CSE 6740: Computational Data Analysis

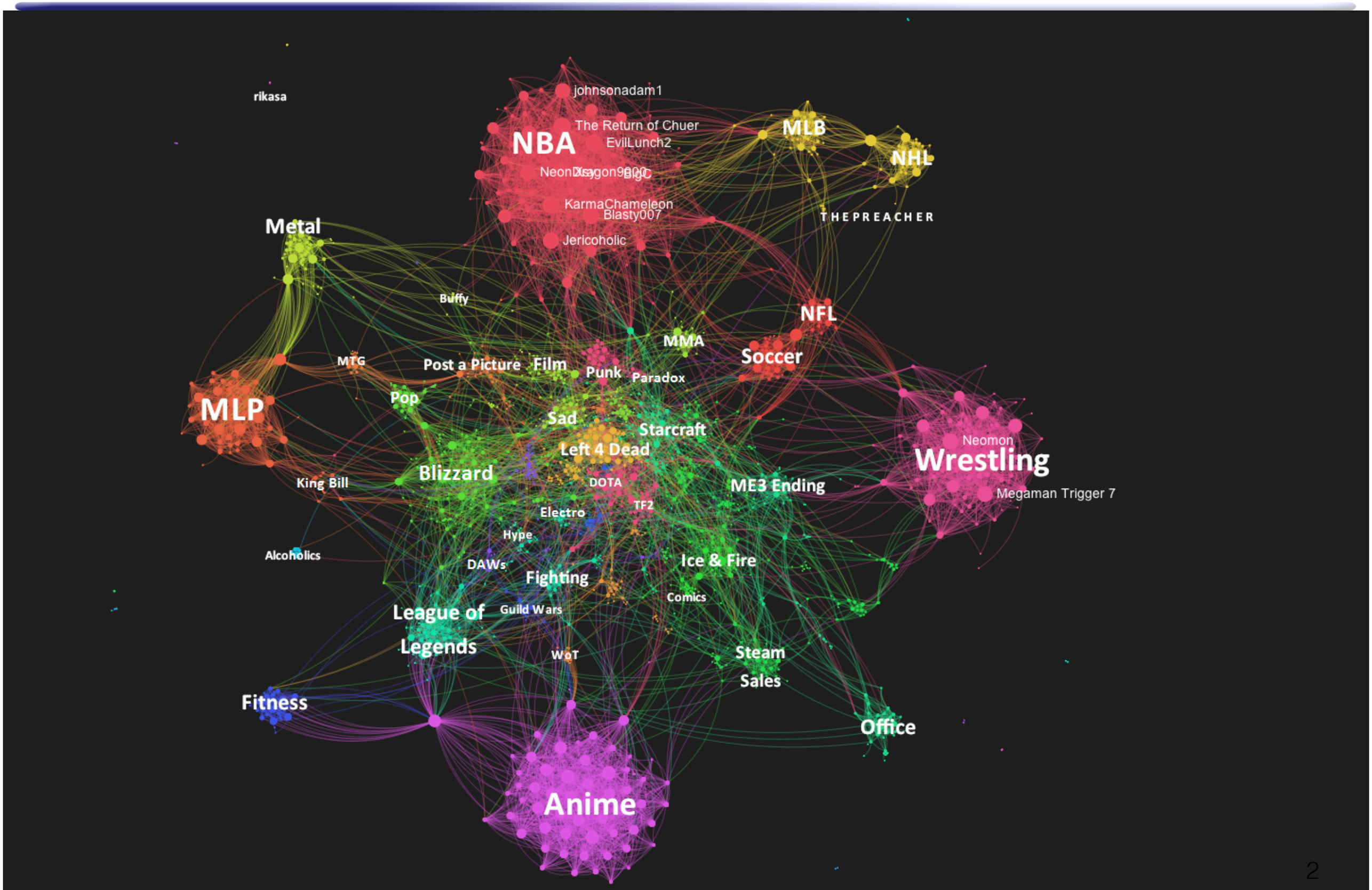
Spring 2026

Spectral clustering, dimensionality reduction & principle component analysis

Anqi Wu

01/22

How about clustering nodes in social networks

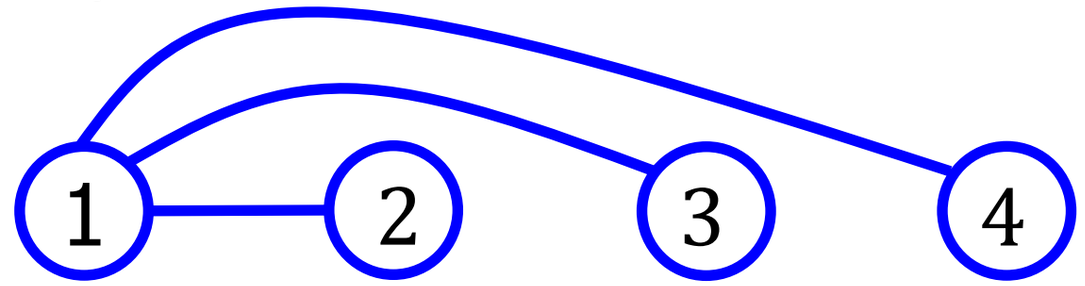


Spectral clustering algorithm

- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$

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

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



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

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Step 2: form a special matrix $L = D - A$, the graph Laplacian

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

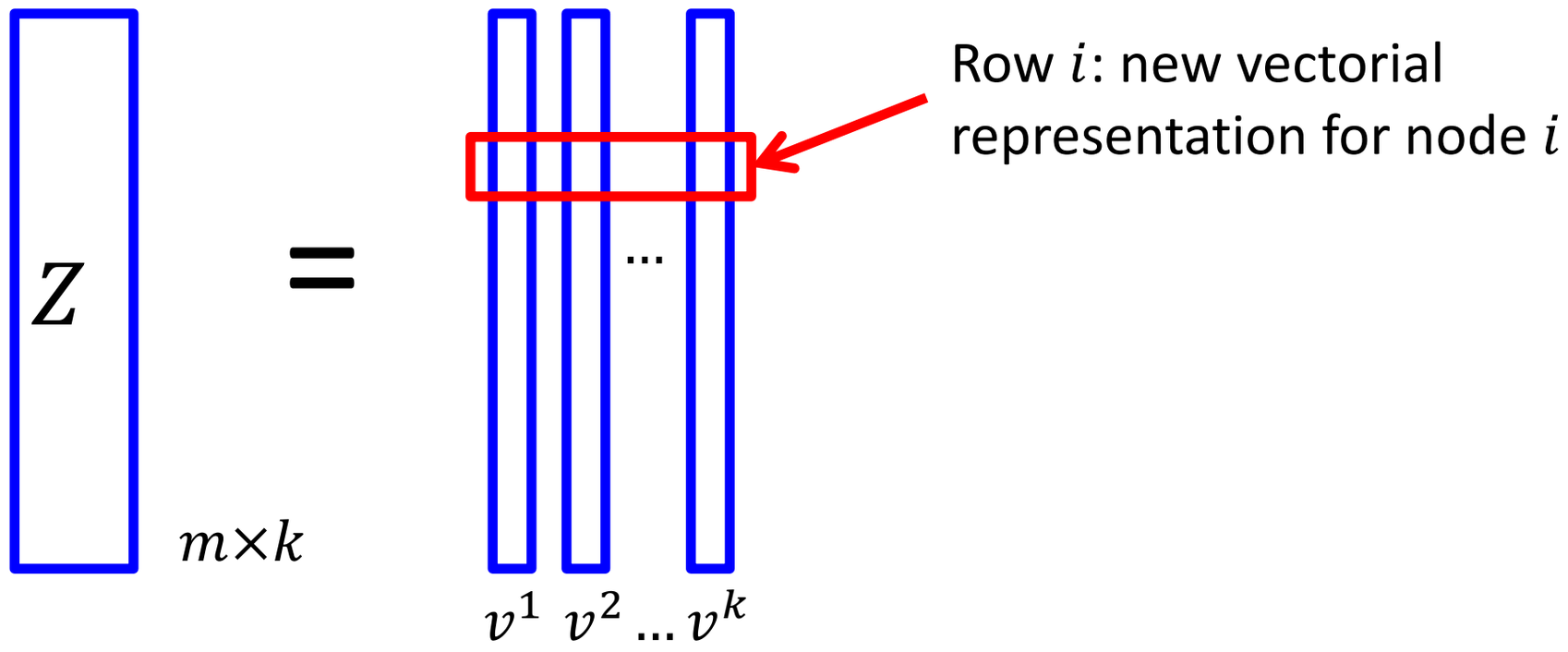
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Spectral clustering algorithm (cont.)

- Step 3: compute k eigenvectors, $v^1, v^2, \ldots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)

$$Lv^1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} v^1 \stackrel{?}{=} \lambda_1 v^1$$

- Step 4: run kmeans algorithm on $Z = \left(v^1, v^2, \ldots v^k\right)$ by treating each row as a new data point



Why Laplacian matrix L?

Step 1: Clustering as a Cut Problem on a Graph

We want to partition nodes into two groups S and $\bar{S}$. For a weighted graph with edge weights w_{ij}, define the cut:

$$\mathrm{Cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij}.$$

Goal: make cross-group connections as weak as possible.

Why Laplacian matrix L?

Step 2: Encode the Partition with a Discrete Indicator

Define a discrete indicator vector $x \in \{+1, -1\}^n$:

$$x_i = \begin{cases} +1 & i \in S \\ -1 & i \in \bar{S} \end{cases}$$

Then for any pair (i, j):

$$(x_i - x_j)^2 = \begin{cases} 0 & \text{same group} \\ 4 & \text{different groups} \end{cases}$$

So $(x_i - x_j)^2$ acts like a "soft switch" that detects boundary edges.

Why Laplacian matrix L?

Step 3: A Cut-Counting Objective (No L Yet)

Consider the objective

$$J(x) \;=\; \frac{1}{2}\sum_{i,j} w_{ij}(x_i - x_j)^2.$$

If $x \in \{+1, -1\}^n$, only cross-group pairs contribute:

$$J(x) \;=\; \frac{1}{2}\sum_{(i,j)\in \mathrm{cut}} w_{ij}\cdot 4 \;=\; 2\,\mathrm{Cut}(S, \bar{S}).$$

Thus, for discrete indicators,

$$\min_{x\in\{+1,-1\}^n} J(x) \quad\Longleftrightarrow\quad \min \mathrm{Cut}(S, \bar{S}).$$

Why Laplacian matrix L?

Step 4: Why Relax to Continuous x?

The discrete optimization

$$\min_{x \in \{+1,-1\}^n} J(x)$$

is combinatorial and hard.
Spectral methods relax the constraint:

$$x \in \mathbb{R}^n,$$

while keeping the same objective form $J(x)$.
We also need constraints to avoid trivial solutions:

$$\|x\|^2 = 1, \qquad x \perp \mathbf{1}.$$

Why Laplacian matrix L?

Step 5: Now Introduce the Laplacian

Let W be the weighted adjacency matrix, and D be the degree matrix:

$$D_{ii} = \sum_j w_{ij}, \qquad L = D - W.$$

A key identity shows the objective can be written compactly:

$$\frac{1}{2} \sum_{i,j} w_{ij}(x_i - x_j)^2 \;=\; x^\top L x.$$

So $J(x) = x^\top L x$. This is not a new objective: it is exactly the same cut-counting form.

Why Laplacian matrix L?

Step 6: Continuous Minimization Leads to Eigenvectors

We solve the relaxed problem:

$$\min_{x \in \mathbb{R}^n} x^\top L x \quad \text{s.t.} \quad \|x\|^2 = 1, \;\; x \perp \mathbf{1}.$$

The solution is the eigenvector corresponding to the smallest eigenvalue that is not the trivial constant eigenvector. This is the second smallest eigenvector of L.

Why smallest eigenvalue?

Problem Statement

Let $L = L^\top \in \mathbb{R}^{n \times n}$ be a symmetric matrix. We want to solve

$$\min_{\|x\|=1} \; x^\top L x.$$

Claim:

$$\min_{\|x\|=1} x^\top L x = \lambda_1,$$

where λ_1 is the smallest eigenvalue of L, and the minimizer is an eigenvector associated with λ_1.

Why smallest eigenvalue?

Step A: Spectral Decomposition

Because L is symmetric, it is orthogonally diagonalizable:

$$L = U\Lambda U^\top, \quad U = [u_1, \ldots, u_n], \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n),$$

with

$$U^\top U = I, \qquad \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Why smallest eigenvalue?

Step B: Expand x in the Eigenbasis

Write any $x \in \mathbb{R}^n$ as

$$x = \sum_{i=1}^{n} \alpha_i u_i.$$

Using orthonormality,

$$\|x\|^2 = x^\top x = \sum_{i=1}^{n} \alpha_i^2.$$

Thus the constraint $\|x\| = 1$ is equivalent to

$$\sum_{i=1}^{n} \alpha_i^2 = 1.$$

Why smallest eigenvalue?

Step C: Quadratic Form Becomes a Weighted Sum

Compute:

$$x^\top L x = \left(\sum_i \alpha_i u_i\right)^\top L \left(\sum_j \alpha_j u_j\right) = \sum_{i,j} \alpha_i \alpha_j \, u_i^\top L u_j.$$

Since $Lu_j = \lambda_j u_j$ and $u_i^\top u_j = \delta_{ij}$,

$$u_i^\top L u_j = u_i^\top (\lambda_j u_j) = \lambda_j \delta_{ij}.$$

Therefore,

$$x^\top L x = \sum_{i=1}^{n} \lambda_i \alpha_i^2.$$

So the original problem is equivalent to

$$\min_{\sum_i \alpha_i^2 = 1} \sum_{i=1}^{n} \lambda_i \alpha_i^2.$$

Why smallest eigenvalue?

