# CSE 6740: Computational Data Analysis

## Spring 2026

# Dimensionality Reduction: PCA

Anqi Wu

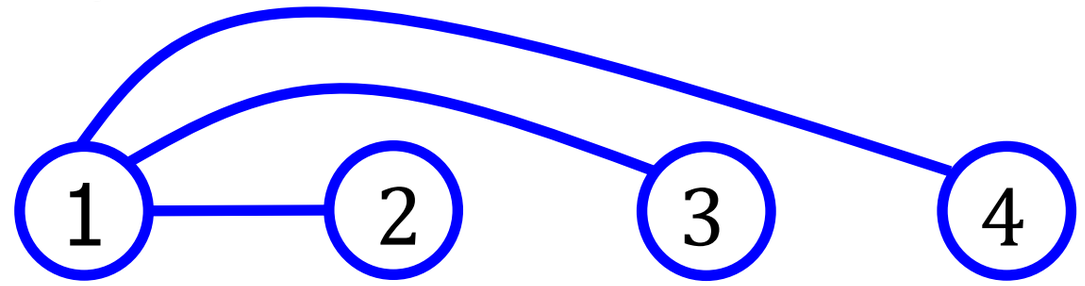
01/27

# 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

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

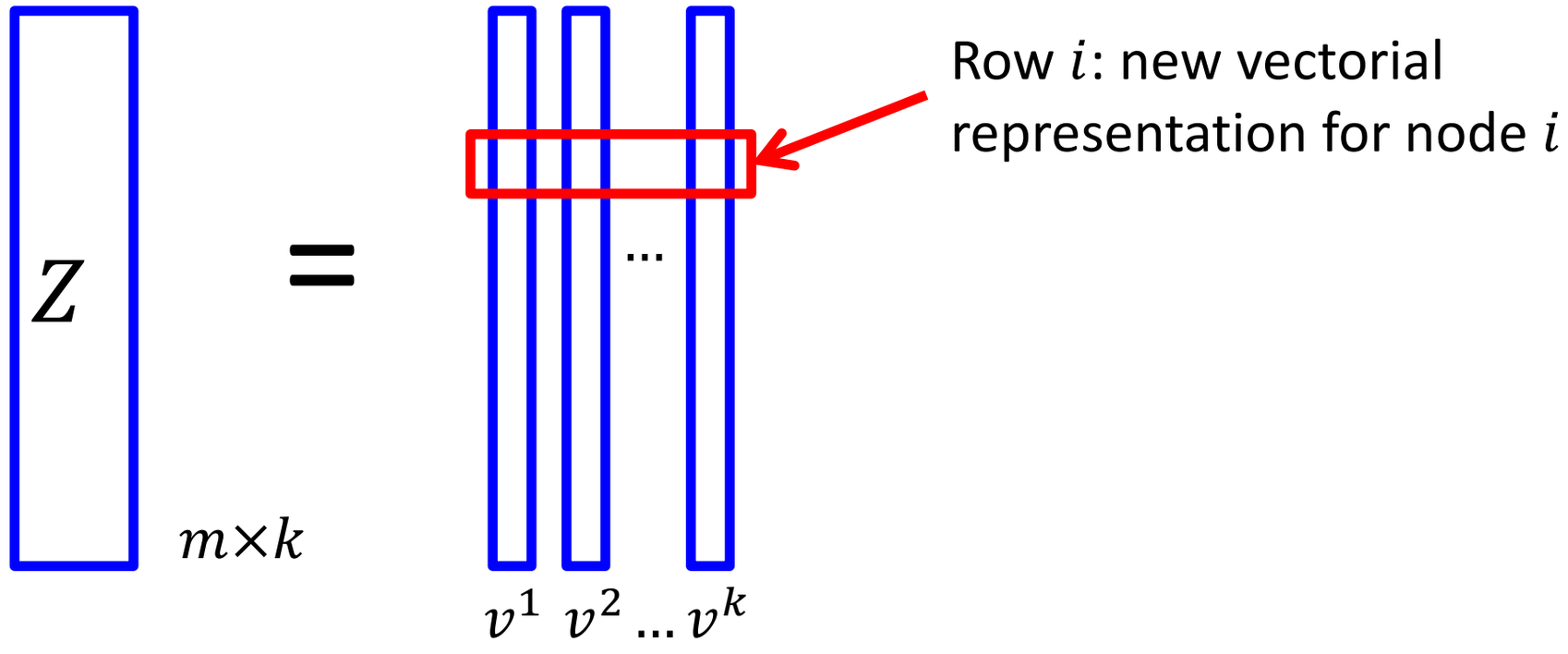
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

# Spectral clustering algorithm (cont.)

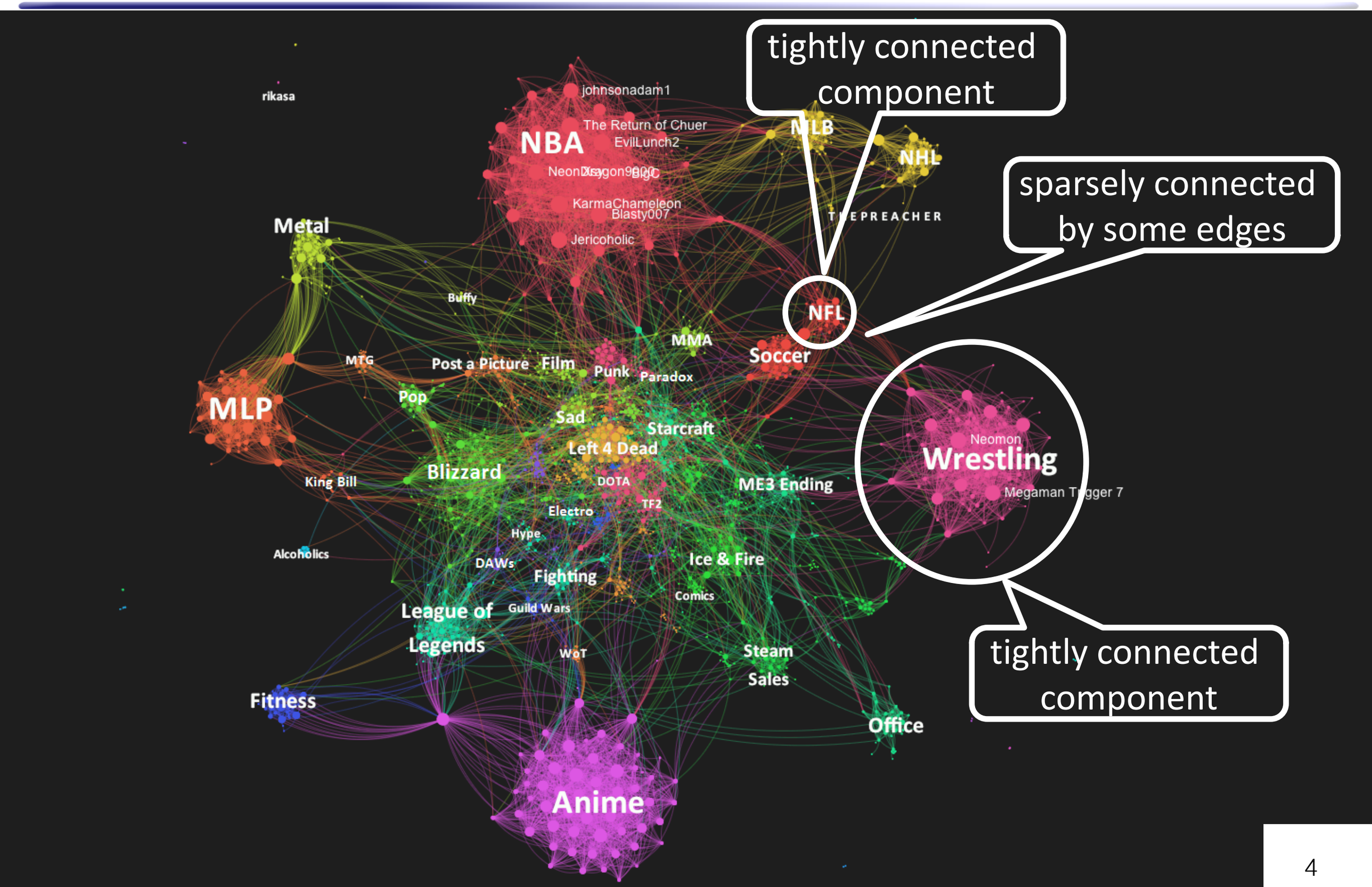
- Step 3: compute $k$ eigenvectors, $v^1, v^2, \ldots, v^k$, of $L$ corresponding to the $k$ smallest eigenvalues ($k \ll m$)

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

- 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



# Real world social networks



# Summary of spectral clustering

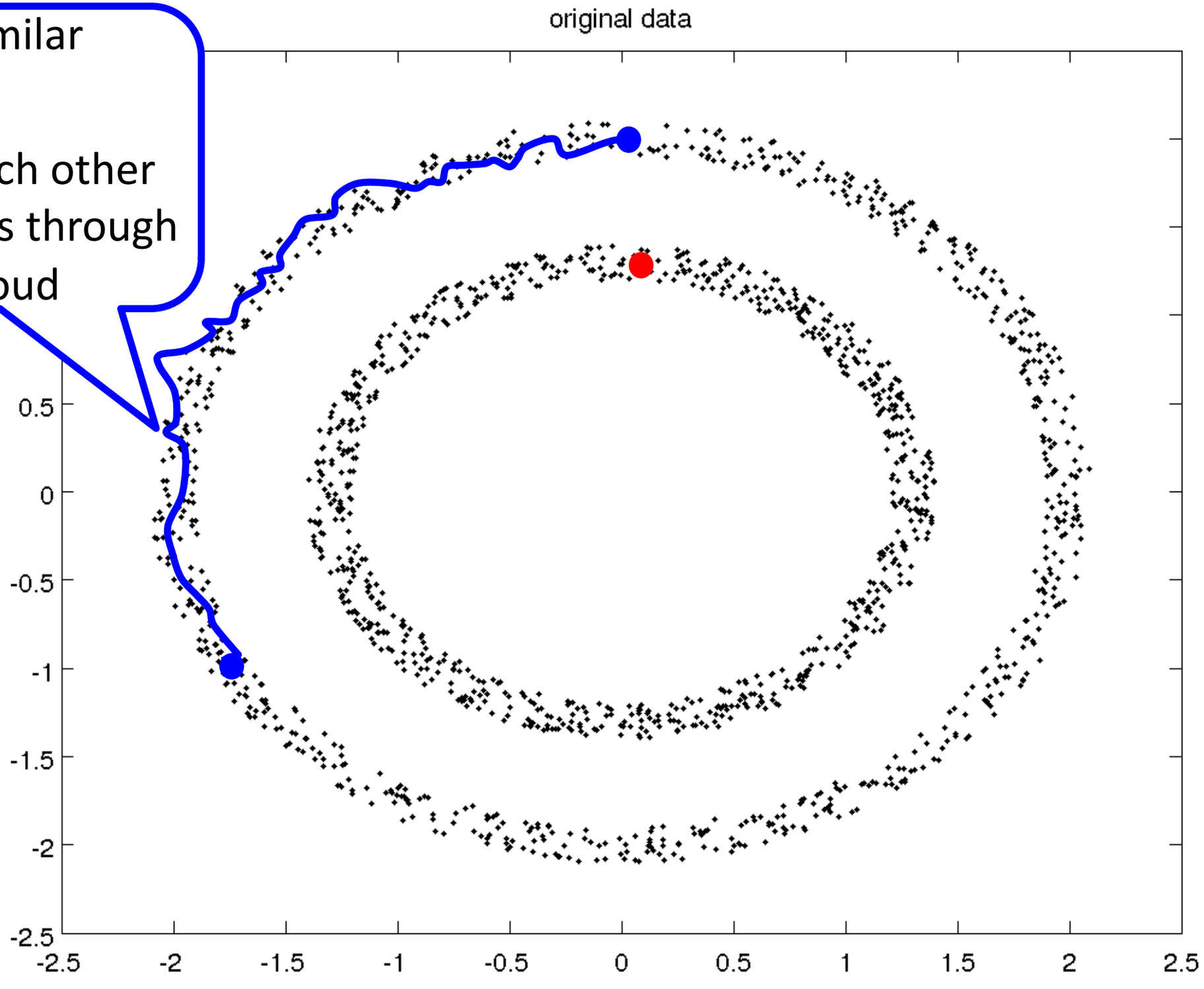
- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 2: form a special matrix $L = D - A$, the graph Laplacian
- Step 3: compute $k$ eigenvectors, $v^1, v^2, \ldots, v^k$, of $L$ corresponding to the $k$ smallest eigenvalues ($k \ll m$)
- Step 4: run kmeans algorithm on $Z = \left(v^1,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

# What's a reasonable similarity measure?

points similar

=

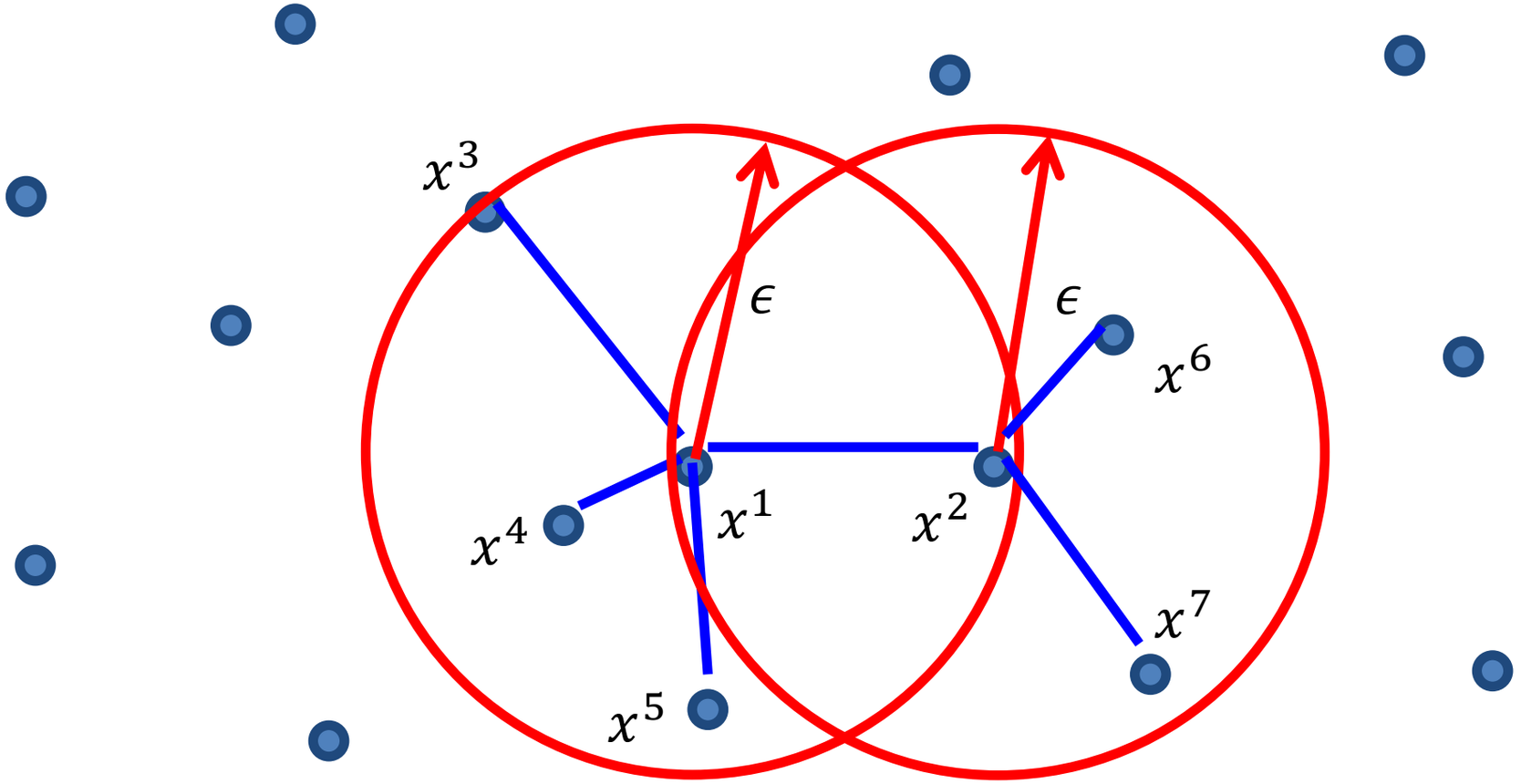
can reach each other by small jumps through data cloud



