 2-partition (i.e., split the data into two sets). What objective should we minimize?

- Take 0: Compute the minimum cut in the graph.
  - Let $G = (V, E)$ be an undirected graph.
  - Let $S \subseteq V$ denote a cut in the graph.
  - Let $\delta(S) \coloneqq |\{(u, v) \in E : u \in S, v \notin S\}|$.
  - In the minimum cut problem, we want to find a cut S that minimizes $\delta(S)$.

Take 0 – Minimum Cut

- For example:



- Looks pretty reasonable!
- What is wrong with this notion, and how would you fix it?

Take 1 – Normalized Minimum Cut

- Somehow, we want to find a *large* partition of vertices with a small number of cut edges.

- Take 1: Compute the minimum *normalized* cut in the graph.
  - Again Let $G = (V, E)$ be an undirected graph and $S \subseteq V$ denote a cut in the graph, with $\delta(S) := |\{(u, v) \in E: u \in S, v \notin S\}|$.
  - In the minimum *normalized* cut problem, we want to find a cut S that minimizes

$$\frac{\delta(S)}{|S| \cdot |V - S|}$$

- (Sometimes called the *isoperimetric ratio*)

Take 2 - Conductance

- The isoperimetric ratio should look very familiar to you.

- Recall that last week we discussed **conductance** as a property of graphs related to how quickly power iteration computes the stationary distribution of a random walk on the graph.

- Let S be a cut. Let $Vol(S) = \sum_{i \in S} d_i$, where d_i is the degree of node i. The **conductance** of S is

$$\phi(S) = \frac{\delta(S)}{\min(Vol(S), Vol(V - S))}.$$

Conductance

- Last week, we were interested in graphs for which power iteration could quickly converge to find the stationary distribution of a random walk.

- For that, we wanted a graph for which **all** cuts had **high** conductance.

- For the spectral partitioning problem, we will look for a particular cut in the graph with very **low** conductance (which means there are a lot of internal edges, but very few cut edges).

Outline

- ~~Motivating Problem: Community Detection~~

- ~~Spectral Clustering and Conductance~~

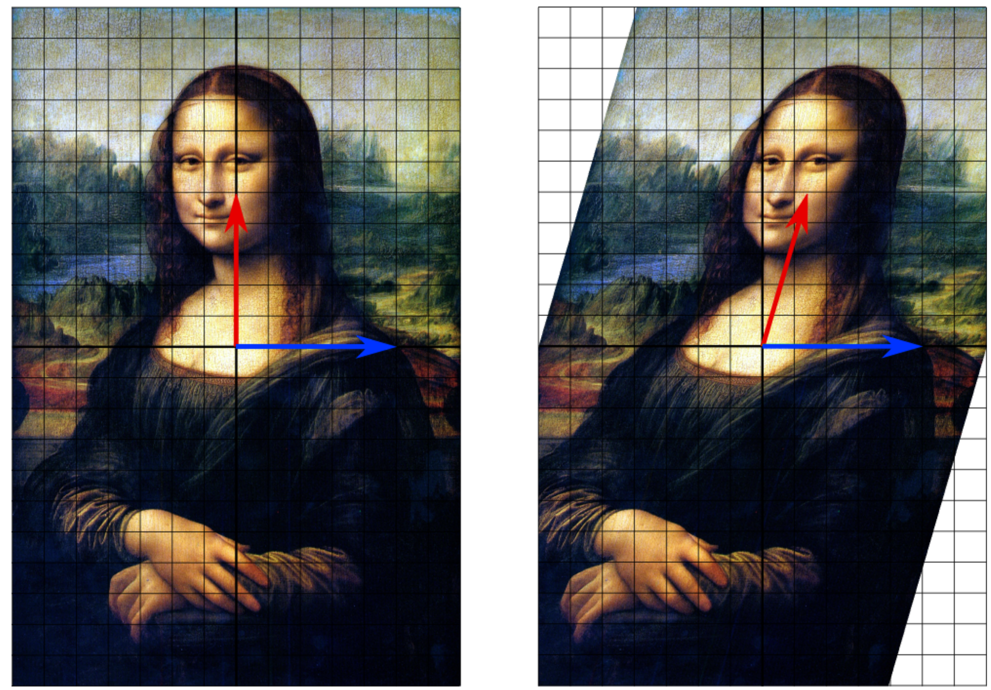
- Graph Laplacian

Eigenvectors and Eigenvalues

- Consider a matrix M. We say that λ is an **eigenvalue** of M with associated **eigenvector** $\vec{v}$ if

$$M\,\vec{v} = \lambda\,\vec{v}.$$

- Think of M as a linear function or linear transformation. An eigenvector is an input that only changes in scale, *not* direction.



Laplacian Matrix

- The particular matrix we are interested in is the **graph Laplacian**, defined as

$$L = D - A$$

where D is the diagonal matrix with $D_{ii} = d_i$ and $D_{ij} = 0$ for $i \neq j$, and A is the adjacency matrix.