Step D: Lower Bound via an Inequality (Method 1)

Because $\lambda_i \geq \lambda_1$ for all i and $\alpha_i^2 \geq 0$,

$$\sum_{i=1}^{n} \lambda_i \alpha_i^2 \;\geq\; \sum_{i=1}^{n} \lambda_1 \alpha_i^2 = \lambda_1 \sum_{i=1}^{n} \alpha_i^2 = \lambda_1.$$

Thus for every feasible x with $\|x\| = 1$,

$$x^\top L x \geq \lambda_1.$$

So λ_1 is a universal lower bound on the objective value.

Why smallest eigenvalue?

Step D (continued): λ_1 Is the Achievable Minimum

We now show the lower bound is achieved.
Take $x = u_1$ (equivalently $\alpha_1 = 1$ and $\alpha_{i\neq 1} = 0$), which satisfies $\|x\| = 1$. Then

$$x^\top L x = u_1^\top L u_1 = u_1^\top (\lambda_1 u_1) = \lambda_1.$$

So the minimum value is at most λ_1, and together with the lower bound we conclude

$$\min_{\|x\|=1} x^\top L x = \lambda_1.$$

Why smallest eigenvalue?

Extra Argument by Contradiction

Assume λ_1 is not the minimum value of the optimization. Then there exists a smaller value $\lambda_1^* < \lambda_1$ achieved by some feasible x^* with $\|x^*\| = 1$:

$$(x^*)^\top L x^* = \lambda_1^* < \lambda_1.$$

But Step D already proved that for every feasible x,

$$x^\top L x \geq \lambda_1.$$

This implies $(x^*)^\top L x^* \geq \lambda_1$, a contradiction. Therefore λ_1 must be the achievable minimum.

Why smallest eigenvalue?

Minimizer Characterization

A minimizer can be chosen as an eigenvector corresponding to λ_1:

$$x^\star = \pm u_1.$$

Why smallest eigenvalue?

One-Sentence Add-on: Multiplicity Case

If the smallest eigenvalue has multiplicity m,

$$\lambda_1 = \lambda_2 = \cdots = \lambda_m < \lambda_{m+1},$$

then every unit vector in the subspace

$$\mathrm{span}\{u_1, \ldots, u_m\}$$

achieves the same minimum value λ_1.

Graph Laplacian example

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

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Property I of Graph Laplacian

- $L = D - A$

- The multiplicity of the eigenvalue 0 corresponds to the number of connected components in the graph

- Example

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$Lv_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$Lv_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Property II of Graph Laplacian

- $L = D - A$

- The eigenvectors with eigenvalue 0 contains cluster assignment information

- Example

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$Lv^1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$Lv^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

What if the graph has only 1 component

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

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Properties of Graph Laplacian

- $L = D - A$

- The smallest eigenvalue of L is 0, corresponding a constant eigenvector $\frac{1}{\sqrt{m}}1$

- Example

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\frac{1}{\sqrt{4}}L1 = \frac{1}{\sqrt{m}}\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

What if the graph has k components

- If a graph has k connected components (or k clusters)

- The graph Laplacian has k blocks

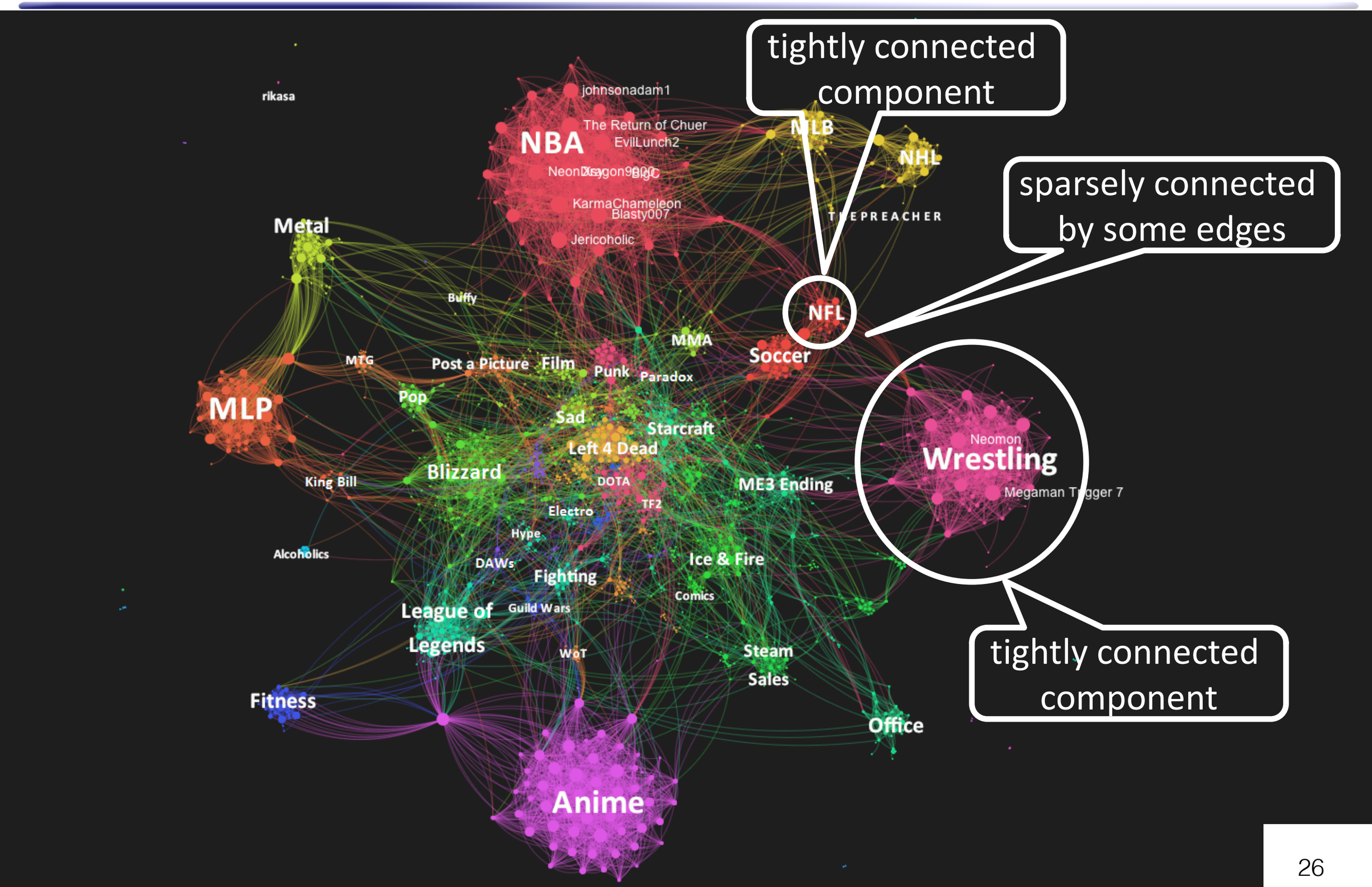
$$L = \begin{pmatrix} L_1 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_k \end{pmatrix}$$

- The graph Laplacian has k eigenvectors with zero eigenvalues

- Eigenvector 1 is constant in block 1, but 0 in other blocks;
  eigenvector 2 is constant in block 2, but 0 in other blocks;

  …

Real world social networks



In most real networks

- If a graph has k tightly connected components (or k clusters) with sparsely connected edges

- The graph Laplacian has approximately k blocks

- The graph Laplancian has k eigenvectors with small eigenvalues

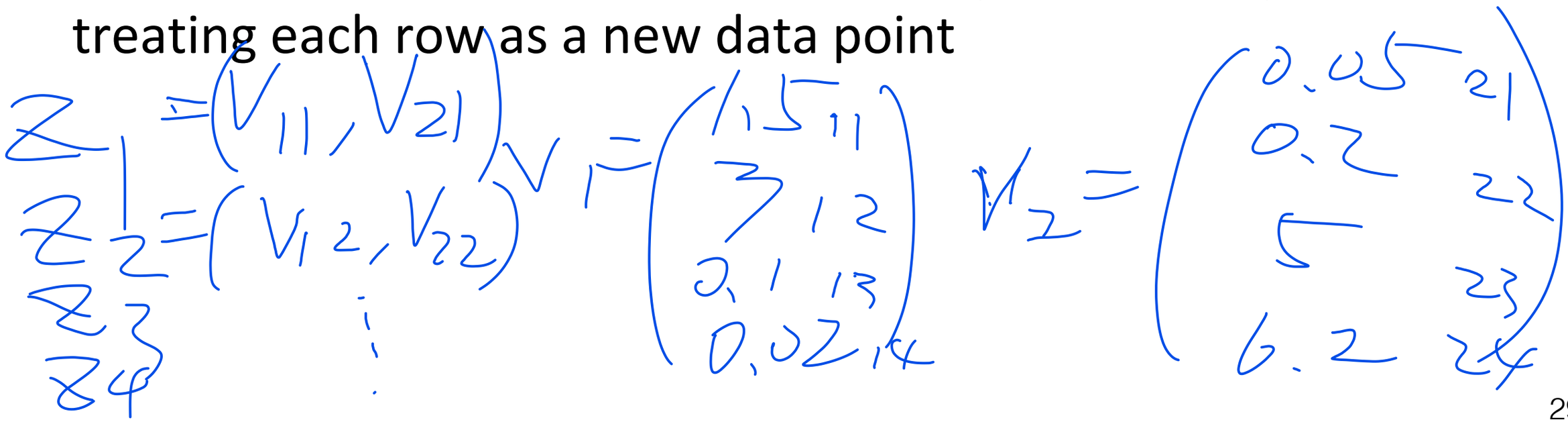
- Eigenvector 1 is approximately constant in block 1, but 0 in other blocks; eigenvector 2 …

High level idea of spectral clustering

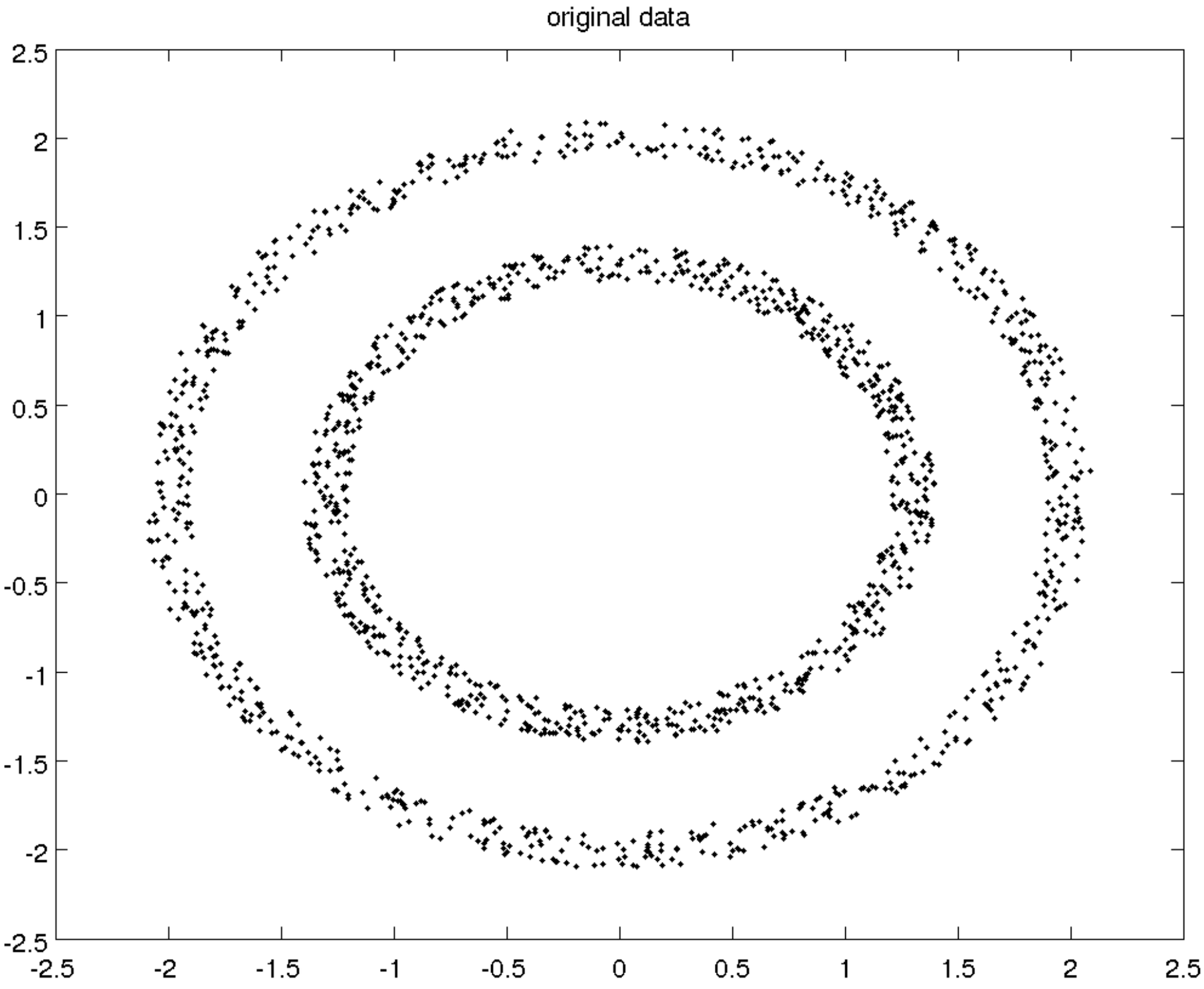
- Examine the properties of graph Laplacian for the perfect cases
  - The number of 0 eigenvalues corresponds to the number of connected components
  - Eigenvectors correspond to cluster assignment
- Then use the intuition from perfect cases to design algorithms for the imperfect case.
  - Eigenvectors not longer correspond exactly cluster indicator
  - Perform post processing to obtain cluster assignment

Summary of spectral clustering

- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 2: form a special matrix $L = D - A$, the graph Laplacian
- Step 3: compute k eigenvectors, $v^1, v^2, \ldots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)
- Step 4: run kmeans algorithm on $Z = \left(v^1, v^2, \ldots v^k\right)$ by treating each row as a new data point



How about this dataset?