# Neighbor graph

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

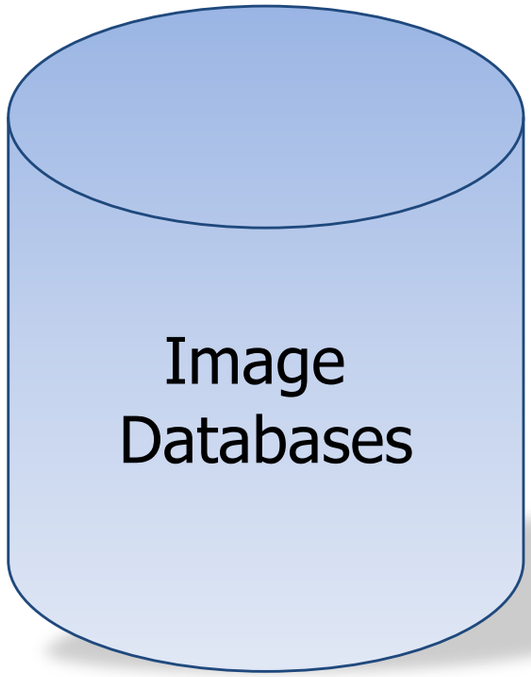
$$A^{ij} = \begin{cases} 1, \text{if } \left\| x^i - x^j \right\| \leq \epsilon \\ 0, \qquad \text{otherwise} \end{cases}$$

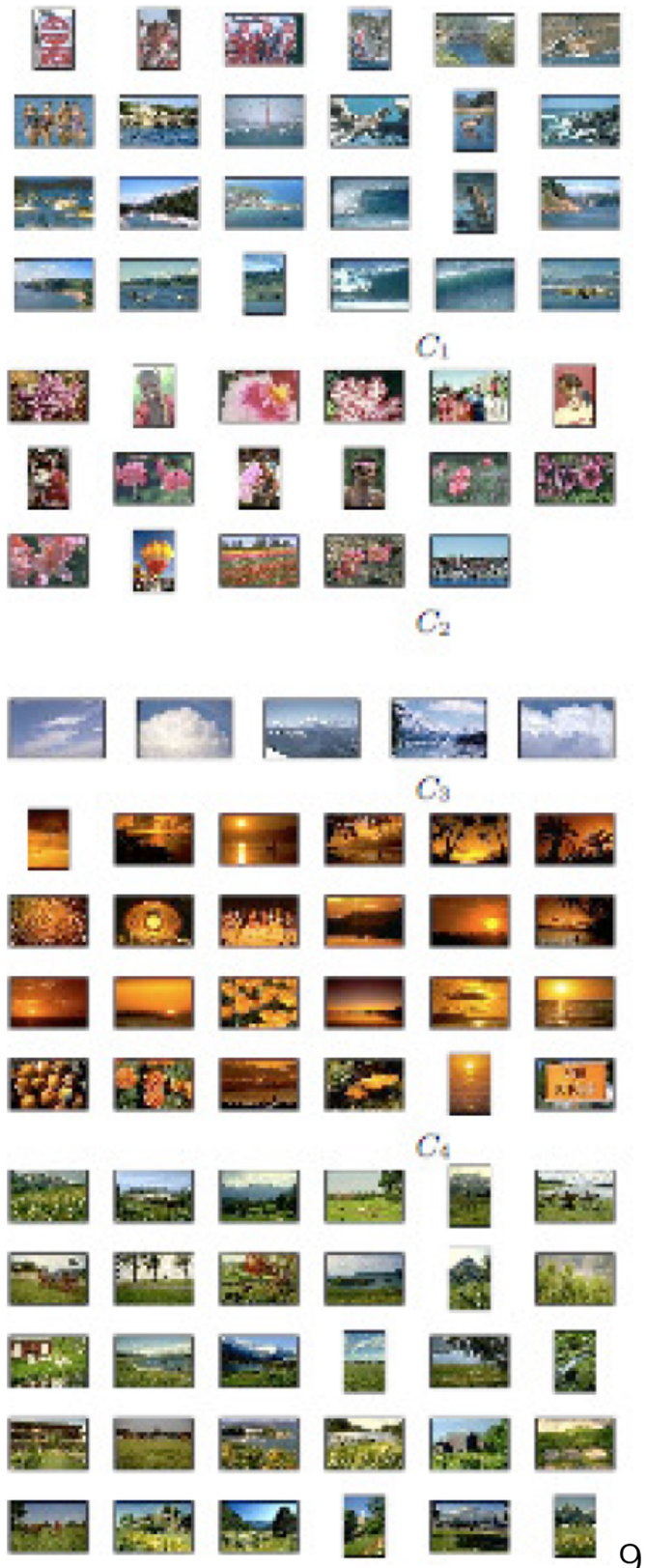


# Spectral clustering for vectorial data

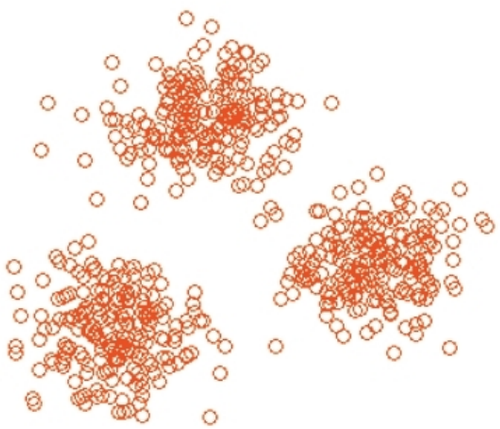
- Given $m$ nodes, $\{x^1, x^2, \dots x^m\} \in R^n$
- Step 1: build an adjacency matrix $A$ using nearest neighbors
- Step 2: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 3: form a special matrix $L = D - A$, the graph Laplacian
- Step 4: compute $k$ eigenvectors, $v^1, v^2, \dots, v^k$, 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

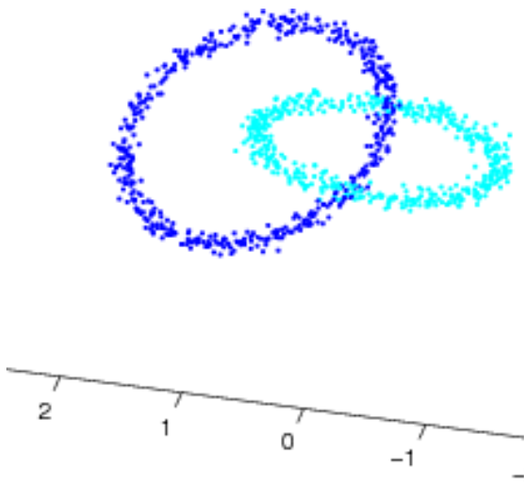
# Image databases

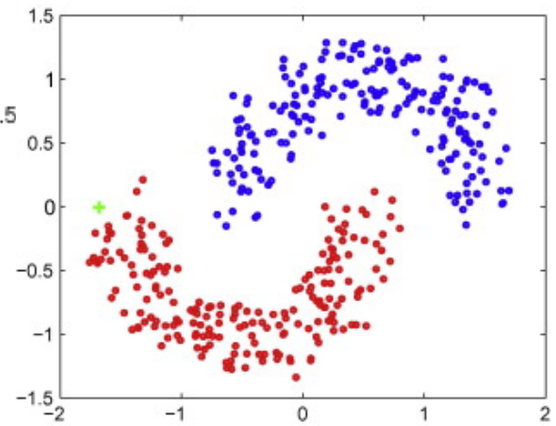


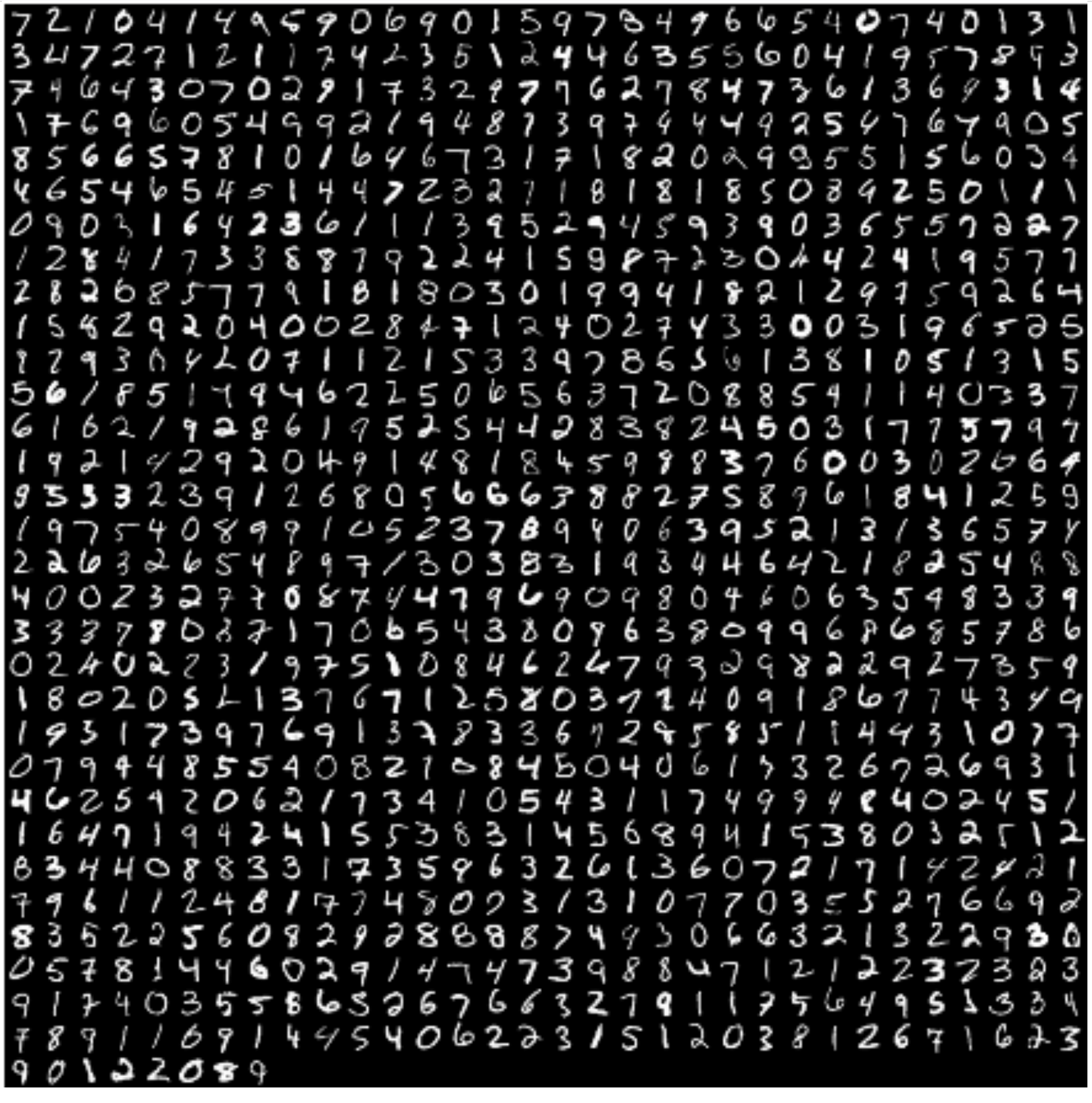

What are the relations between data points?