- There is also a **normalized graph Laplacian** defined as

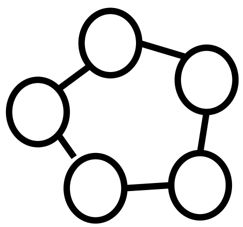
$$\hat{L} = I - D^{-\frac{1}{2}} A \, D^{-\frac{1}{2}}.$$

It is helpful to get some intuition by trying to write down some graph Laplacians, so let’s try a few.

Laplacian Matrix

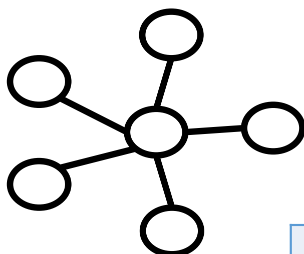
- Write down the Laplacian (un-normalized) of the following:

1. A 5-cycle



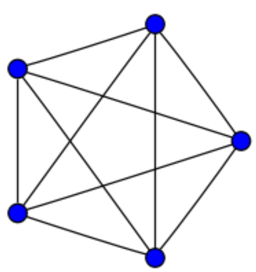
2	-1	0	0	-1
-1	2	-1	0	0
0	-1	2	-1	0
0	0	-1	2	-1
-1	0	0	-1	2

2. A 5-star



5	-1	-1	-1	-1	-1
-1	1	0	0	0	0
-1	0	1	0	0	0
-1	0	0	1	0	0
-1	0	0	0	1	0
-1	0	0	0	0	1

3. K_5



4	-1	-1	-1	-1
-1	4	-1	-1	-1
-1	-1	4	-1	-1
-1	-1	-1	4	-1
-1	-1	-1	-1	4

Laplacian Matrix

- Why do we care about this matrix? The first intuition is to note that

$$(Lv)_i = d_i v_i - \sum_{j:(i,j)\in E} v_j = \sum_{j:(i,j)\in E} v_i - v_j \,.$$

Therefore,

$$v^t L v = \sum_{i\in V} v_i \left(\sum_{j:(i,j)\in E} v_i - v_j \right) = \sum_{(i,j)\in E} v_i \left(v_i - v_j\right)$$

$$v^t L v = \sum_{(i,j)\in E, i<j} \left(v_i - v_j\right)^2$$

Laplacian Matrix

- In other words, given a graph with Laplacian L, and a vector v that assigns a value to every vertex in the graph, $v^t L v$ is the sum of squared differences *of neighbors* in the graph.

- So the Laplacian matrix is definitely recording *something* about connectivity.

- But actually, this simple fact allows us to show a few other connections.

Laplacian Matrix

- Given $v^t L v = \sum_{(i,j)\in E, i<j} (v_i - v_j)^2$.
- Then L is a positive semidefinite matrix (by definition). It follows that all of the eigenvalues of L are nonnegative (an excellent exercise).
- Note that it's also clear that $L\vec{1} = (0)\vec{1}$, so the smallest eigenvalue is 0, and $\vec{1}$ is an associated eigenvector.
- In fact, there is an important connection between this eigenvalue and the graph.

Laplacian Matrix

- **Claim.** The number of connected components in a graph equals the number of orthonormal eigenvectors associated with eigenvalue 0 of the graph Laplacian.
- **Proof.** Suppose there are k connected components in the graph. For each component $S \subseteq V$, define $\overrightarrow{u^S}$ such that $u_i^S = 1$ if $i \in S$, else 0.
- Clearly these vectors are orthogonal, and since $(Lv)_i = \sum_{j:(i,j)\in E} v_i - v_j$, we know that $L\,\overrightarrow{u^S} = (0)\,\overrightarrow{u^S}$. So there are at least k such eigenvectors.
- But any other eigenvector also has to take the same value on every vertex of a connected component, so any other such vector is not orthogonal to these. So there are just k such eigenvectors.

Laplacian Matrix

- **Corollary.** Given the eigenvectors of the graph Laplacian with eigenvalue 0, one can read off the connected components of the graph.
  - In particular, for each of these 0 eigenvalue eigenvectors, you just have to pick the vertices for which the eigenvector has the same value.

- So far then, we have built a purely algebraic way of computing the connected components of a graph.
- But, of course, you learned how to do this (much more efficiently) with depth first search last week. So who cares?

Laplacian and Spectral Clustering

- What we really care about for clustering and community detection: finding low conductance cuts in the graph.

- Recall that $(Lv)_i = \sum_{j:(i,j)\in E} v_i - v_j$. So if v is an eigenvector associated with a small (but positive) eigenvalue, it assigns very similar values to neighboring vertices.

- Of course, if the graph is connected, we already know that the eigenvector for the smallest eigenvalue (0) is just $\vec{1}$, so that isn't very helpful.

Laplacians and Spectral Clustering

- What if we take the eigenvalue associated with the *second* smallest eigenvalue (for a connected graph)?
- Then we take a large subset of vertices to which this vector assigns very similar values.
- Such a subset of vertices should be more likely to be neighbors.
- How do we formalize this idea? Can we prove anything about its performance? (Stay tuned for next time!)