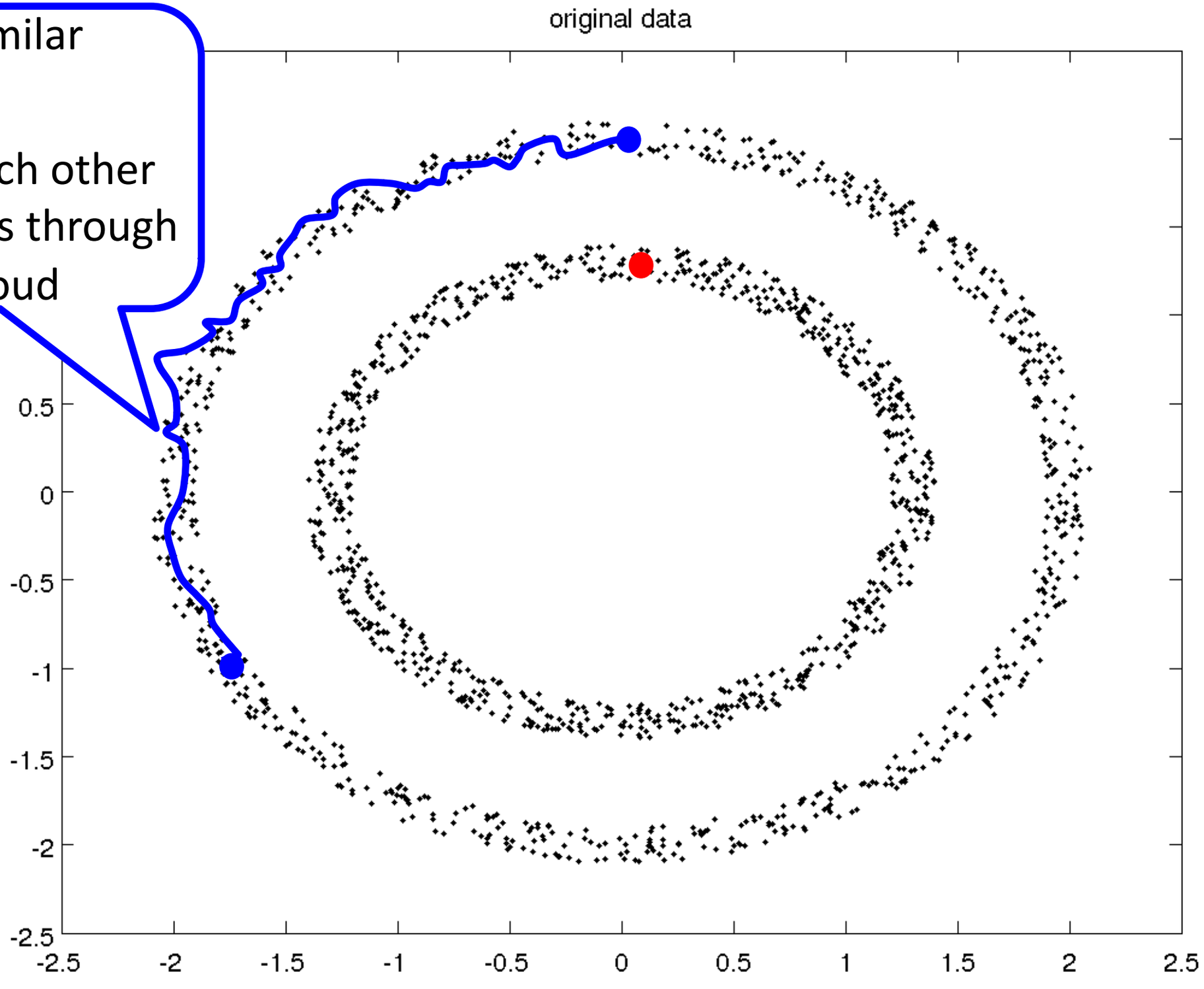
What's a reasonable similarity measure?

Georgia Tech

points similar

=

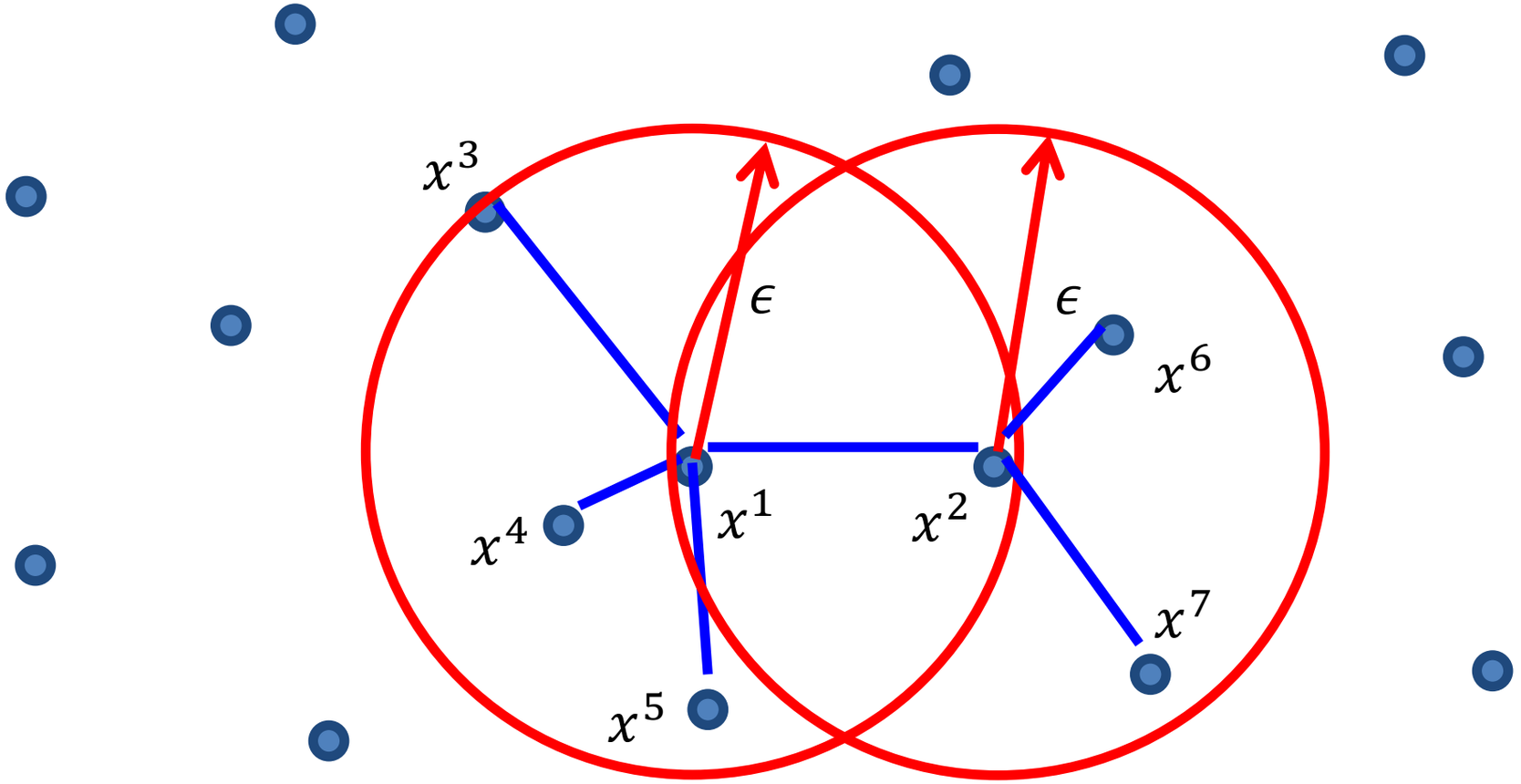
can reach each other by small jumps through data cloud



Neighbor graph

- Given m data points, threshold ϵ, construct matrix $A \in R^{m \times m}$

$$A^{ij} = \begin{cases} 1, \text{if } \left\| x^i - x^j \right\| \leq \epsilon \\ 0, \qquad \text{otherwise} \end{cases}$$



Spectral clustering for vectorial data

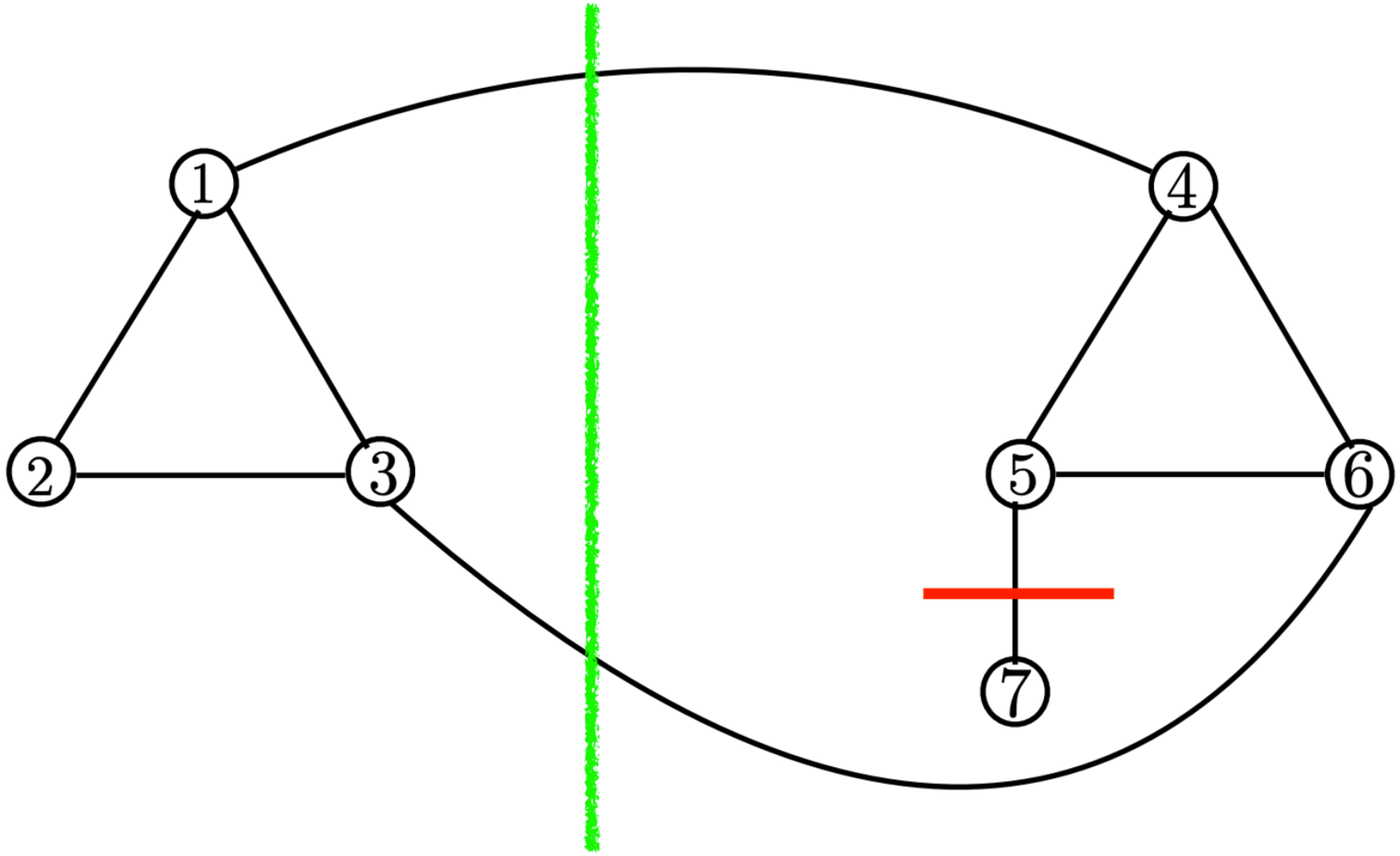
- Given m nodes, $\{x^1, x^2, \dots x^m\} \in R^n$
- Step 1: build an adjacency matrix A using nearest neighbors
- Step 2: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 3: form a special matrix $L = D - A$, the graph Laplacian
- Step 4: compute k eigenvectors, $v^1, v^2, \dots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)
- Step 5: run kmeans algorithm on $Z = \left(v^1, v^2, \dots v^k\right)$ by treating each row as a new data point

Variants of spectral clustering (Ng et al.)

- Given m data points (nodes), $\{x^1, x^2, \dots x^m\} \in R^n$
- Build an adjacency matrix A using kernel functions
- Compute $B = D^{-1/2}AD^{-1/2}$ (or $B = I - D^{-1/2}AD^{-1/2}$), where $D = diag(A1)$
- Compute k eigenvectors, $v^1, v^2, \dots, v^k$, of B corresponding to the k largest (or smallest) eigenvalues
- Run kmeans algorithm on $Z = \left(v^1, v^2, \dots v^k\right)$ by treating each row as a new data point

Unnormalized vs normalized

- Unnormalized Spectral clustering aims to cluster based on minimizing cut
- cut: Number of edges that need to be deleted to have no links between the cluster and other nodes outside
- But is cut the right metric?