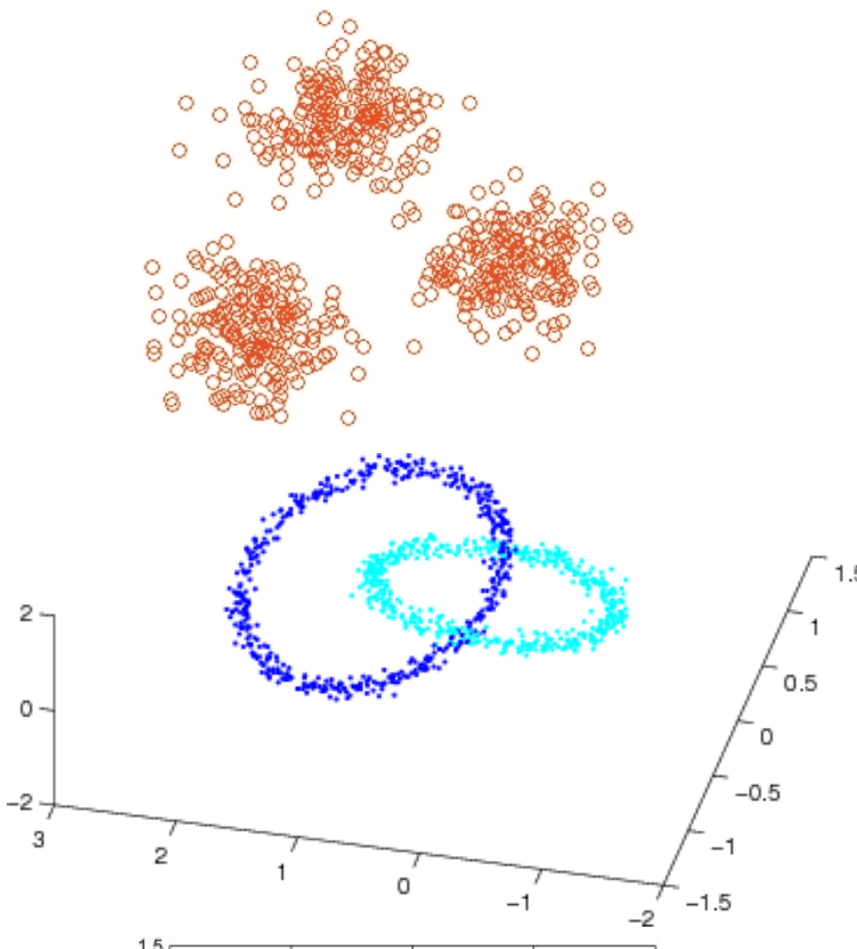


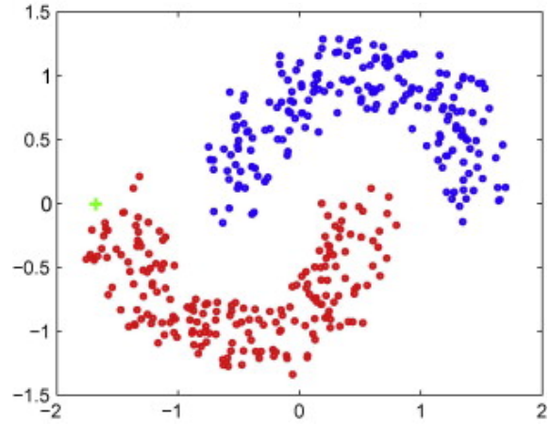



# Handwritten digits



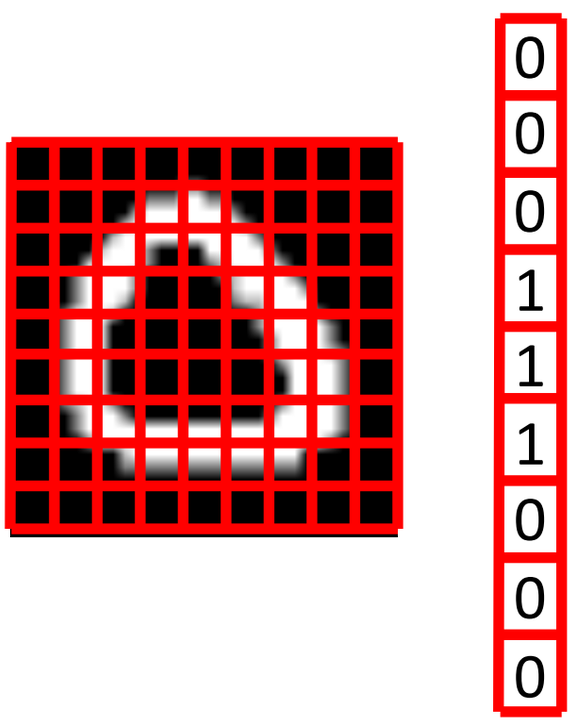
What are the relations between data points?



# 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

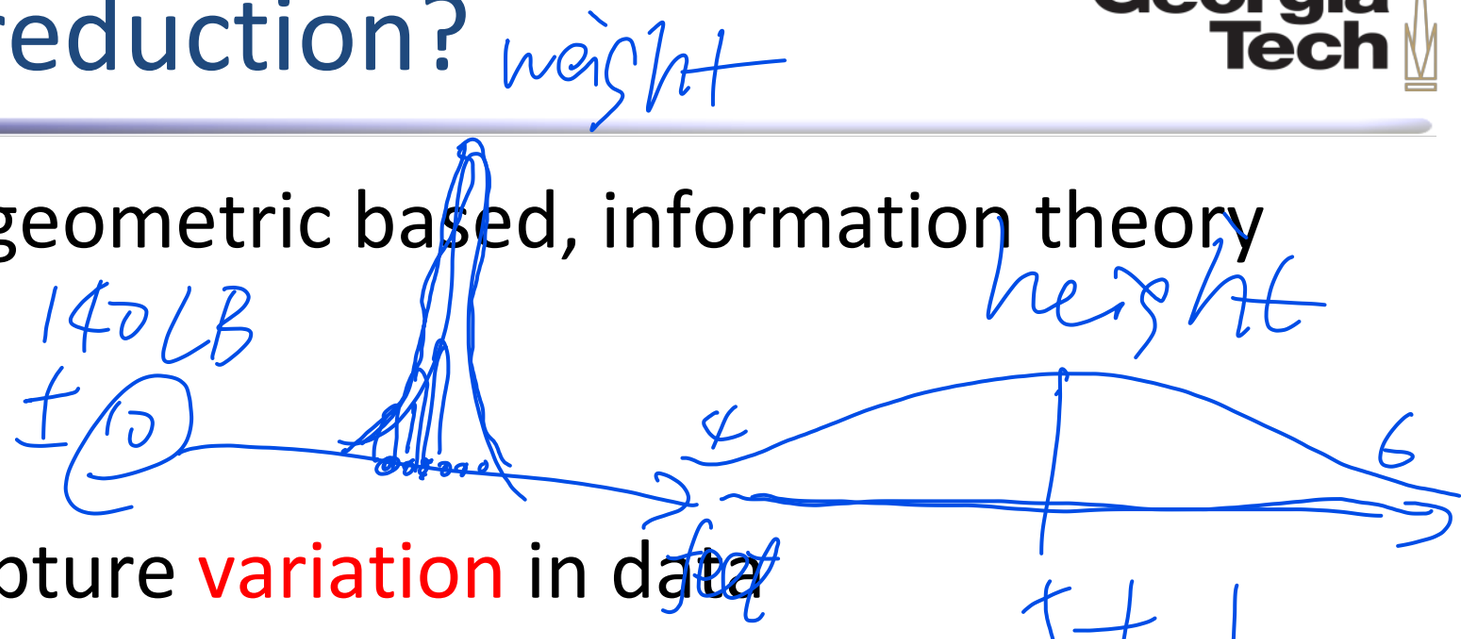


# Why dimensionality reduction and how to think

- 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

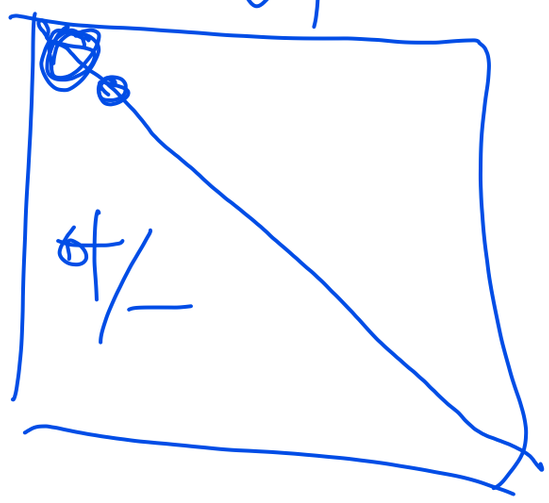
$$\frac{1}{m}\sum_{i=1}^{m}\left(w^\top x^i - w^\top \mu\right)^2$$

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

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

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

covariance matrix $C$

# 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

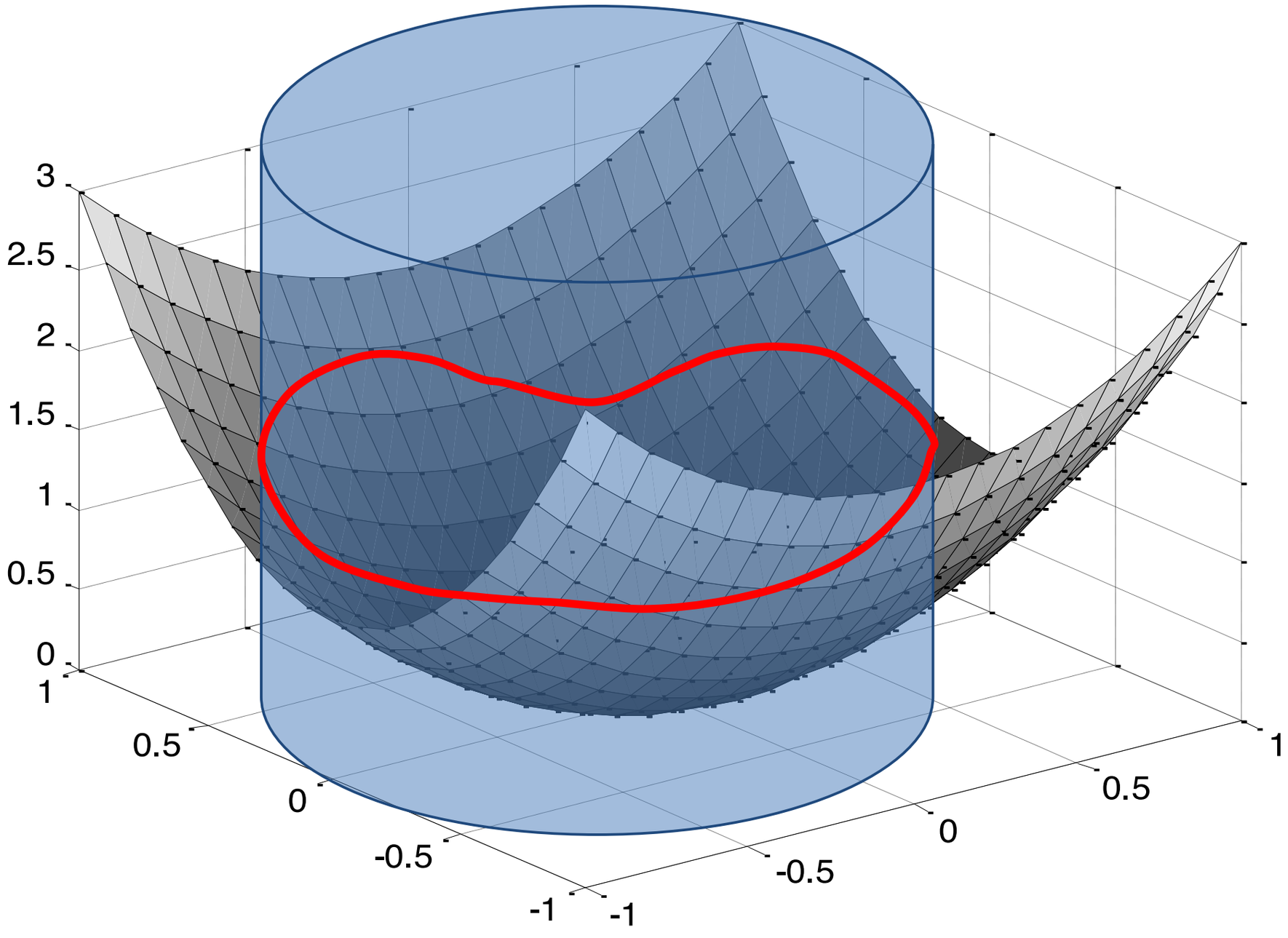
$$\max_{w:\|w\|=1} w^\top C w$$

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

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

# 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

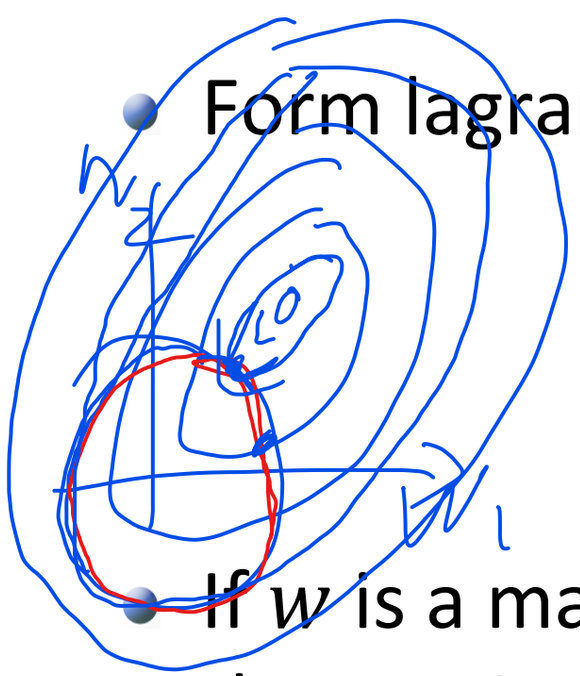
$$Cw = \lambda w$$

- 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:

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



- Form lagrangian function of the optimization problem

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

Necessary condition

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

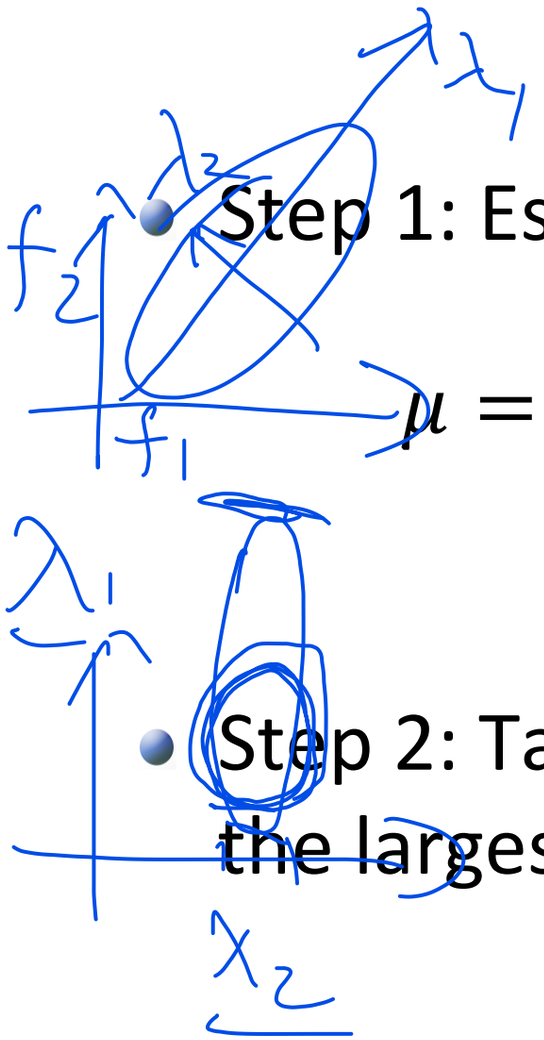
# Multiple principal directions

- 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 $\lambda_1$,
  - the second largest eigenvalue $\lambda_2$
  - …