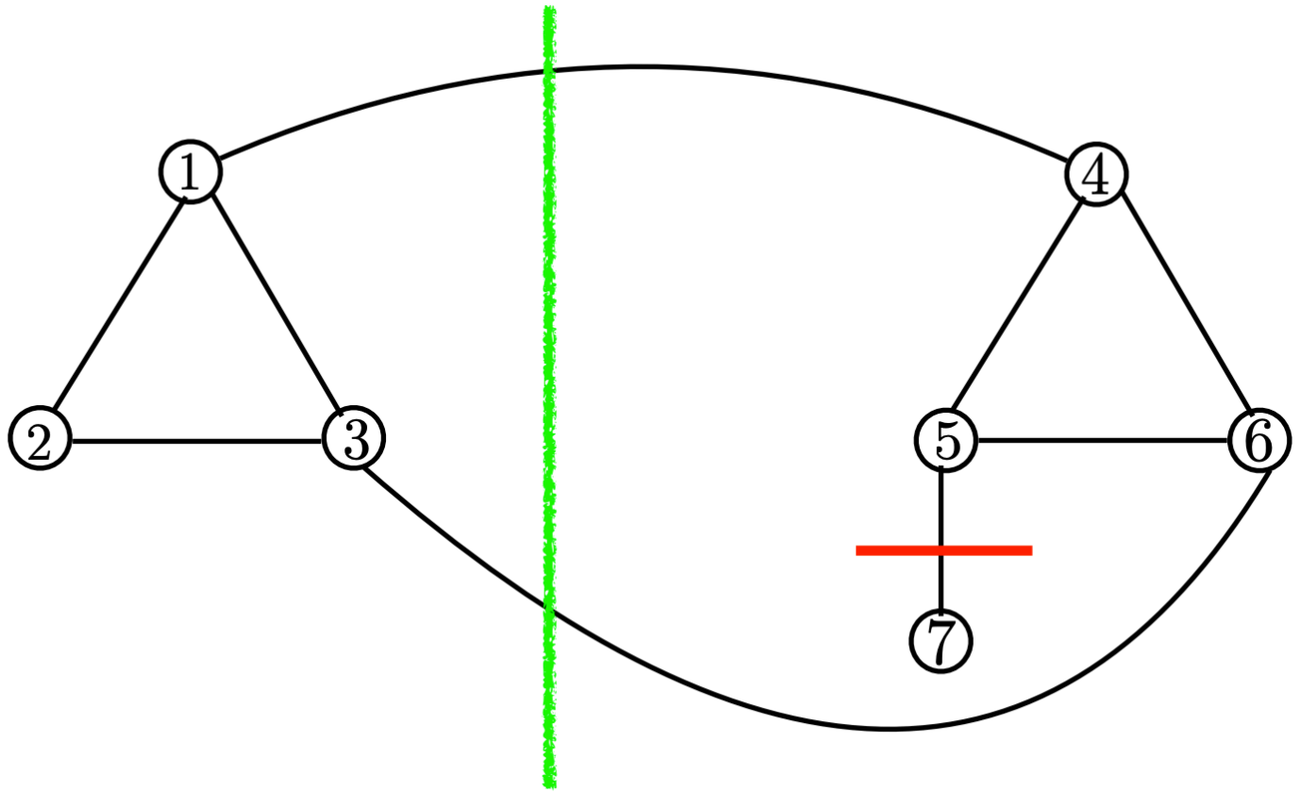
Unnormalized vs normalized

- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

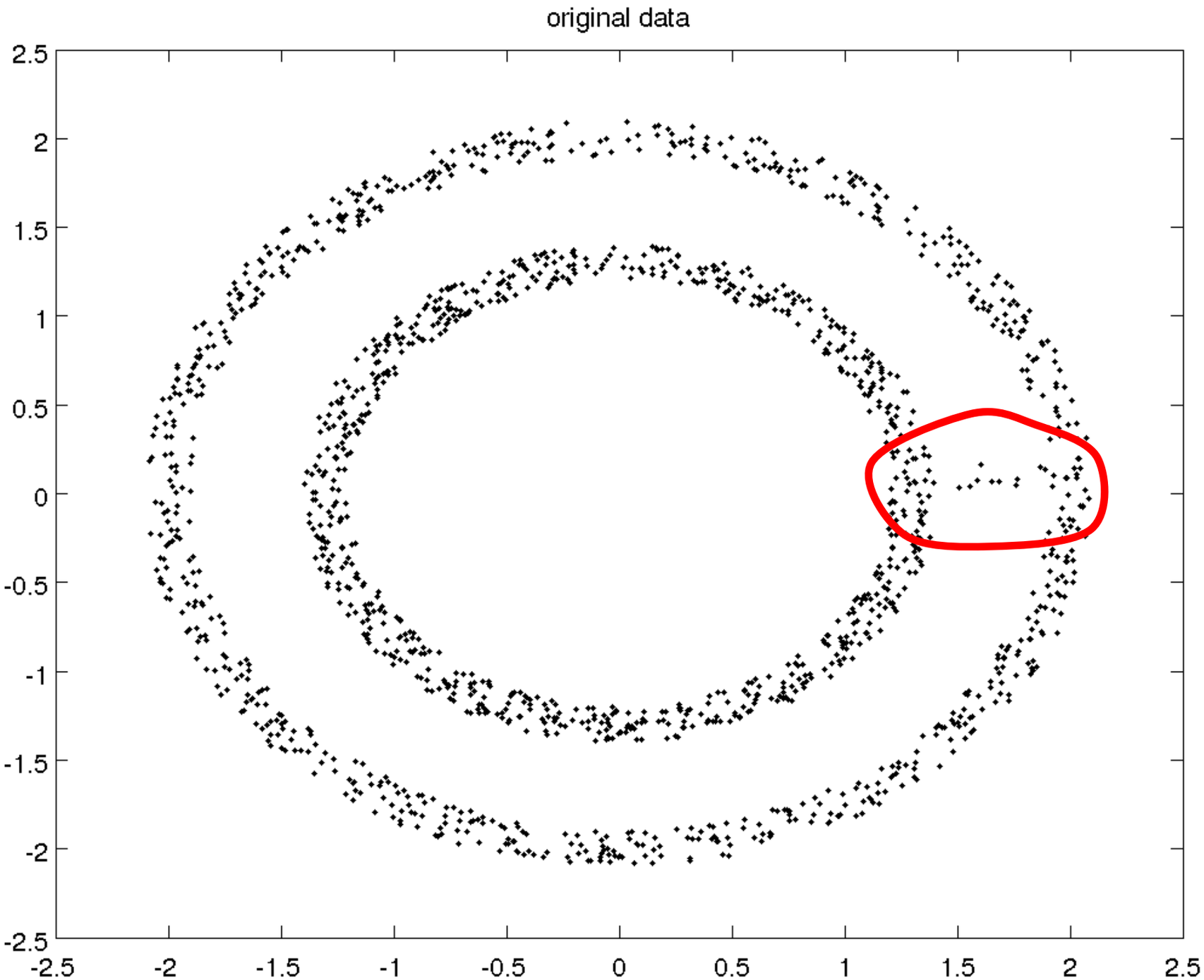
$$\text{NCUT} = \sum_j \frac{\text{CUT}(C_j)}{\text{Edges}(C_j)}$$

- Example K = 2

$$\text{CUT}(C_1, C_2)\left(\frac{1}{\text{Edges}(C_1)} + \frac{1}{\text{Edges}(C_2)}\right)$$



What happens by adding more data points?



What if my graph is large?

- Key challenge:
  - Eigen-decomposition of a large graph Laplacian is expensive
  - How to scale the algorithm up to millions of nodes?

- One solution:
  - Use randomized linear algebra to approximately find top eigenvectors of $B = D^{-1/2}AD^{-1/2} \in R^{m \times m}$ (big)
    - Generate a Gaussian random matrix $\Omega \in R^{d \times m}$ $(d \ll m)$
    - Find a set of orthonormal basis Q for column of

$$Y = B\Omega^{\mathsf{T}}$$

      eg. using Gram-Schmidt orthogonalization
    - Eigendecomposition of $C = \Omega B\Omega^{\mathsf{T}} \in R^{d \times d}$ (tiny)

$$C = U\Lambda U^{\mathsf{T}}$$

    - $Z = QU$
    - References: Halko, Martinsson and Tropp (2009).

Run test_citationgraph.m

- Compare matlab eigs to the randomized eigendecomposition
- Compare matlab implemenation of kmeans with our vectorized implementation
- Our codes run much faster.

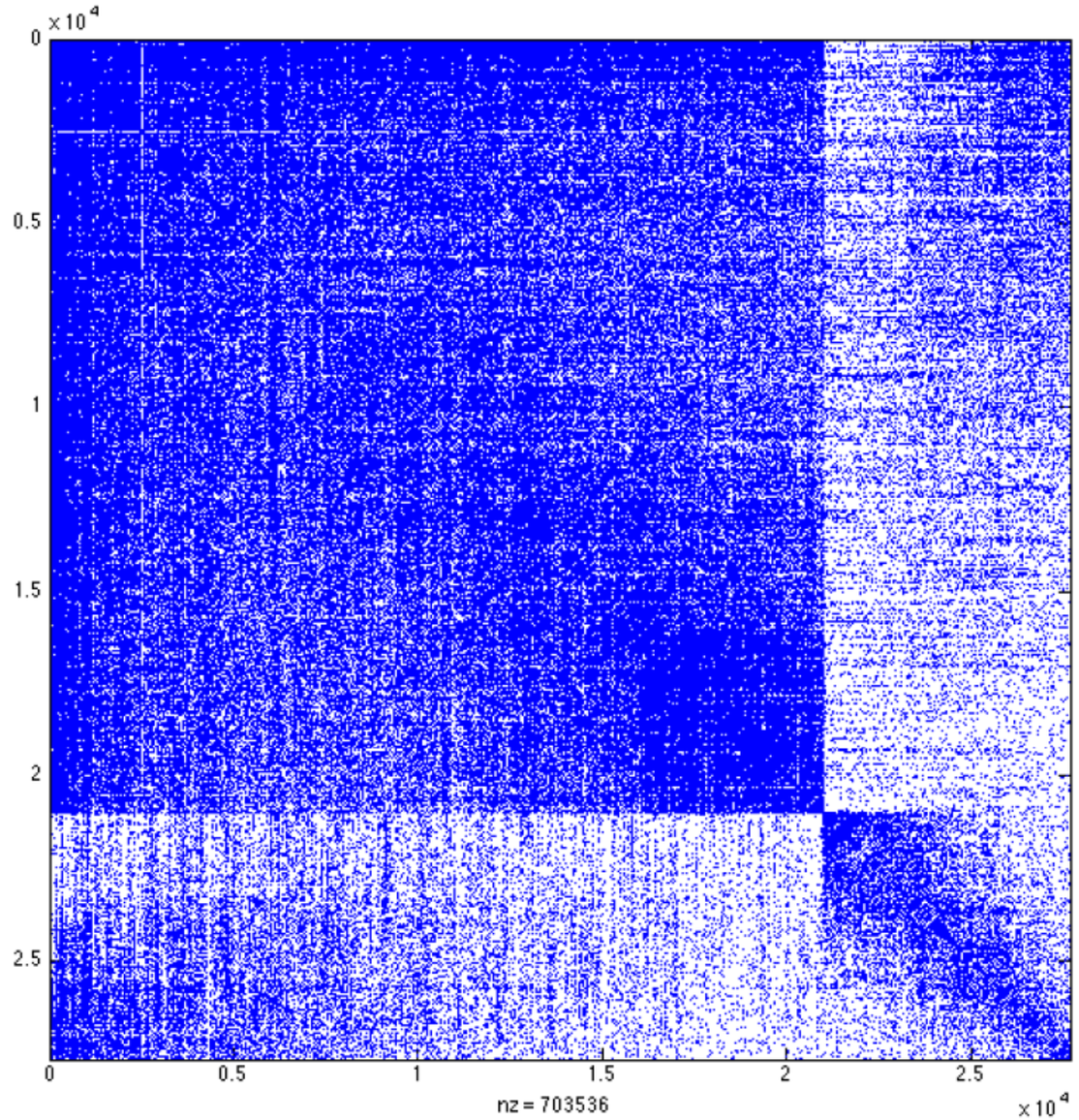
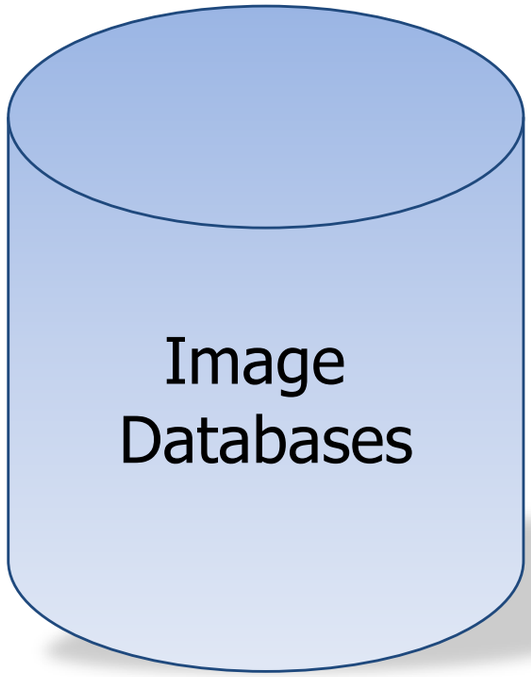
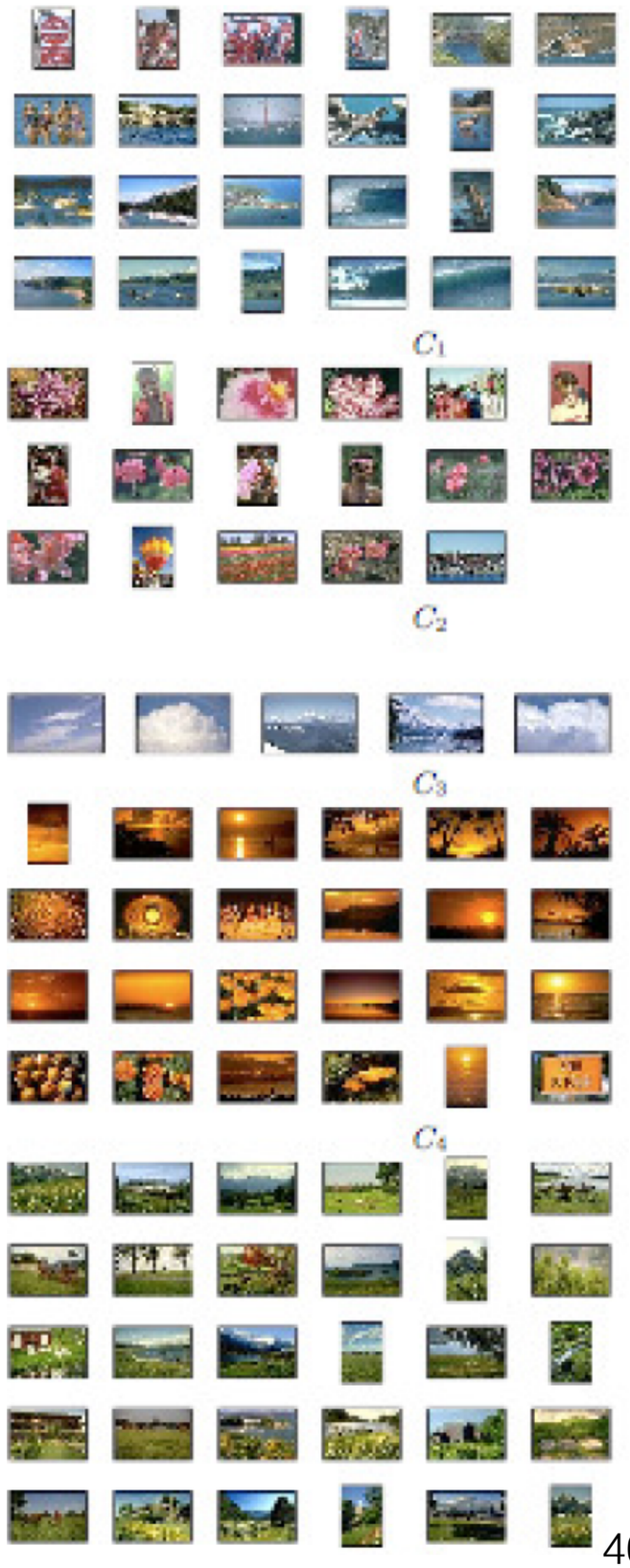
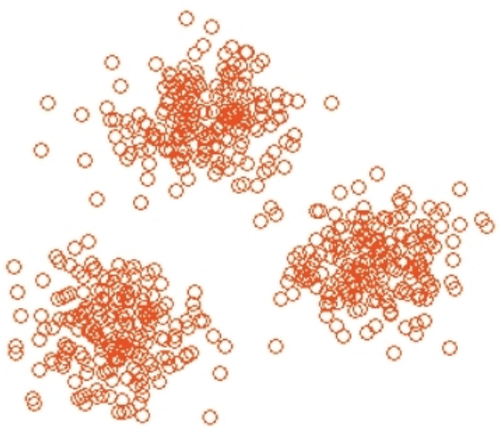