# 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

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

Principal directions

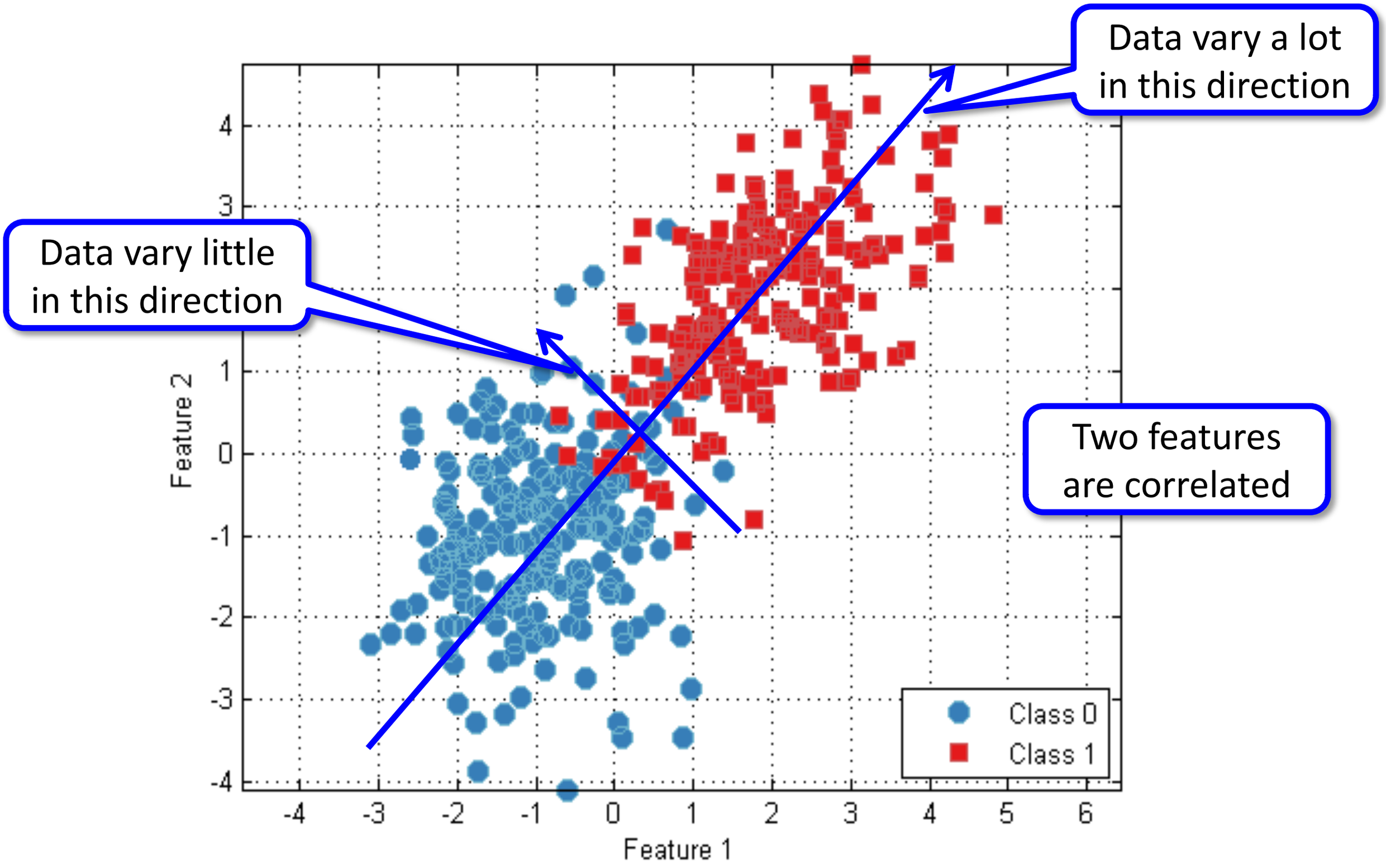
- Step 2: Take the eigenvectors $w^1, w^2, \ldots$ of $C$ corresponding to the largest eigenvalue $\lambda_1$, the second largest eigenvalue $\lambda_2$ …

- Step 3: Compute reduced representation

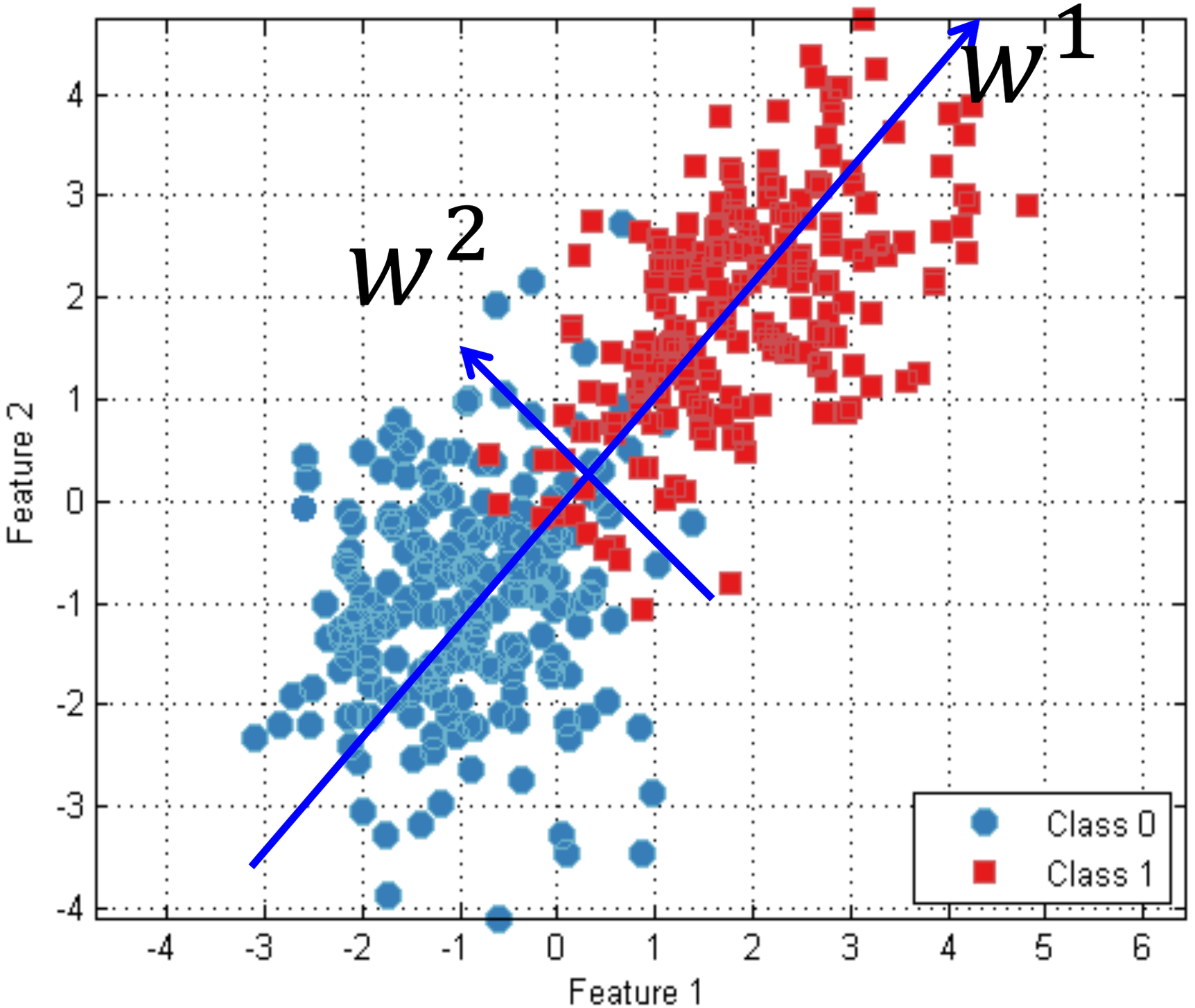
$$z^i = \begin{pmatrix} w^{1\mathsf{T}}(x^i - \mu)/\sqrt{\lambda_1} \\ w^{2\mathsf{T}}(x^i - \mu)/\sqrt{\lambda_2} \\ \vdots \end{pmatrix}$$

Normalize by standard deviation

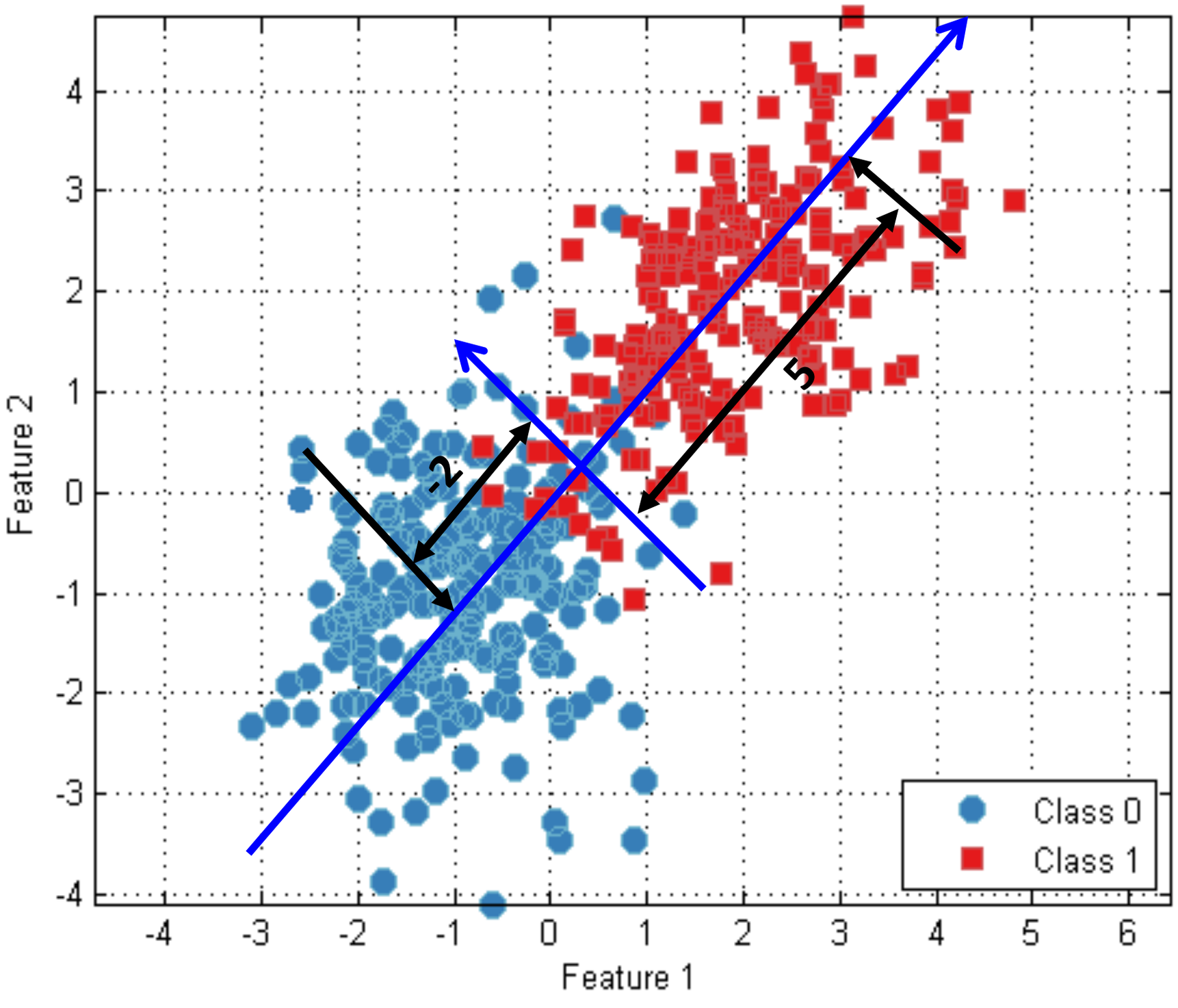
# An example


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

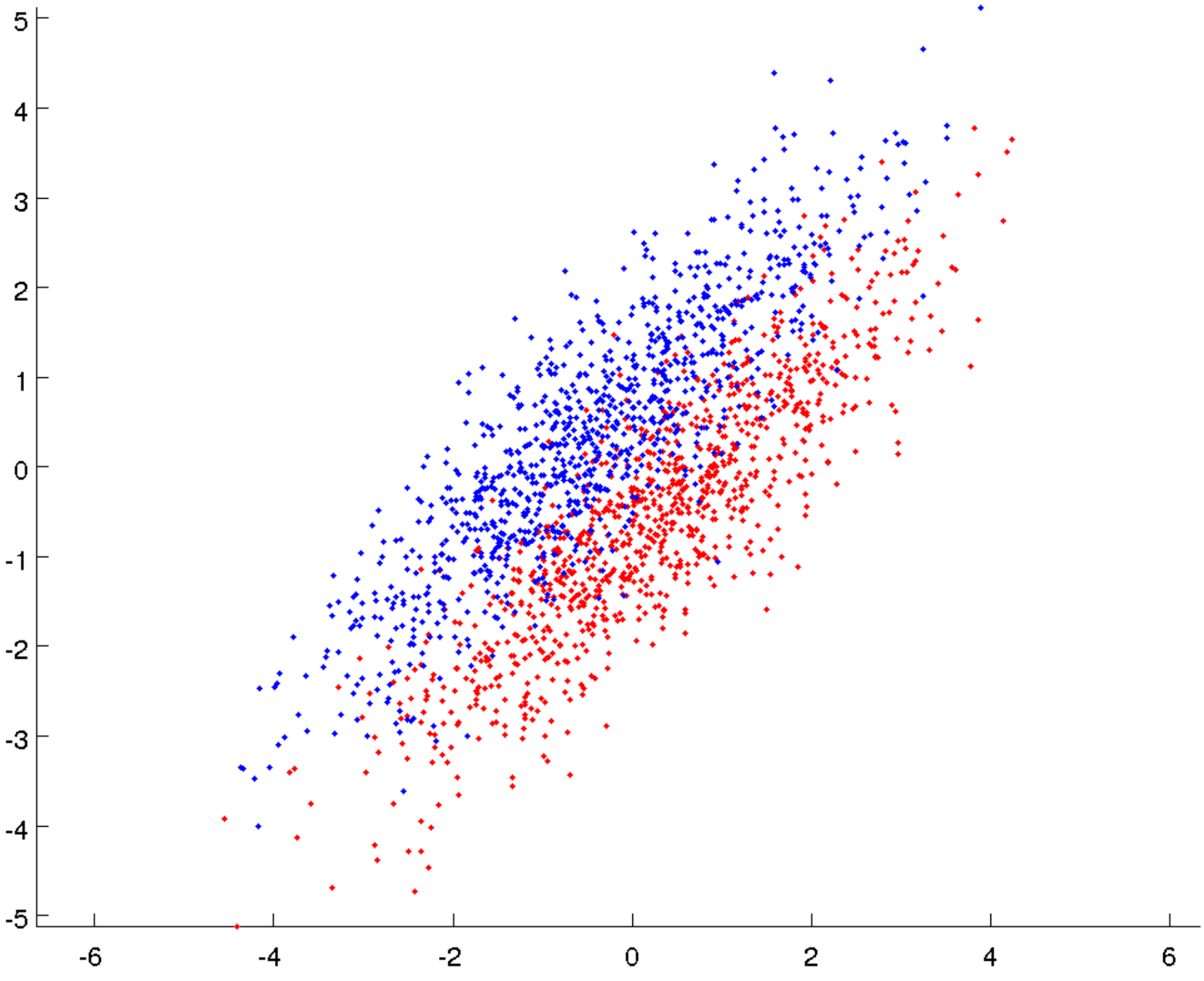
# Principal direction of the data



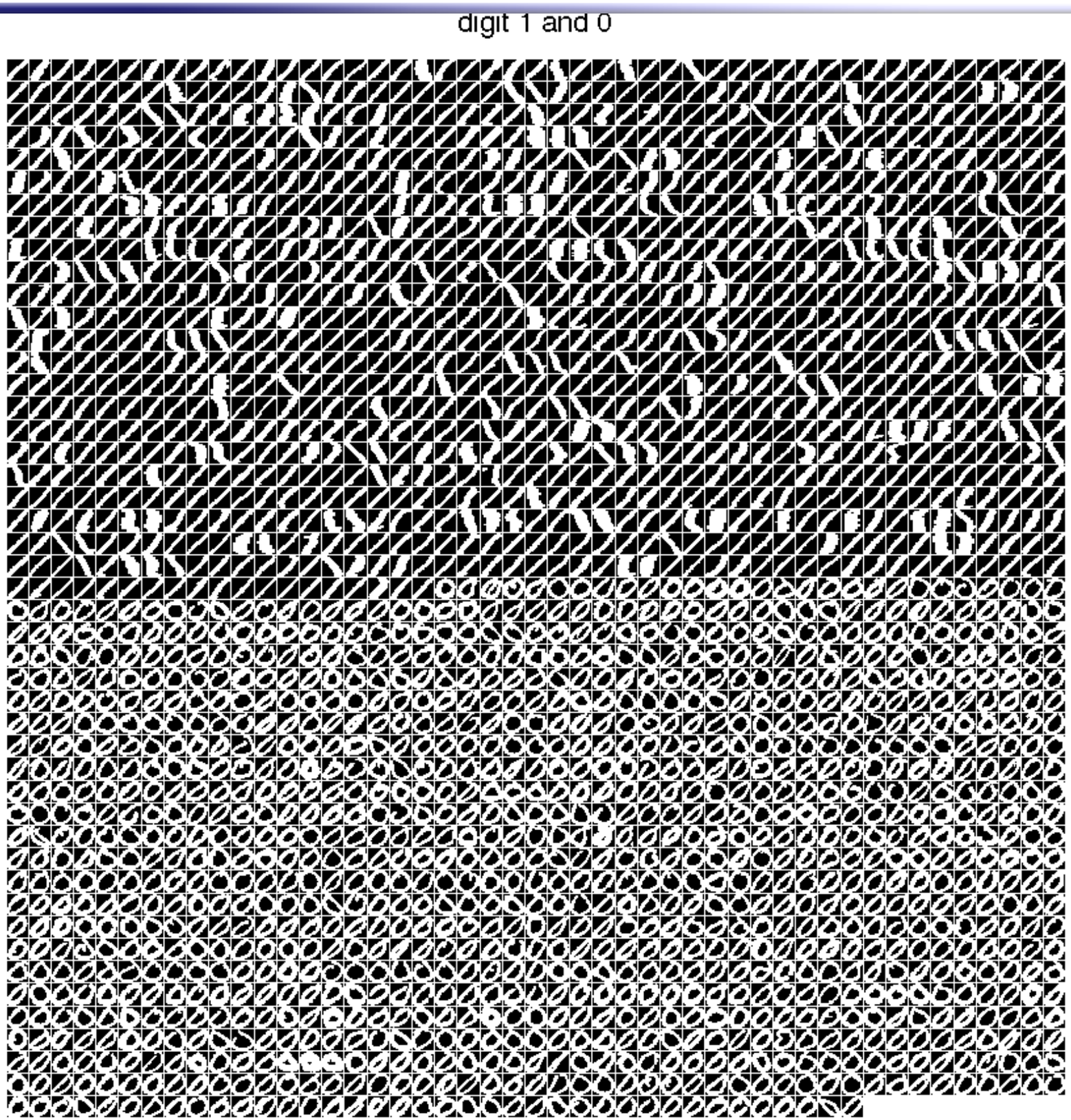
# Reduce to 1 dimension



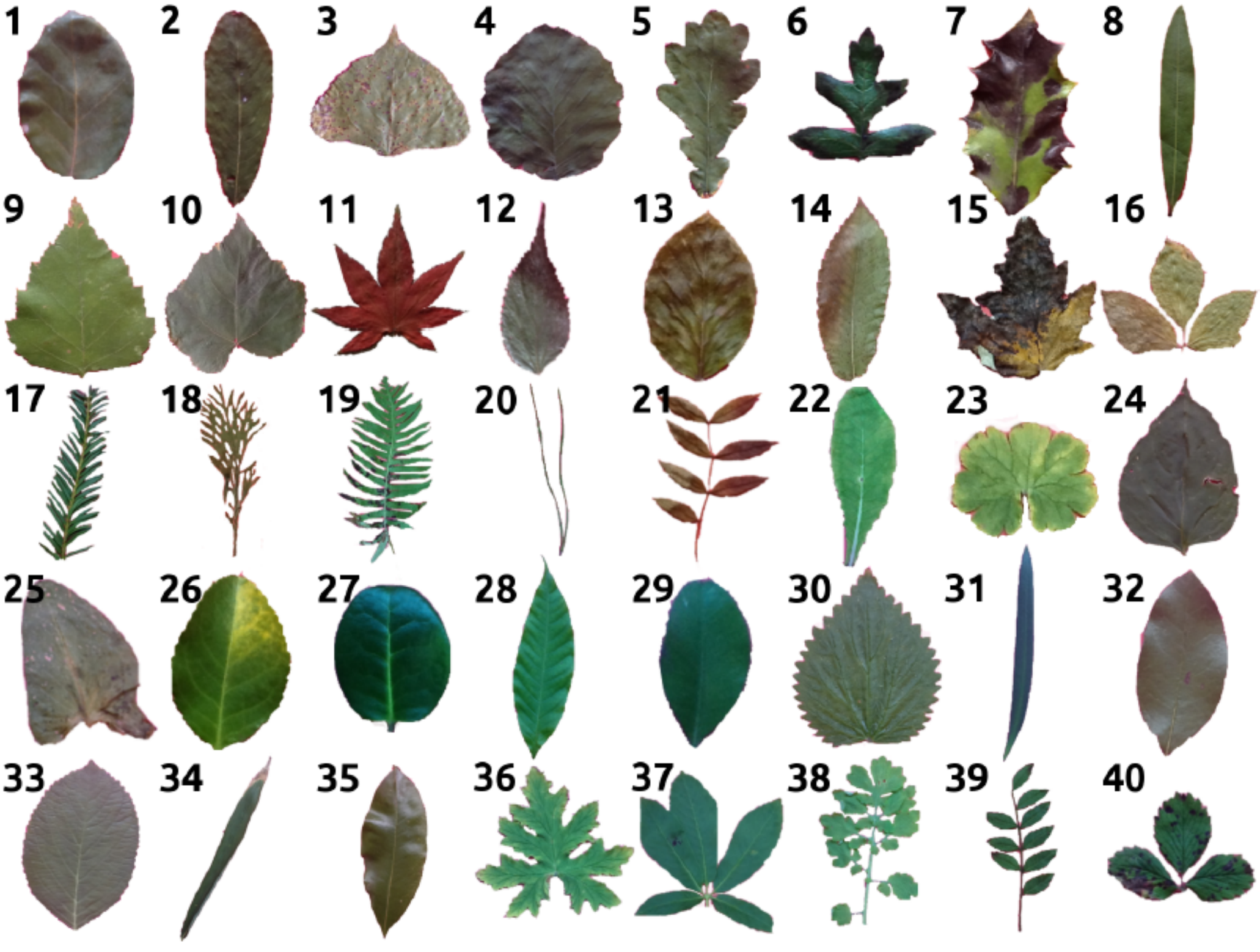
# Are principal components good for classification?



# Run demo PCA_digits.m



# Run demo PCA_leaf.m



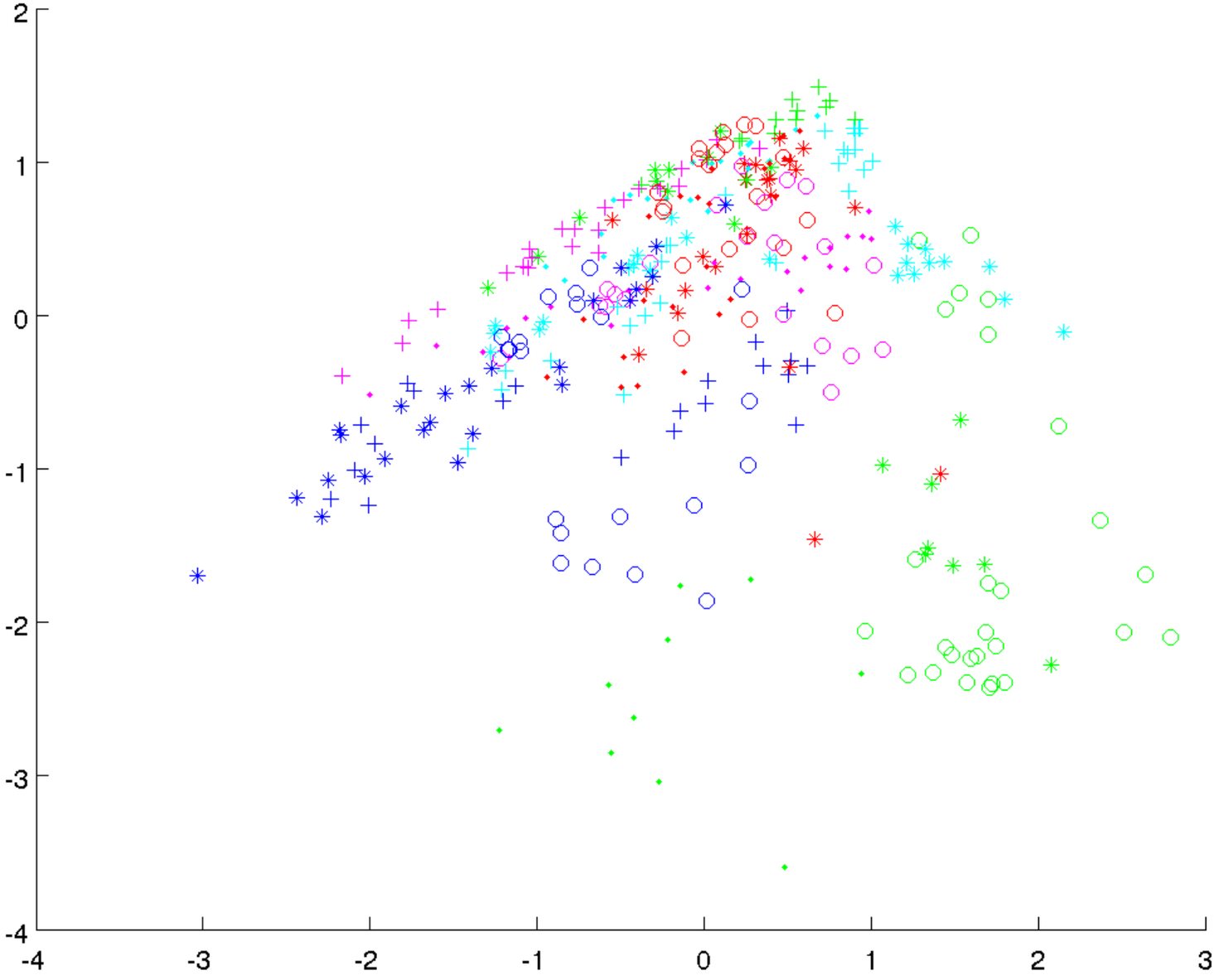
# Input features (representation)

| 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 $\partial I$ 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. $\sigma^2$ is normalized by $(L-1)^2$ to ensure that $R \in [0,1]$. |
| *Third moment* | $\mu_3$ 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 $\Sigma$, $\Sigma_{ii}$ are known as the singular values of M.

- Let $v$ be the i$^{th}$ column of $V$, and $u$ be the i$^{th}$ column of $U$, and $\sigma$ be the i$^{th}$ diagonal element of $\Sigma$

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

# 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
    - $M^{\mathsf{T}}\mathrm{u} = \sigma v$
    - $\Rightarrow M M^{\mathsf{T}}\mathrm{u} = \sigma M v$
    - $\Rightarrow M M^{\mathsf{T}}\mathrm{u} = \sigma^2 u$
    - $\Rightarrow Cu = \lambda u$

# How to recover the original data point?

- Given data mean $\mu$, principal directions $w^1$, $w^2$, … and the corresponding eigenvalues $\lambda_1$, $\lambda_2$, …

- Recover $x^i$ from the reduced representation $z^i$ approximately?

$$z^i = \begin{pmatrix} z_1^i \\ z_2^i \\ \vdots \end{pmatrix} = \begin{pmatrix} w^{1\top}(x^i - \mu)/\sqrt{\lambda_1} \\ w^{2\top}(x^i - \mu)/\sqrt{\lambda_2} \\ \vdots \end{pmatrix}$$

- $x^i \approx \hat{x}^i = \mu + z_1^i \cdot \sqrt{\lambda_1} \cdot w^1 + z_2^i \cdot \sqrt{\lambda_2} \cdot w^2 + \cdots$

- Matrix-vector expression?

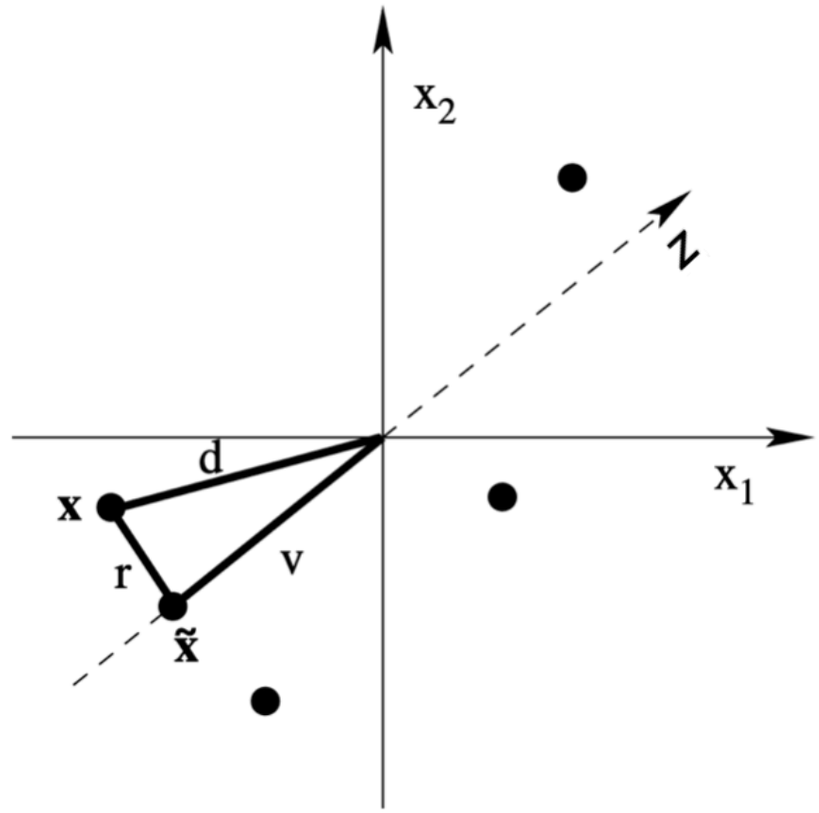
# Two alternatives

- Maximize the variance of the code vectors

$$\max \sum_j \text{Var}(z_j) = \frac{1}{N} \sum_j \sum_i (z_j^{(i)} - \bar{z}_j)^2$$