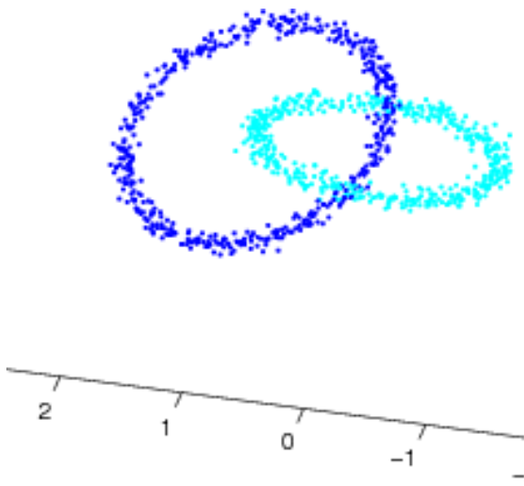
Image databases

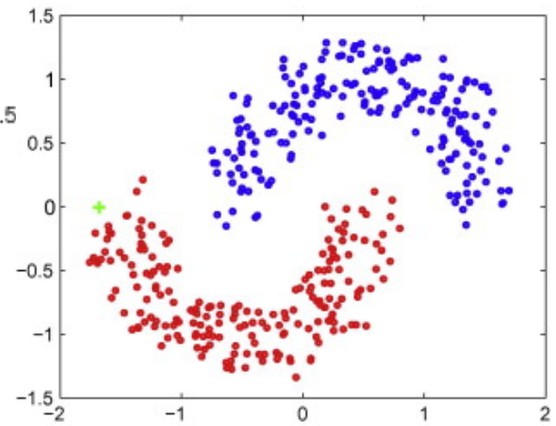




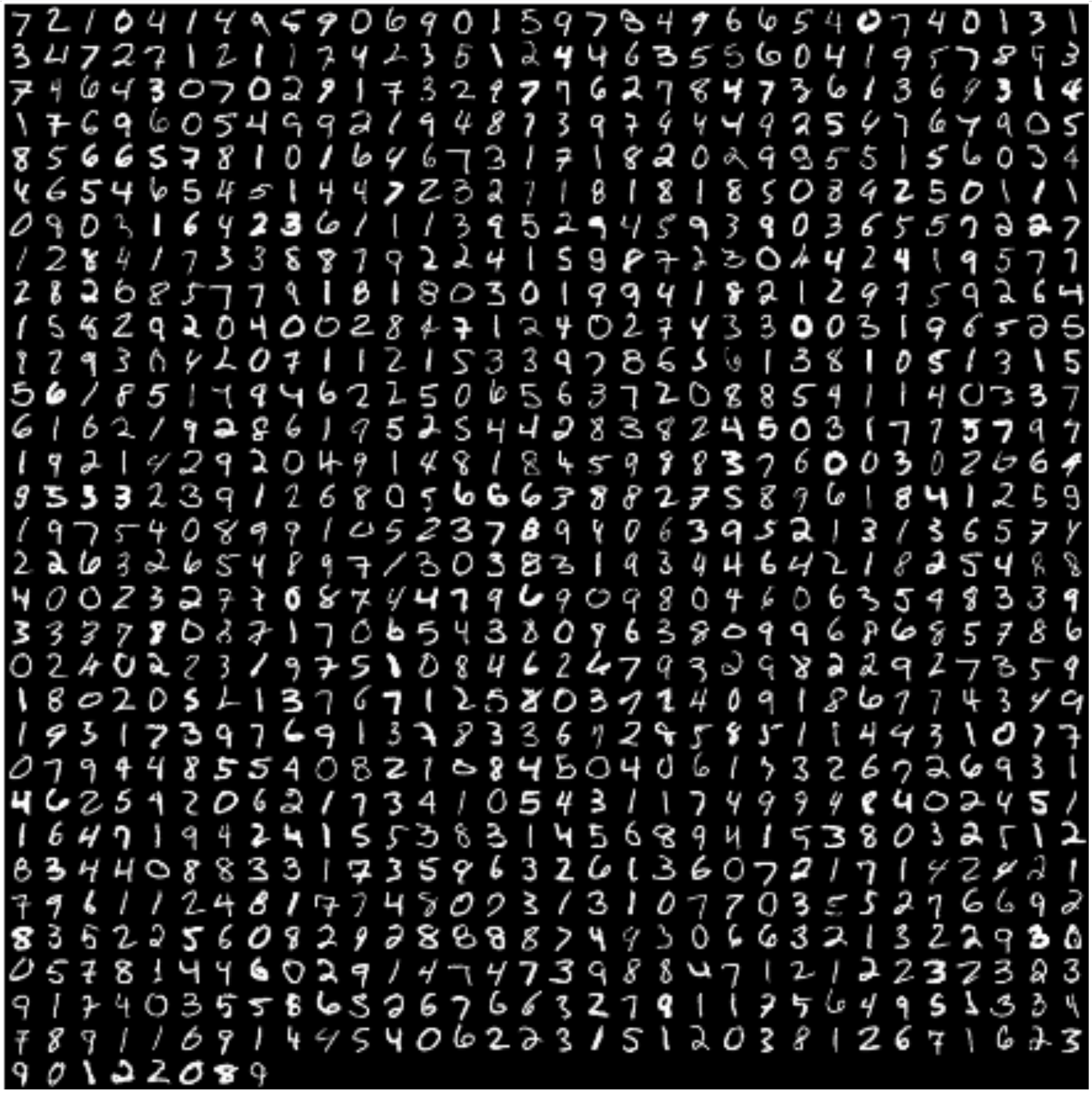
What are the relations between data points?



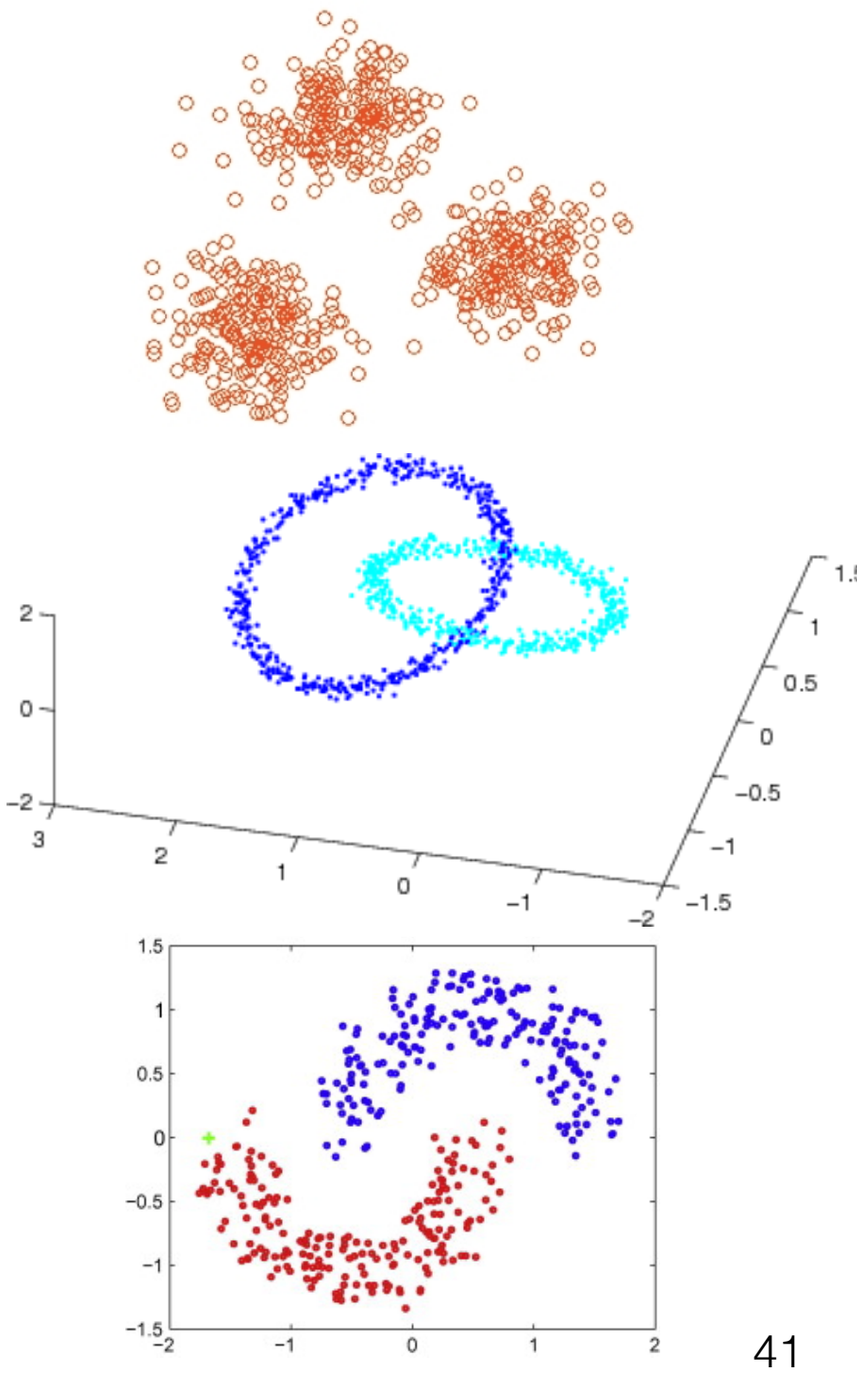




Handwritten digits

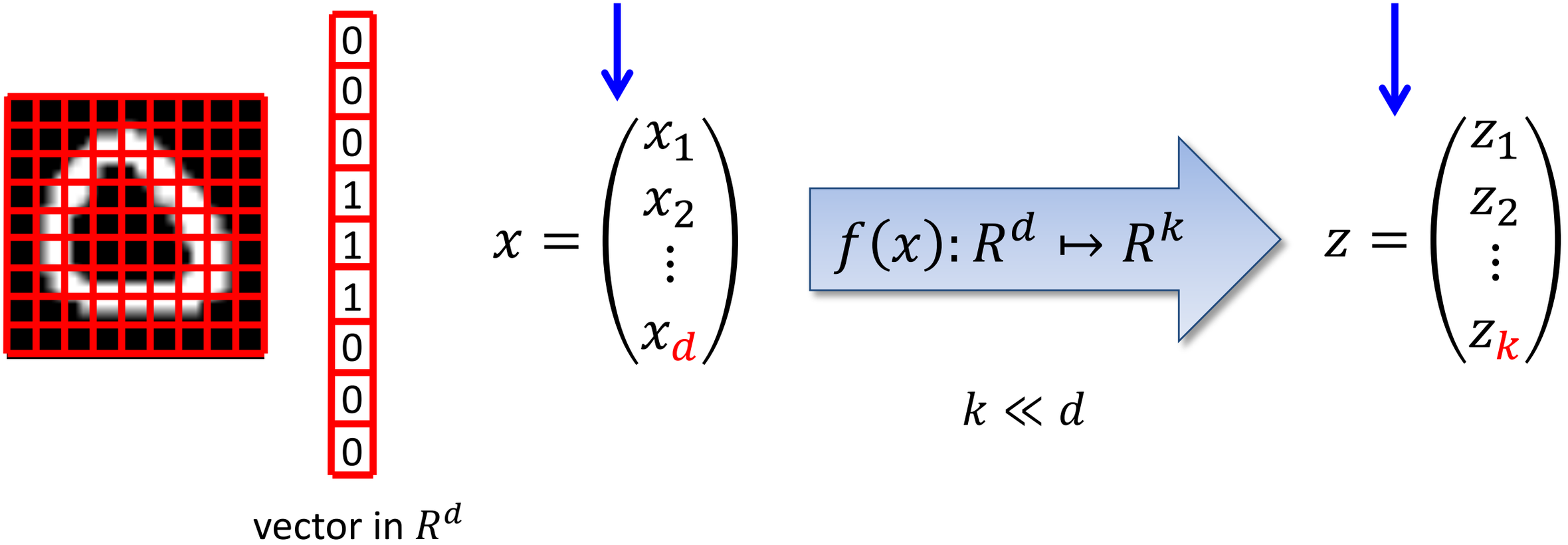


What are the relations between data points?