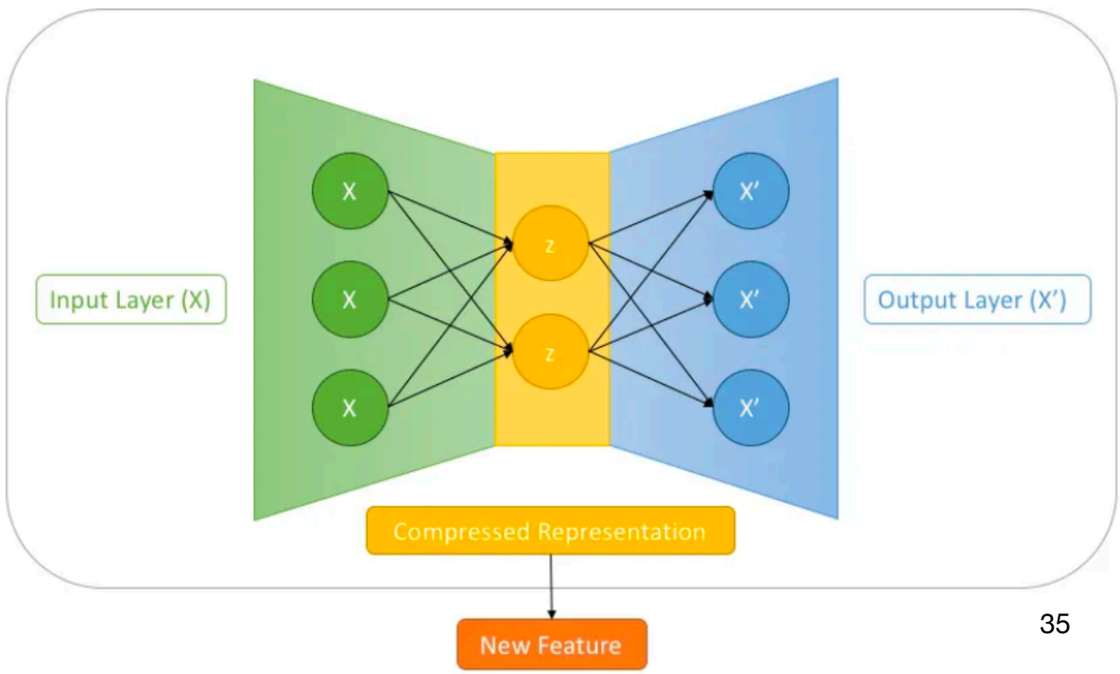
$$= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)} - \bar{\mathbf{z}}\|^2$$

$$= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)}\|^2$$



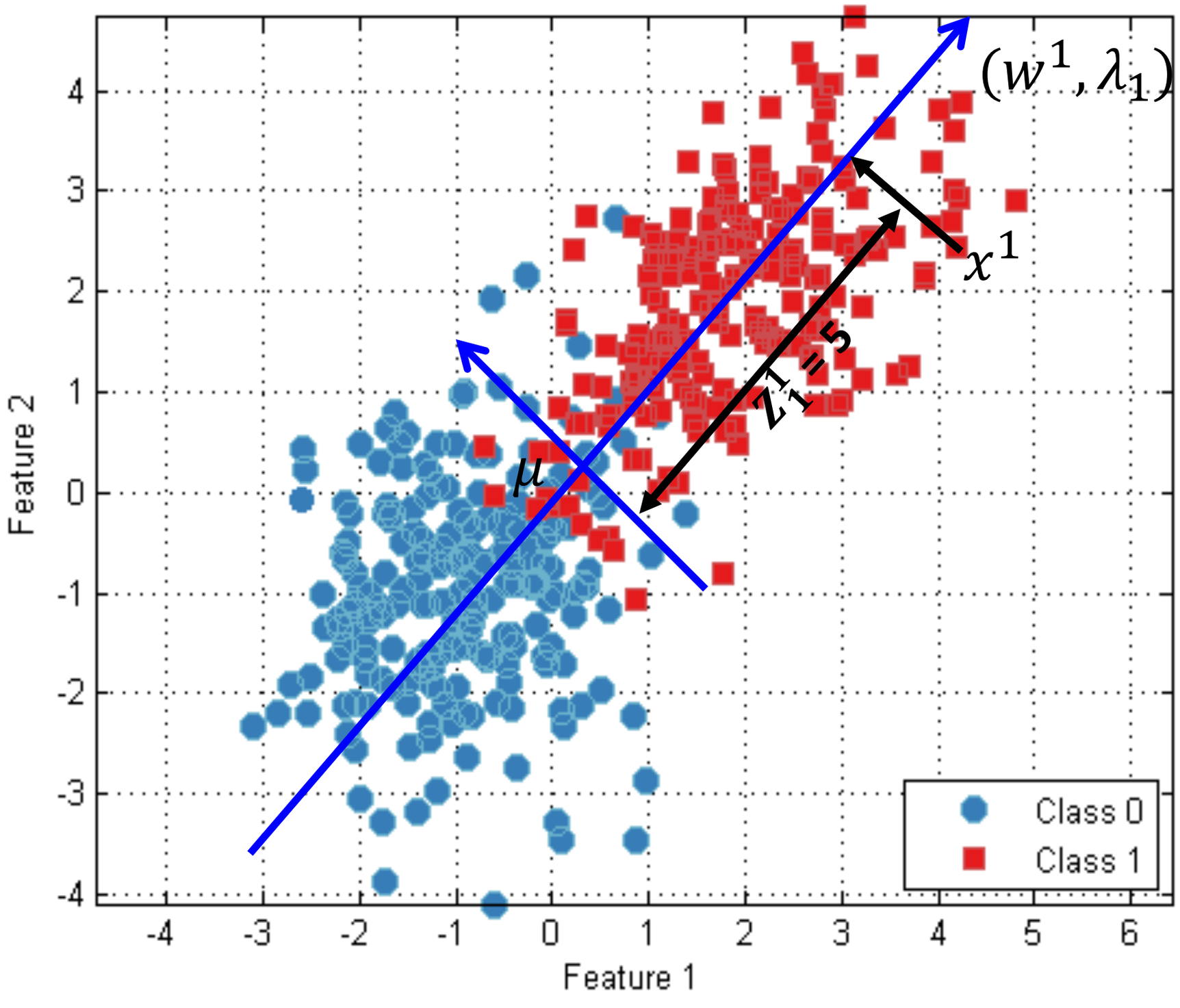

- Minimize the reconstruction error

$$\min \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2$$

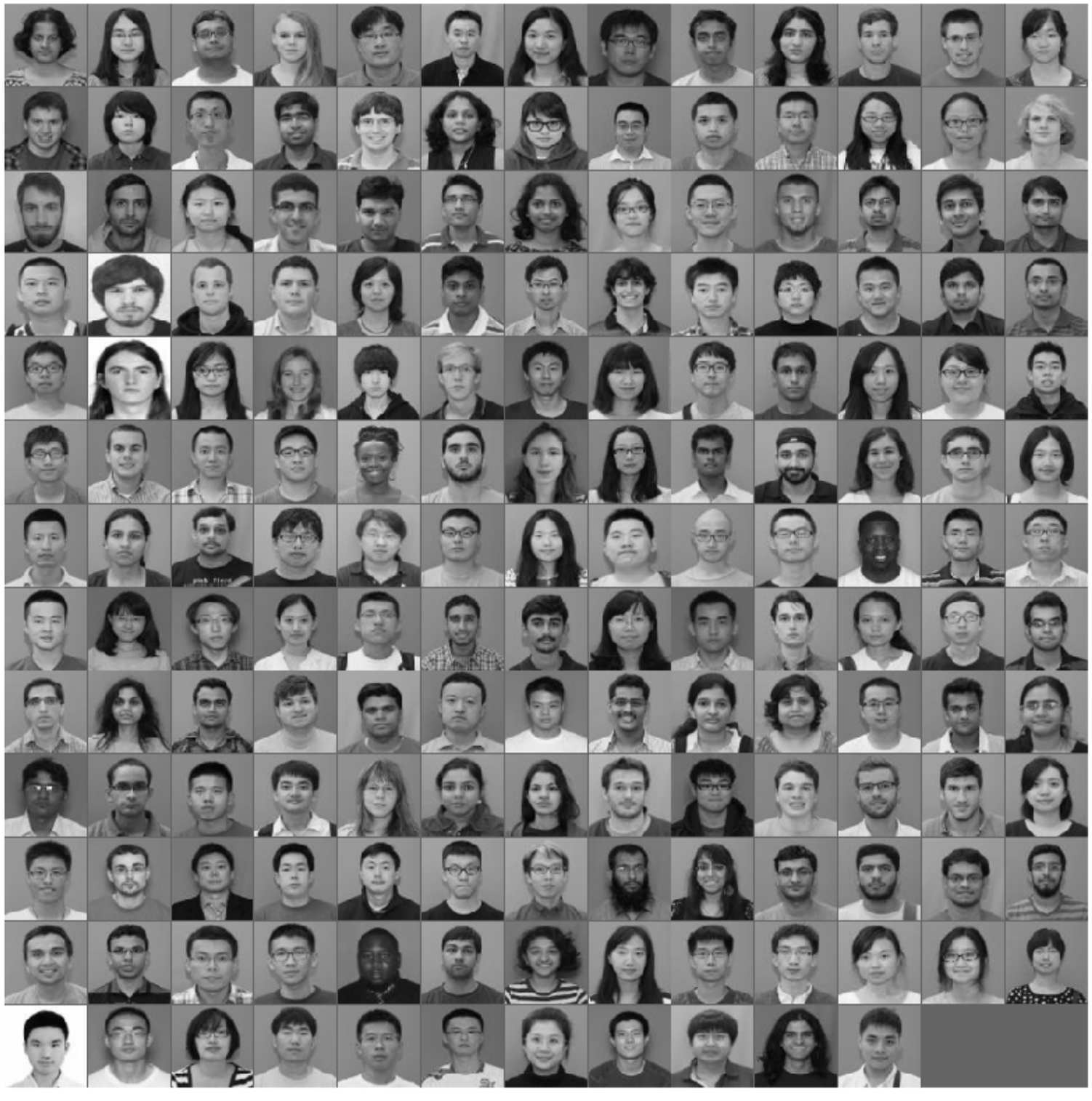


# Reduce to 1d, reconstruct 2d

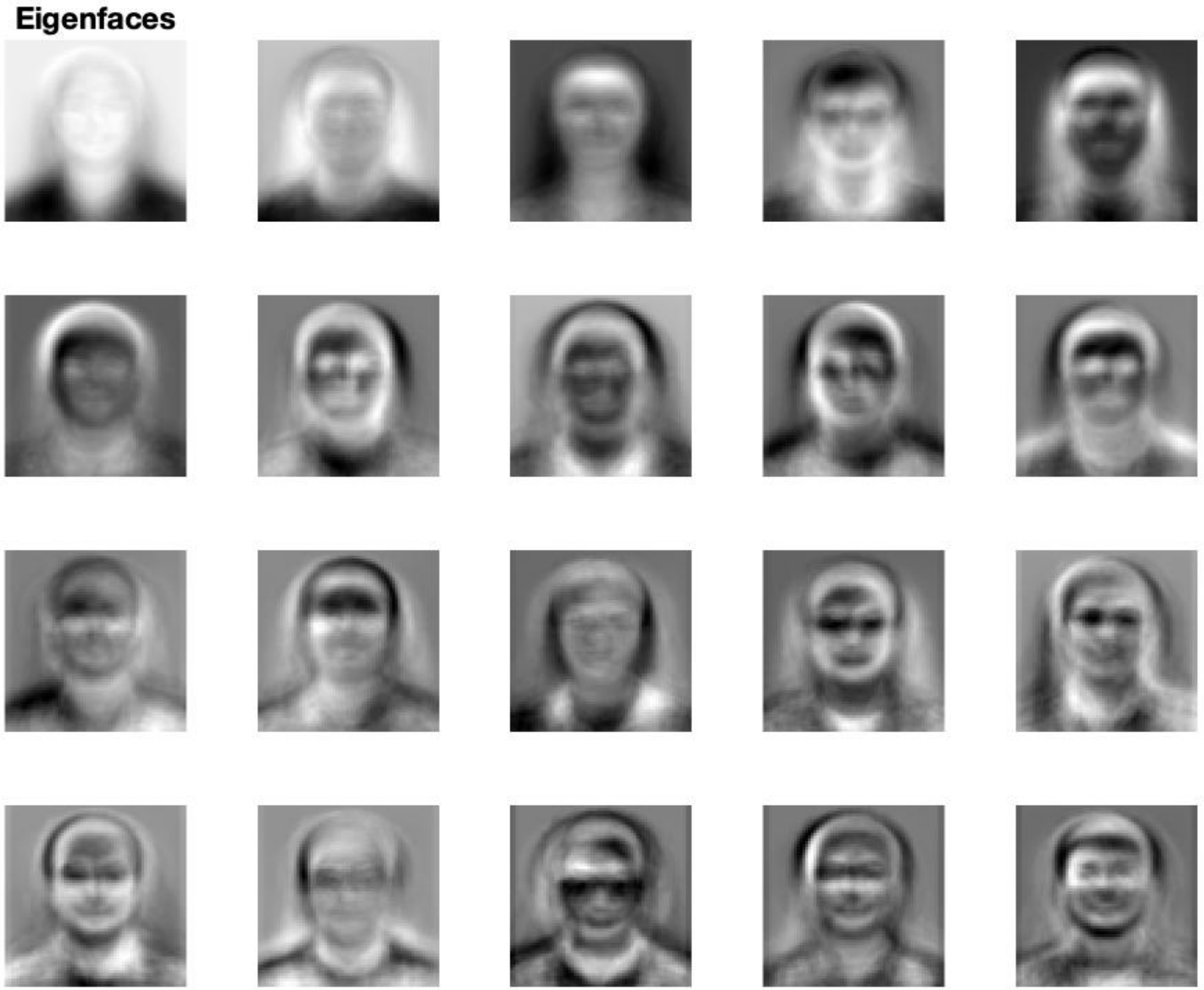
$$x^1 \approx \hat{x}^1 = \mu + z_1^1 \cdot \sqrt{\lambda_1} \cdot w^1$$



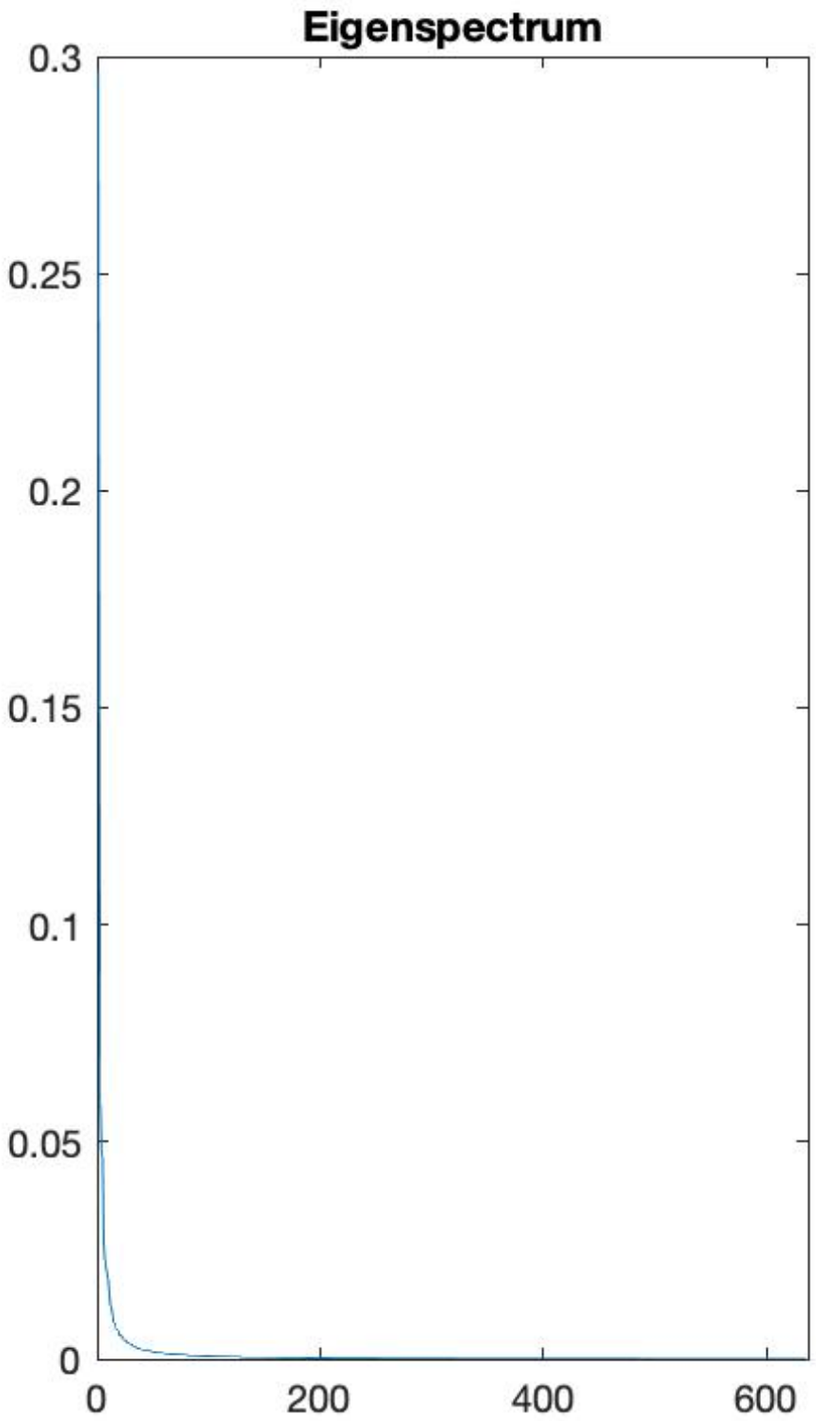
# Eigenfaces (demo test_eigenface.m)