What is dimensionality reduction?

- The process of reducing the number of random variables under consideration
  - One can combine, transform or select variables
  - One can use linear or nonlinear operations



Why dimensionality reduction and how to think

- The dimension-reduced data can be used for
  - Visualizing, exploring and understanding the data
  - Aggregating weak signals in the data
  - Cleaning the data
  - Speeding up subsequent learning task
  - Building simpler model later

- Key questions of a dimensionality reduction algorithm
  - What is the criterion for carrying out the reduction process?
  - What are the algorithm steps?

Use what criterion for reduction?

- There are many criteria (geometric based, information theory based, etc.)

- One criterion: want to capture variation in data
  - variations are “signals” or information in the data
  - need to normalize each variables first

- In the process, also discover variables or dimensions highly correlated
  - represent highly related phenomena
  - combine them to form a stronger signal
  - lead to simpler presentation

How to formulate the problem

- Given m data points, $\{x^1, x^2, \dots x^m\} \in R^d$, with their mean $\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$

- Find a direction $w \in R^d$ where $\|w\| = 1$

- Such that the variance (or variation) of the data along direction w is maximized

$$\max_{w:\|w\|=1} \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(w^\top x^i - w^\top \mu\right)^2}_{\text{variance}}$$

Is it an easy optimization problem?

- Manipulate the objective with linear algebra

$$\frac{1}{m}\sum_{i=1}^{m}\left(w^{\top}x^{i}-w^{\top}\mu\right)^{2}$$

$$=\frac{1}{m}\sum_{i=1}^{m}\left(w^{\top}(x^{i}-\mu)\right)^{2}$$

$$=\frac{1}{m}\sum_{i=1}^{m}w^{\top}(x^{i}-\mu)(x^{i}-\mu)^{\top}w$$

$$=w^{\top}\underbrace{\left(\frac{1}{m}\sum_{i=1}^{m}(x^{i}-\mu)(x^{i}-\mu)^{\top}\right)}_{\text{covariance matrix } C}w$$

Landscape of the optimization problem

- Suppose the data has two dimension ($d = 2$)

- C is a diagonal matrix

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

- The optimization problem becomes

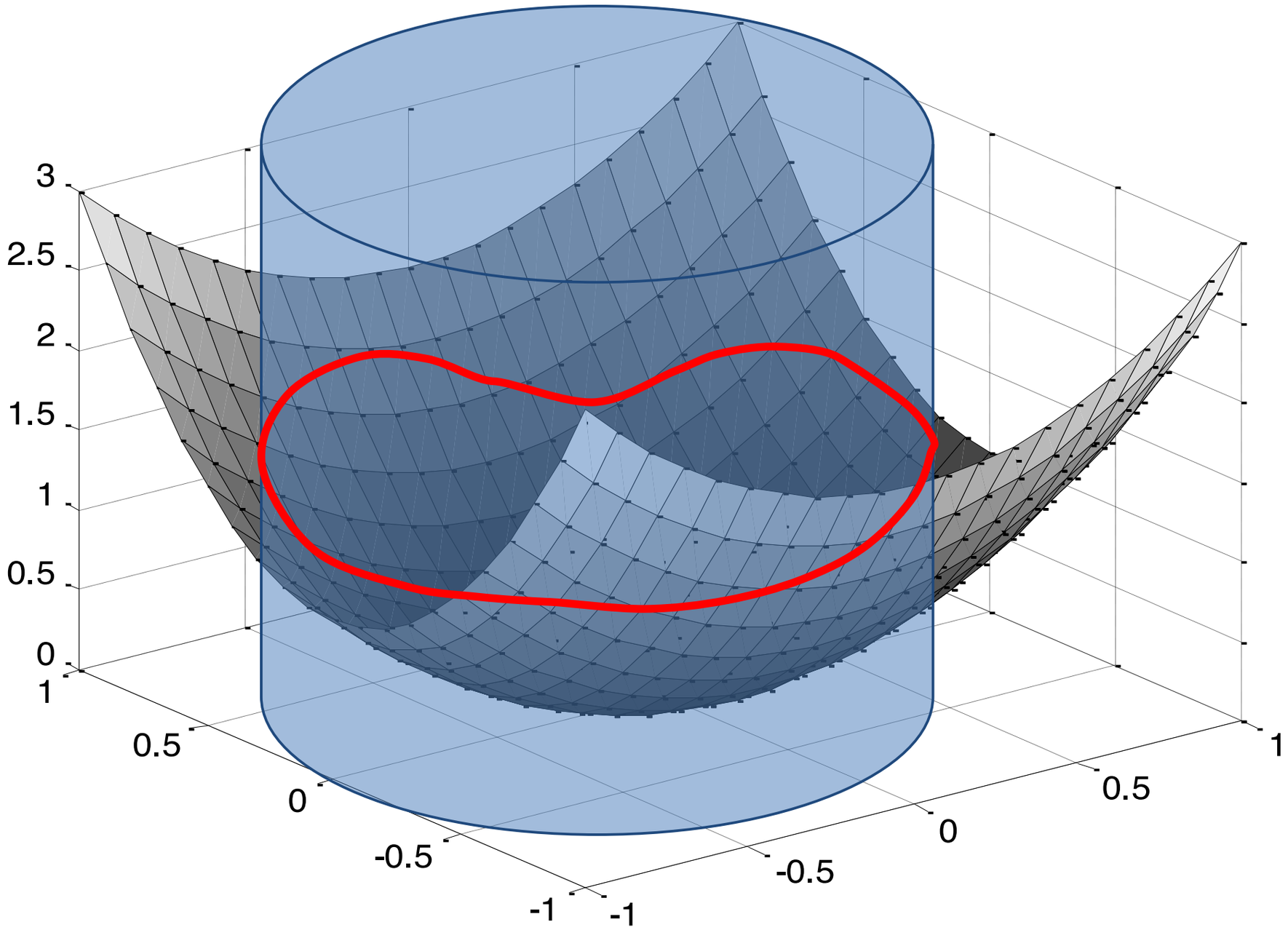
$$\max_{w:\|w\|=1} w^\top C w$$

$$= \max_{w:\|w\|=1} (w_1, w_2) \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

$$= \max_{w:\|w\|=1} w_1^2 + 2w_2^2$$

Landscape of the optimization problem

- $f(w_1, w_2) = w_1^2 + 2w_2^2$



Eigen-value problem

- Eigen-value problem
  - Given a symmetric matrix $C \in R^{d \times d}$
  - Find a vector $w \in R^d$ and $\|w\| = 1$
    - Such that

$$Cw = \lambda w$$

- There will be multiple solution of $w^1, w^2, \ldots$ with different $\lambda_1, \lambda_2, \ldots$
  - They are ortho-normal: ${w^i}^{\mathsf{T}} w^i = 1, {w^i}^{\mathsf{T}} w^j = 0$

Equivalent to eigen-value problem

- Claim:

$$\max_{w:\|w\|=1} w^\top Cw \Leftrightarrow Cw = \lambda w$$

- Form lagrangian function of the optimization problem

$$L(w,\lambda) = w^\top Cw + \lambda(1 - \|w\|^2)$$

Necessary condition

- If w is a maximum of the original optimization problem, then there exists a λ, where (w,λ) is a stationary point of $L(w,\lambda)$

- This implies that

$$\frac{\partial L}{\partial w} = 0 = 2Cw - 2\lambda w$$

Variance of in the principal direction

- Principal direction w satisfies

$$Cw = \lambda w$$

- Variance in principal direction is

Variance of in the principal direction

- Principal direction w satisfies

$$Cw = \lambda w$$

- Variance in principal direction is

$$w^\top C w$$

$$= \lambda w^\top w$$

$$= \lambda$$

eigen-value

Multiple principal directions

- Directions $w^1, w^2, \ldots$ which has
  - the largest variances
  - but are orthogonal to each other

- Take the eigenvectors $w^1, w^2, \ldots$ of C corresponding to
  - the largest eigenvalue λ_1,
  - the second largest eigenvalue λ_2
  - …

Principal component analysis

- Given m data points, $\{x^1, x^2, \ldots x^m\} \in R^d$, with mean

- Step 1: Estimate the mean and covariance matrix from data

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i \quad and \quad C = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)(x^i - \mu)^{\mathsf{T}}$$

Principal directions

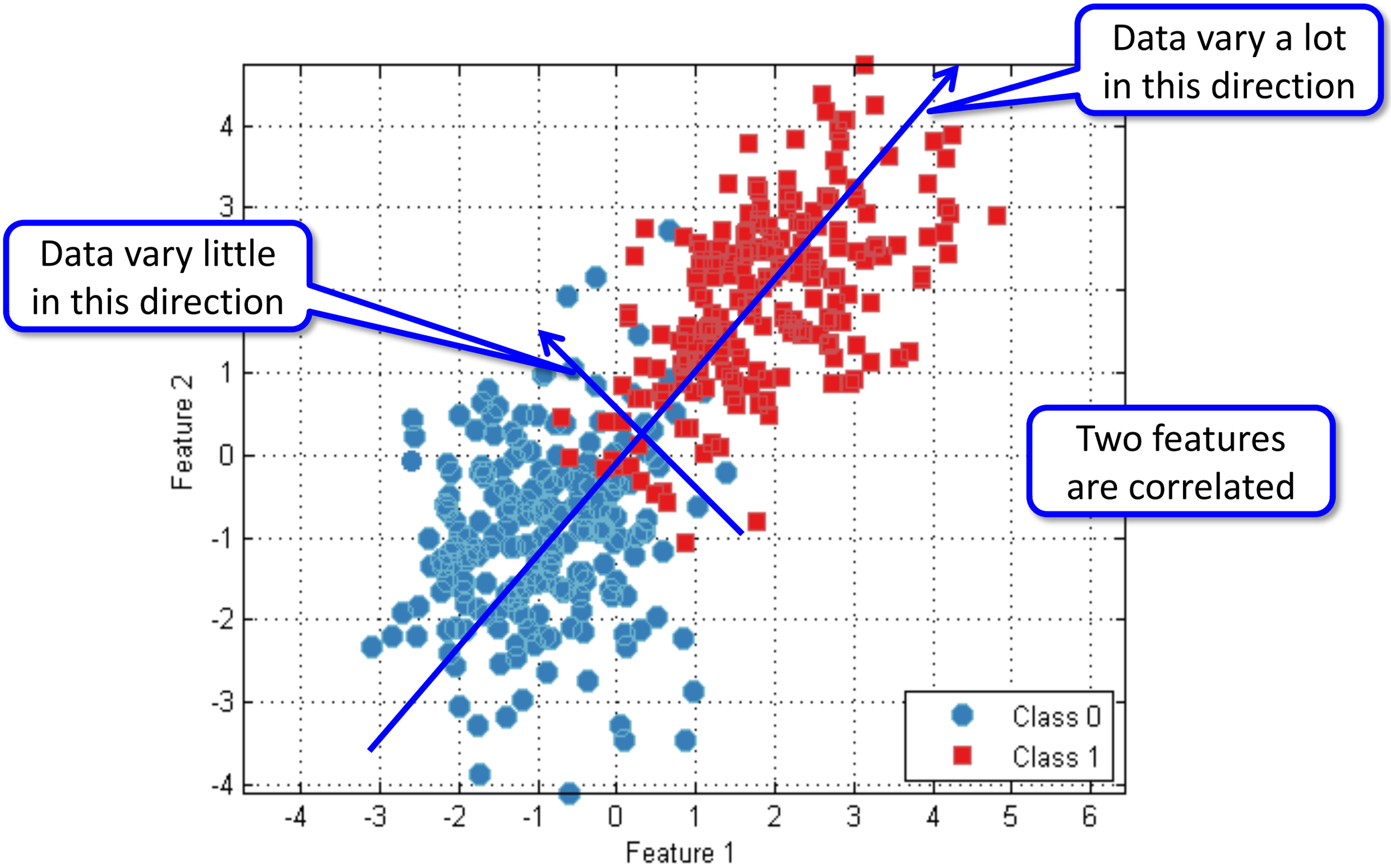
- Step 2: Take the eigenvectors $w^1, w^2, \ldots$ of C corresponding to the largest eigenvalue λ_1, the second largest eigenvalue λ_2 …

- Step 3: Compute reduced representation

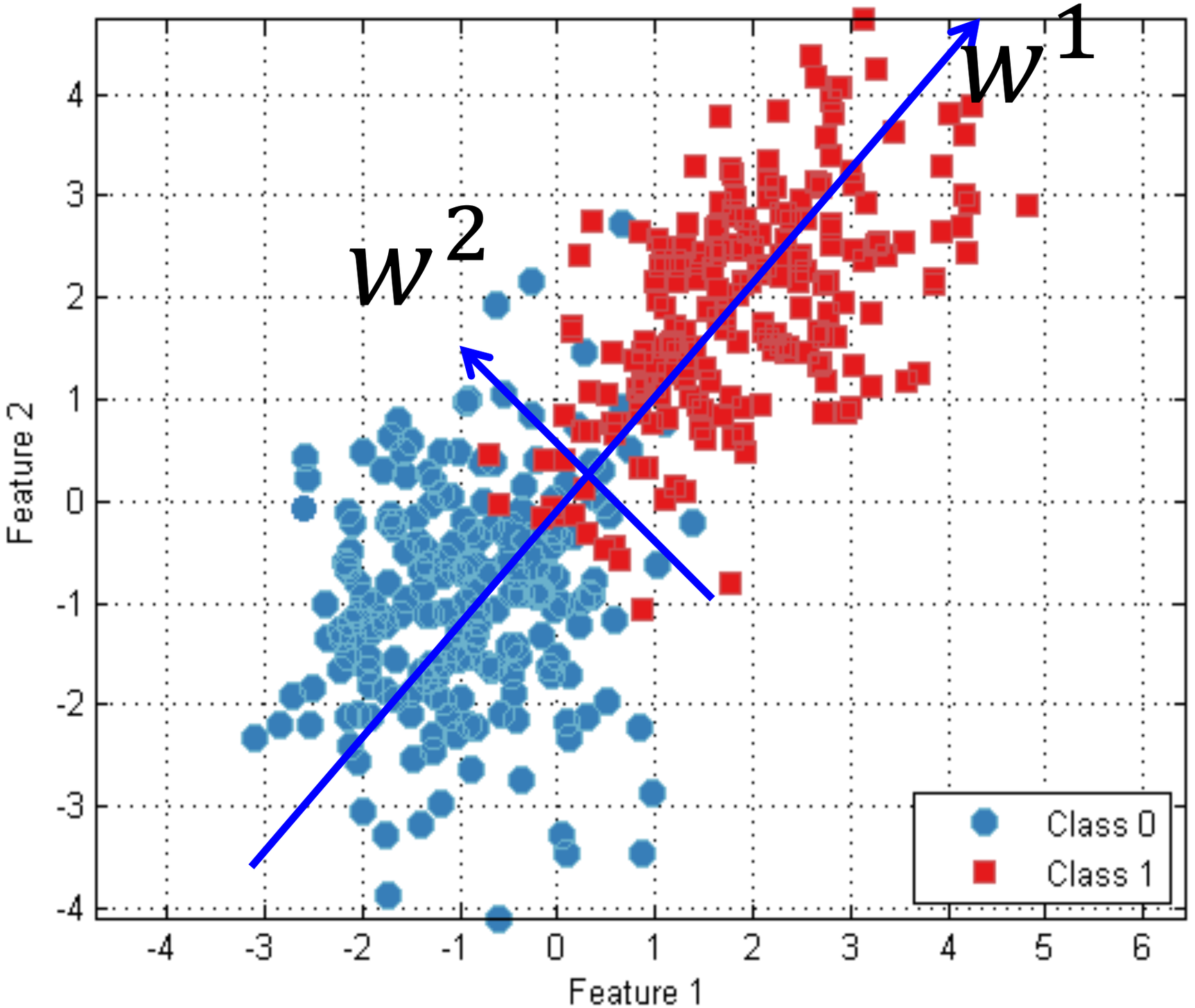
$$z^i = \begin{pmatrix} w^{1\mathsf{T}}(x^i - \mu)/\sqrt{\lambda_1} \\ w^{2\mathsf{T}}(x^i - \mu)/\sqrt{\lambda_2} \\ \vdots \end{pmatrix}$$

Normalize by standard deviation

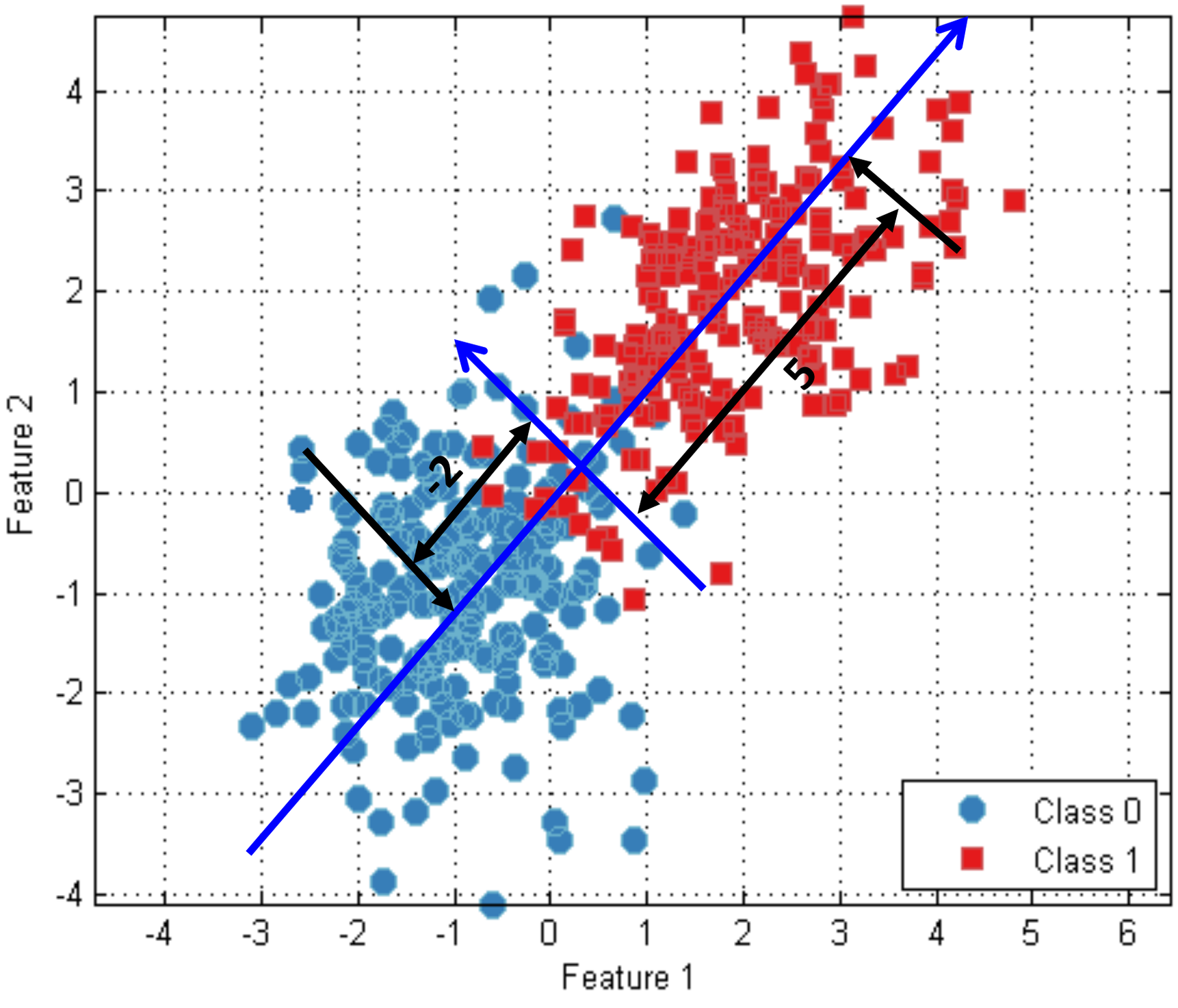
An example


Data vary a lot
in this direction
Data vary little
in this direction
Two features
are correlated
Feature 1
Feature 2
Class 0
Class 1

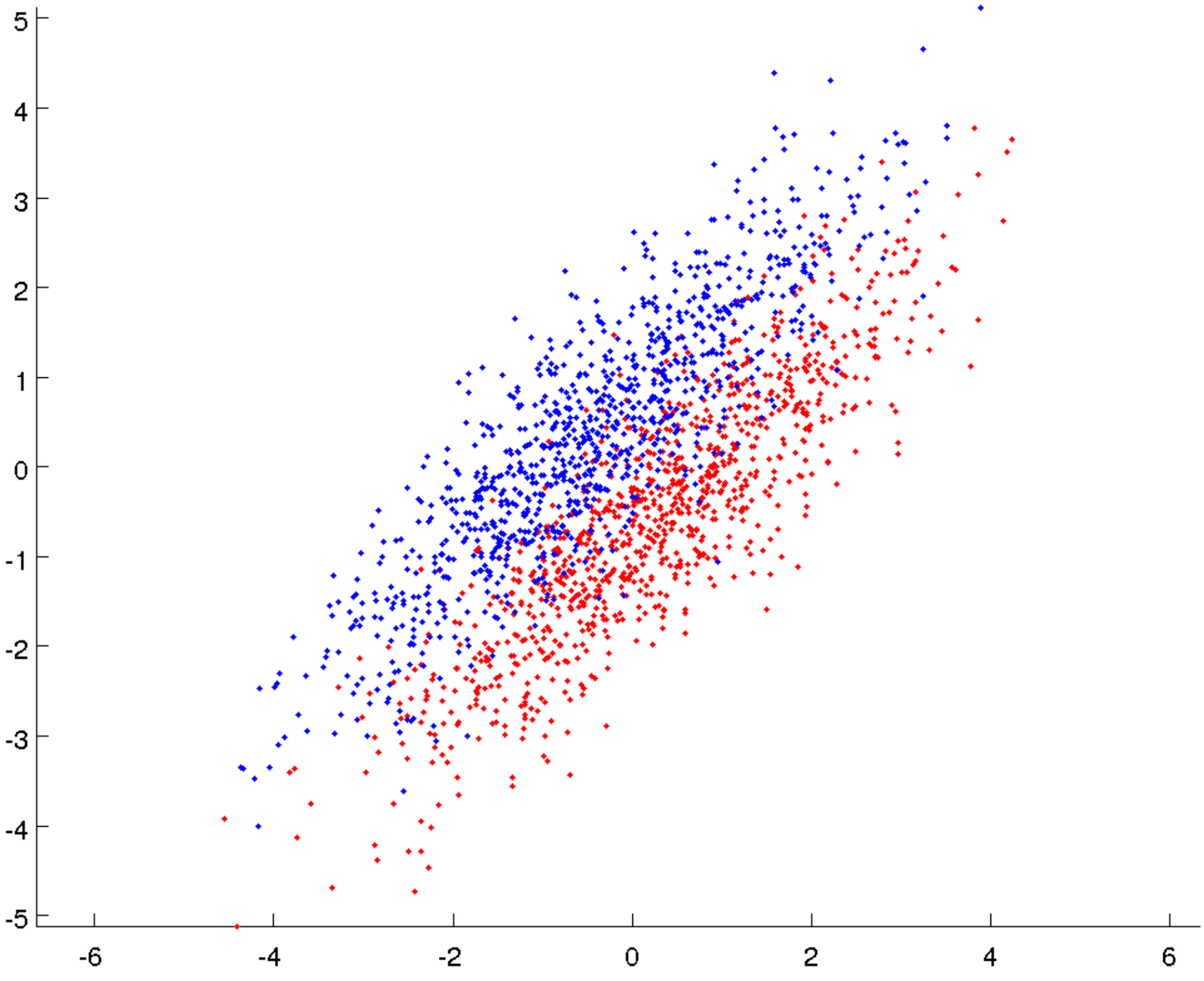
Principal direction of the data



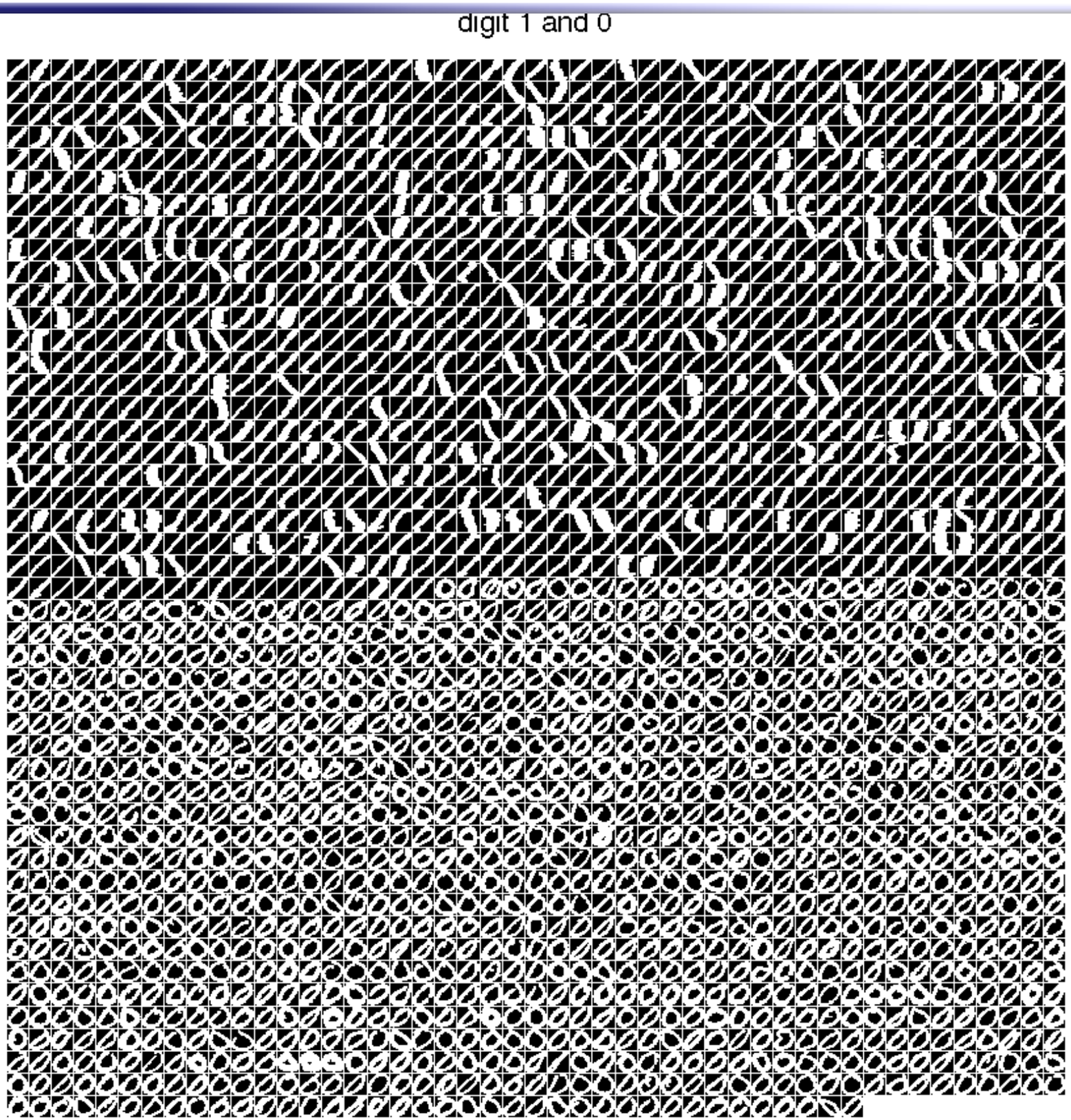
Reduce to 1 dimension



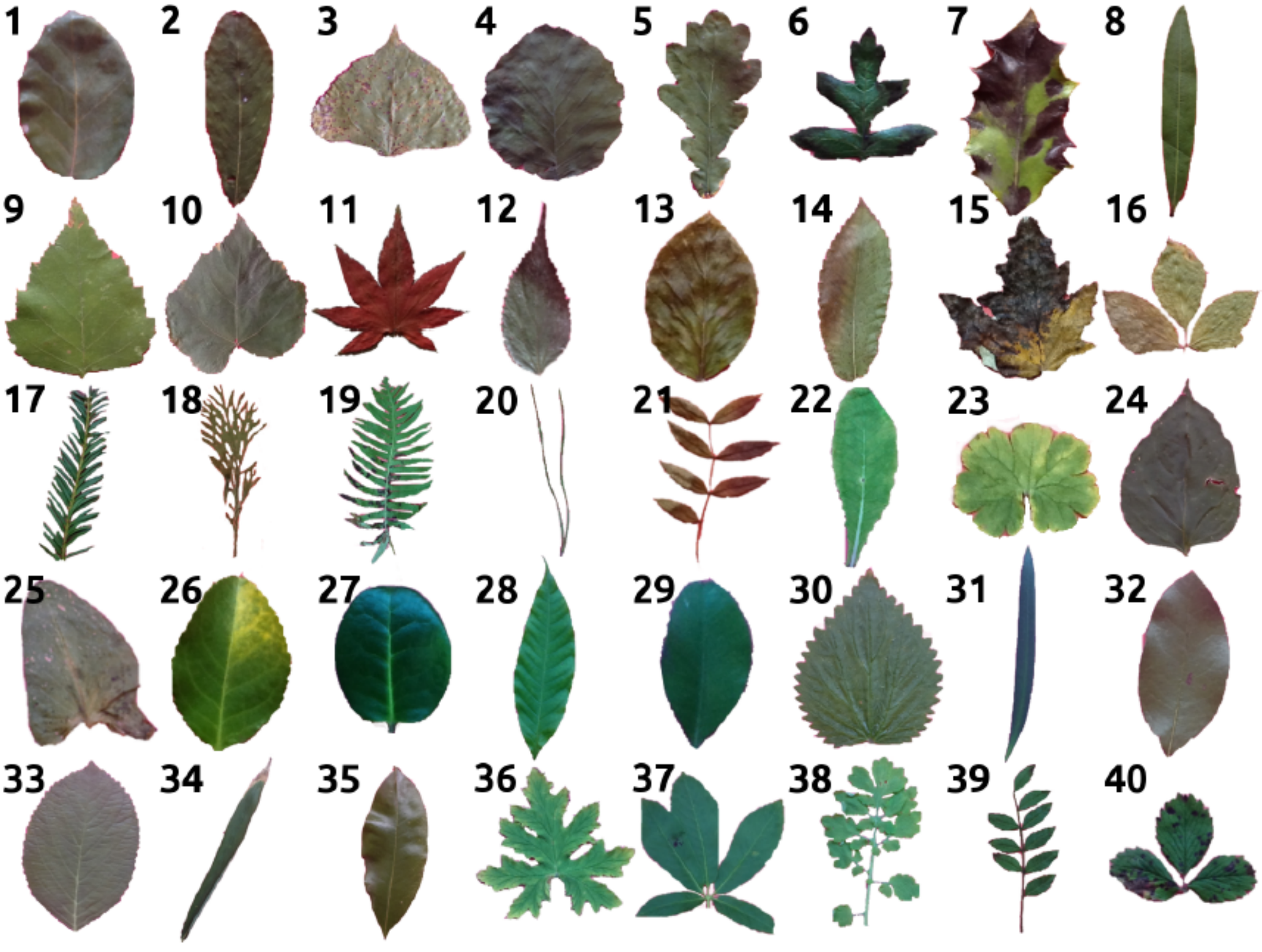
Are principal components good for classification?



Run demo PCA_digits.m



Run demo PCA_leaf.m



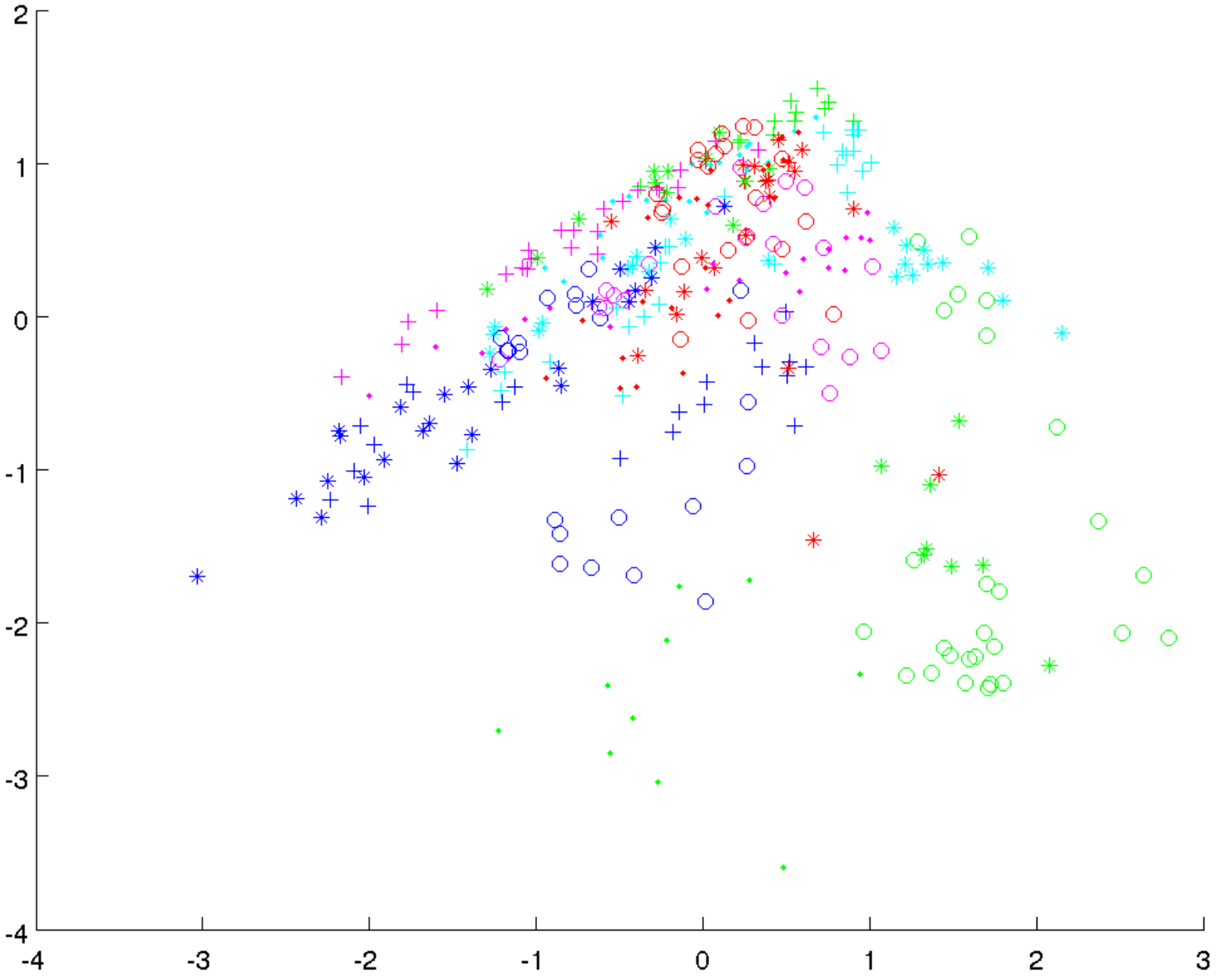
Input features (representation)

Shape feature	Description
Eccentricity	Eccentricity of the ellipse with identical second moments to I. This value ranges from 0 to 1.
Aspect Ratio	Consider any $X, Y \in \partial I$. Choose X and Y such that $d(X,Y) = D(I)$. Find $Z, W \in \partial I$ maximizing $D^{\perp} = d(Z,W)$ on the set of all pairs of ∂I that define a segment orthogonal to $[XY]$. The aspect ratio is defined as the quotient $D(I)/D^{\perp}$. Values close to 0 indicate an elongated shape.
Elongation	Compute the maximum escape distance $d_{\max} = \max_{X \in I} d(X, \partial I)$. Elongation is obtained as $1 - 2d_{\max}/D(I)$ and ranges from 0 to 1. The minimum is achieved for a circular region. Note that the ratio $2d_{\max}/D(I)$ is the quotient between the diameter of the largest inscribed circle and the diameter of the smallest circumscribed circle.
Solidity	The ratio $A(I)/A(H(I))$ is computed, which can be understood as a certain measure of convexity. It measures how well I fits a convex shape.
Stochastic Convexity	This variable extends the usual notion of convexity in topological sense, using sampling to perform the calculation. The aim is to estimate the probability of a random segment $[XY]$, $X, Y \in I$, to be fully contained in I.
Isoperimetric Factor	The ratio $4\pi A(I)/L(\partial I)^2$ is calculated. The maximum value of 1 is reached for a circular region. Curvy intertwined contours yield low values.
Maximal Indentation Depth	Let $C_{H(I)}$ and $L(H(I))$ denote the centroid and arclength of $H(I)$. The distances $d(X, C_{H(I)})$ and $d(Y, C_{H(I)})$ are computed $\forall X \in H(I)$ and $\forall Y \in \partial I$. The indentation function can then be defined as $[d(X, C_{H(I)}) - d(Y, C_{H(I)})]/L(H(I))$, which is sampled at one degree intervals. The maximal indentation depth $\mathfrak{D}$ is the maximum of this function.
Lobedness	The Fourier Transform of the indentation function above is computed after mean removal. The resulting spectrum is normalized by the total energy. Calculate lobedness as $F \times \mathfrak{D}^2$, where F stands for the smallest frequency at which the cumulated energy exceeds 80%. This feature characterizes how lobed a leaf is.

Texture feature	Description
Average Intensity	Average intensity is defined as the mean of the intensity i
Average Contrast	Average contrast is the the standard deviation of the inte age, $\sigma = \sqrt{\mu_2(z)}$.
Smoothness	Smoothness is defined as $R = 1 - 1/(1+\sigma^2)$ and meas relative smoothness of the intensities in a given region. For of constant intensity, R takes the value 0 and R approac regions exhibit larger disparities in intensity values. σ^2 is normalized by $(L-1)^2$ to ensure that $R \in [0,1]$.
Third moment	μ_3 is a measure of the intensity histogram's skewness. This is generally normalized by $(L-1)^2$ like smoothness.
Uniformity	Defined as $U = \sum_{i=0}^{L-1} p^2(z_i)$, uniformity's maximum reached when all intensity levels are equal.
Entropy	A measure of intensity randomness.

8 shape features and
6 texture features

Reduce representation



Principal direction:

W =

0.0938	0.1924
0.1902	0.0253
0.2266	-0.1800
-0.1850	0.4084
-0.1600	0.3825
-0.2063	0.3488
0.1940	-0.4037
0.2150	-0.3566
-0.3723	-0.2001
-0.3657	-0.1974
-0.3602	-0.2037
-0.3175	-0.1886
-0.3056	-0.1243
-0.3482	-0.1829

Shape features

Texture features

Singular Value Decomposition

- Singular value decomposition, known as SVD, is a factorization of a real matrix with applications in calculating pseudo-inverse, rank, solving linear equations, and many others.

- For a matrix $M \in \mathbb{R}^{m \times n}$ assume $n \leq m$
  - $M = U\Sigma V^{\top}$ where U $\in \mathbb{R}^{m \times m}$, V$^{\top}$ $\in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{m \times n}$

- The m columns of U, and the n columns of V are called the left and right singular vectors of M. The diagonal elements of Σ, Σ_{ii} are known as the singular values of M.

- Let v be the ith column of V, and u be the ith column of U, and σ be the ith diagonal element of Σ

$$Mv = \sigma u \quad \text{and} \quad M^{\top}u = \sigma v$$

Singular Value Decomposition - II

- $M = [u_1 \quad u_2 \ \ldots \quad u_m] \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{m1} & \cdots & \Sigma_{mn} \end{bmatrix} [v_1 \quad v_2 \ldots \quad v_n]^{\mathsf{T}}$

principal directions

Scaling factor

Projection in principal directions

- Singular value decomposition is related to eigenvalue decomposition
  - Suppose $M = [x_1 - \mu \quad x_2 - \mu \ \ldots \quad x_m - \mu] \in \mathbb{R}^{m \times n}$
  - Then covariance matrix is $C = \frac{1}{m} M M^{\mathsf{T}}$
  - Starting from singular vector pair
    - $M^{\mathsf{T}}\mathrm{u} = \sigma v$
    - $\Rightarrow M M^{\mathsf{T}}\mathrm{u} = \sigma M v$
    - $\Rightarrow M M^{\mathsf{T}}\mathrm{u} = \sigma^2 u$
    - $\Rightarrow Cu = \lambda u$