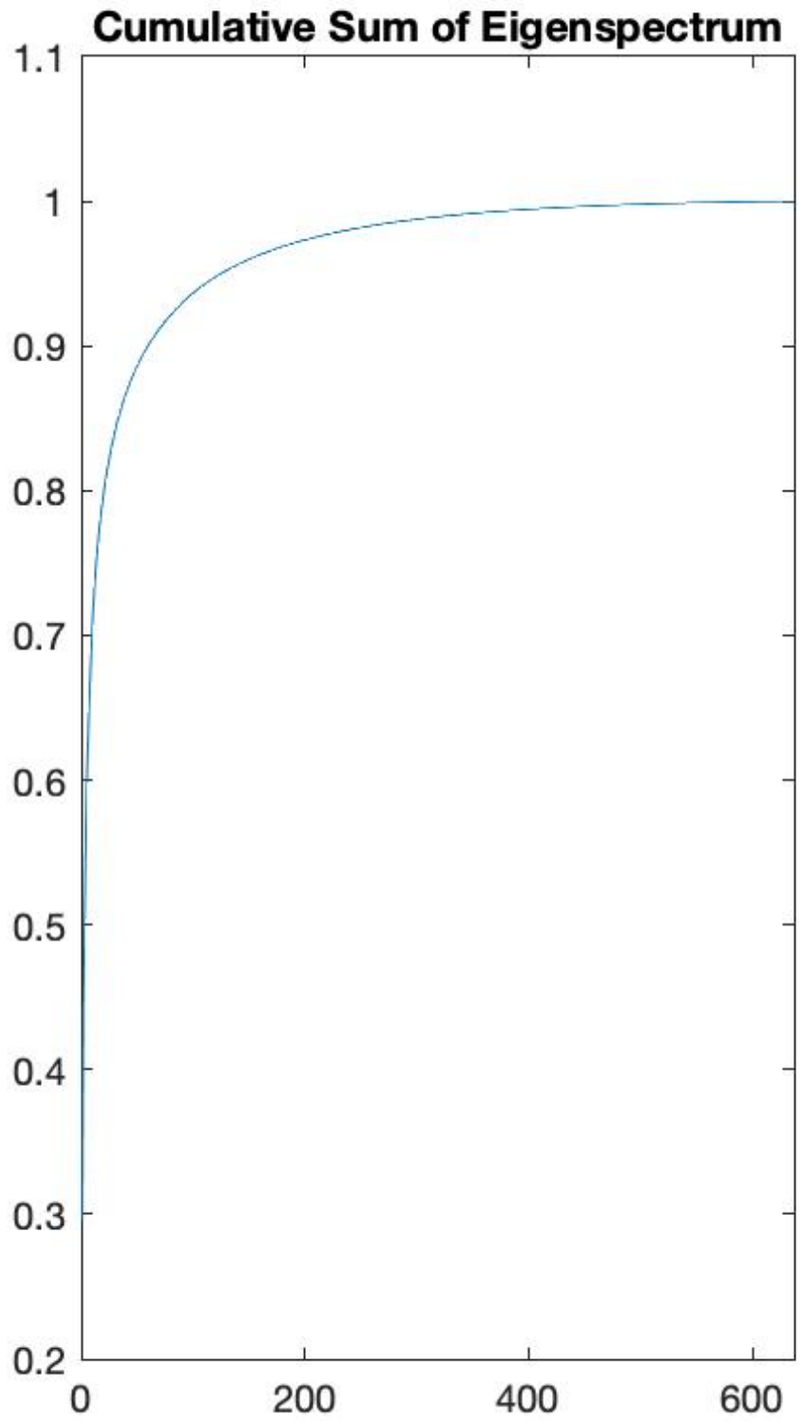


# Eigenfaces (demo test_eigenface.m)

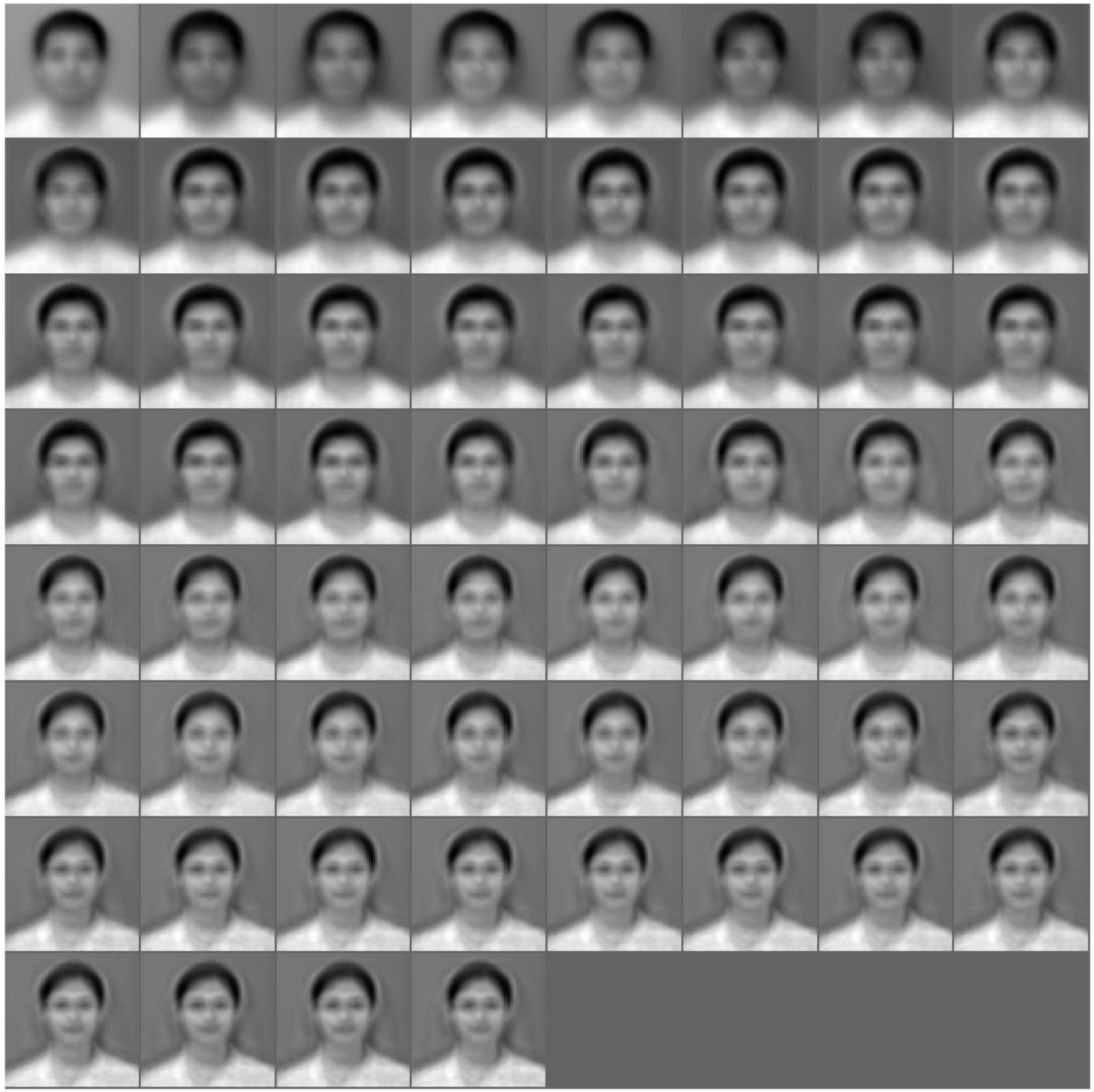


# Eigenfaces (demo test_eigenface.m)

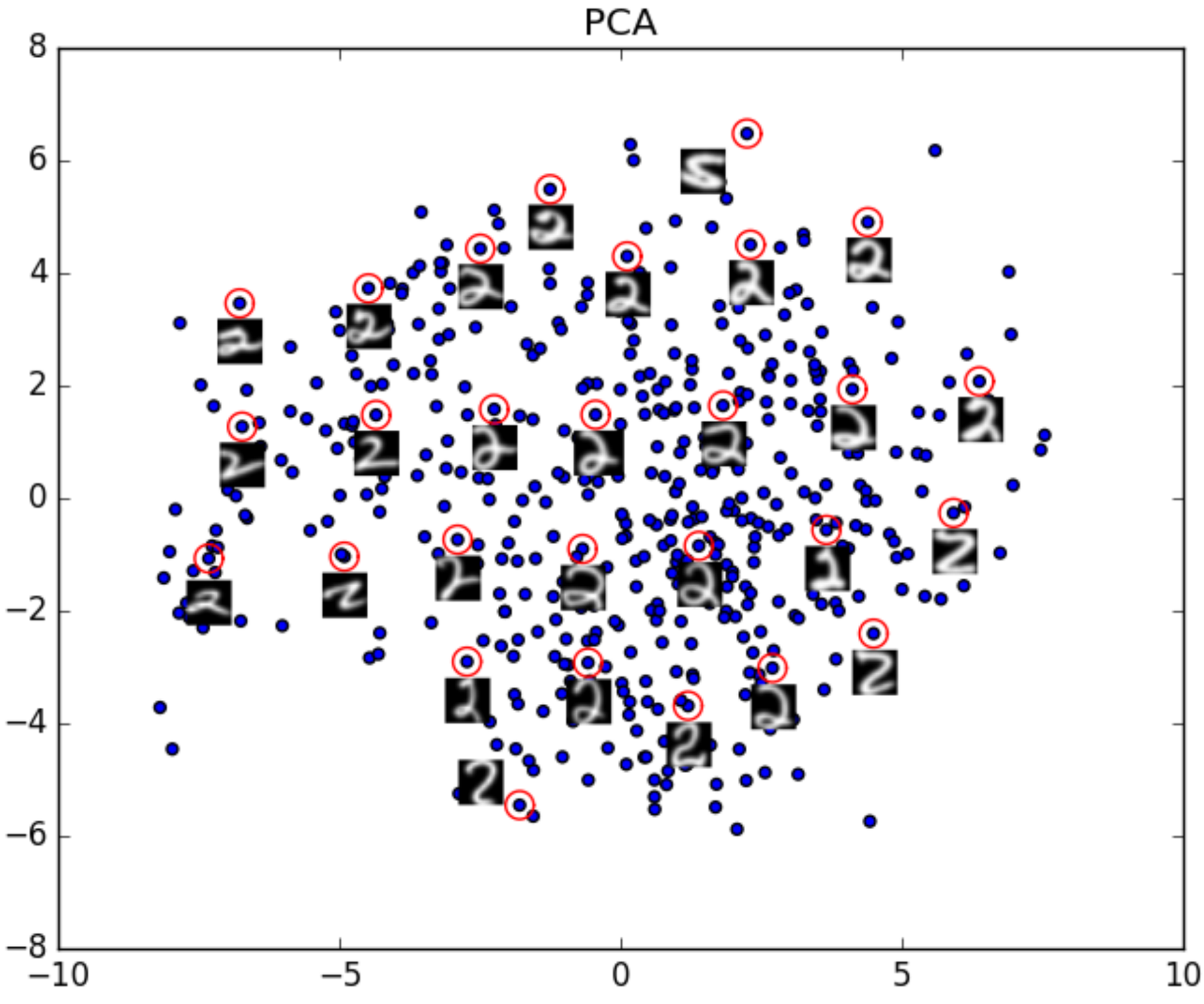


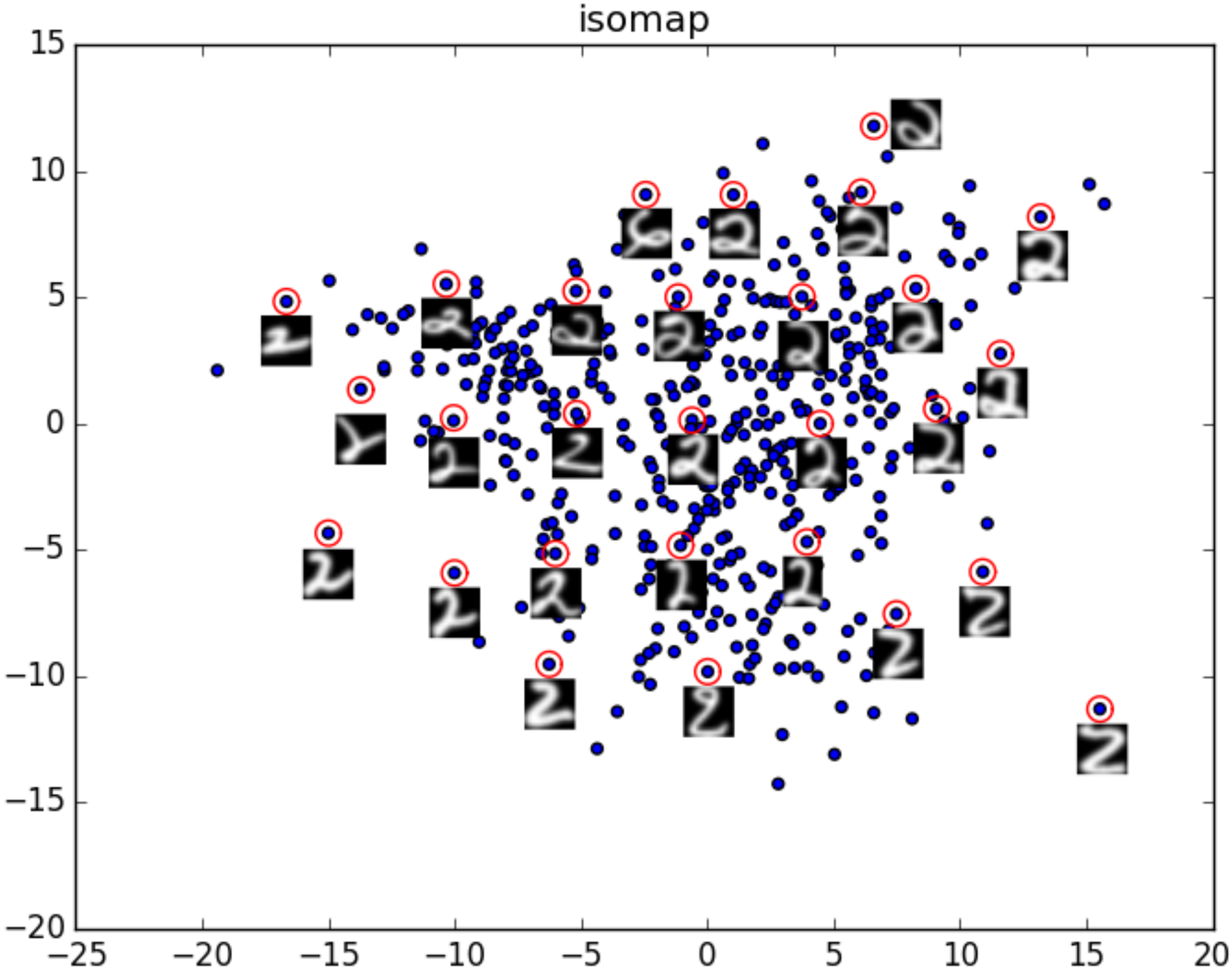
# Eigenfaces (demo test_eigenface.m)



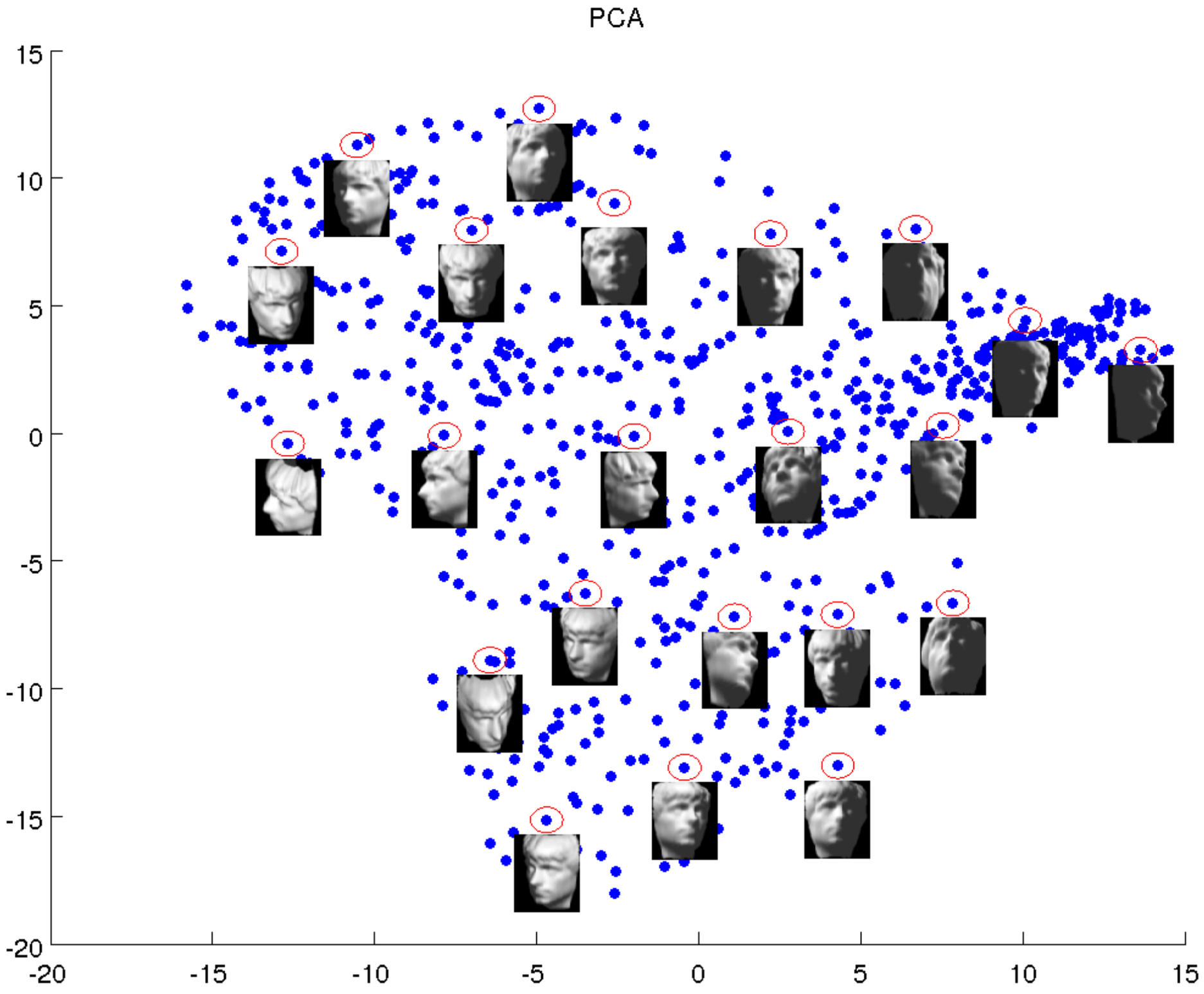
# Is the principal direction interpretable?



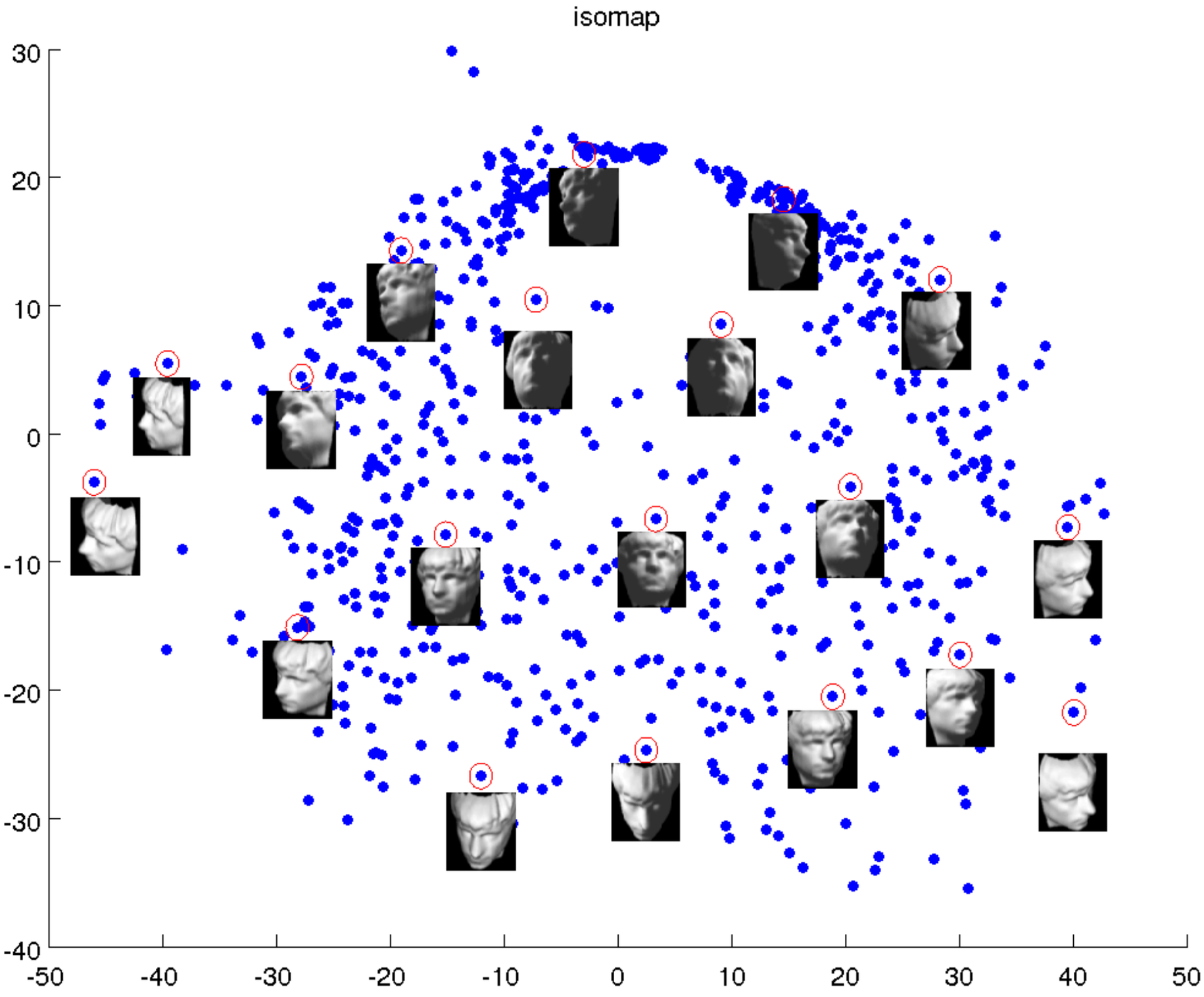
# How about this one?



# Is the principal direction interpretable?

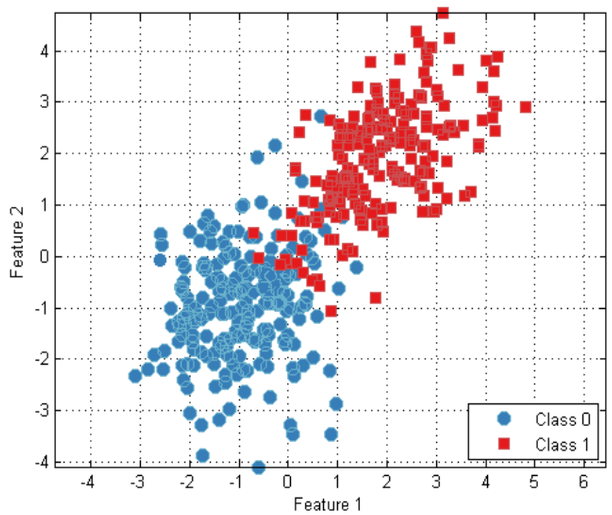


# How about this one?

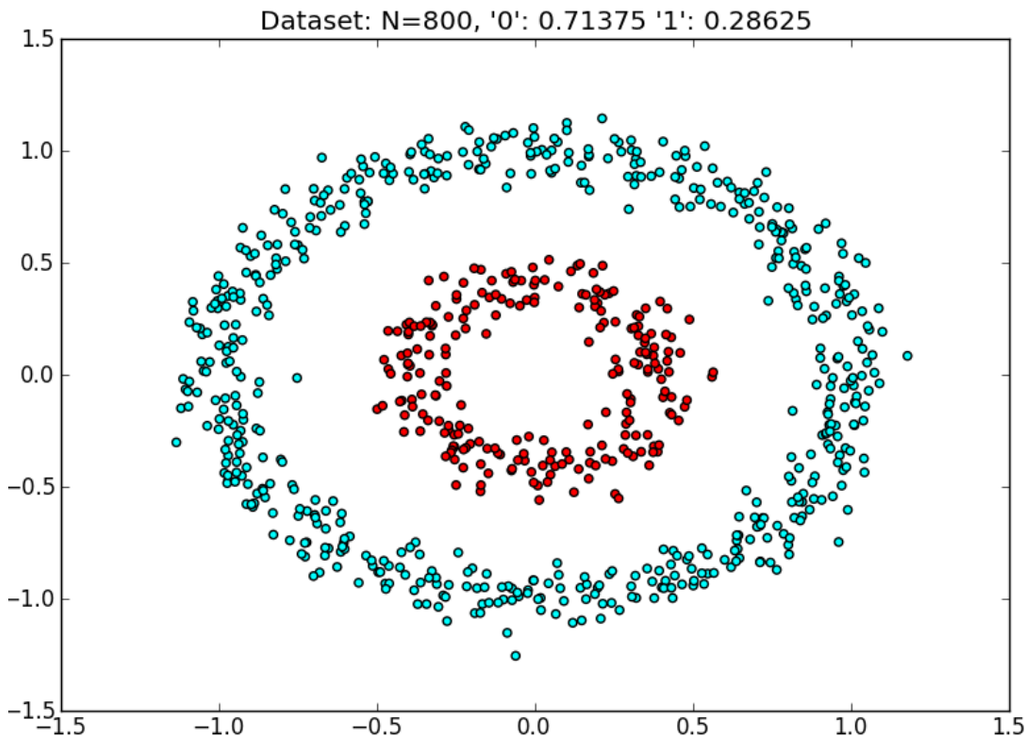


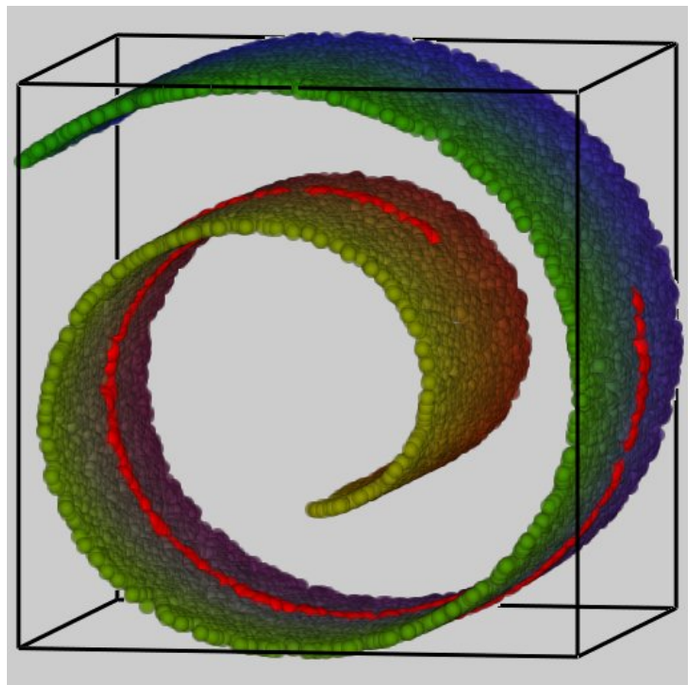
# Limitation of PCA and SVD

- Suitable when variables are linearly correlated

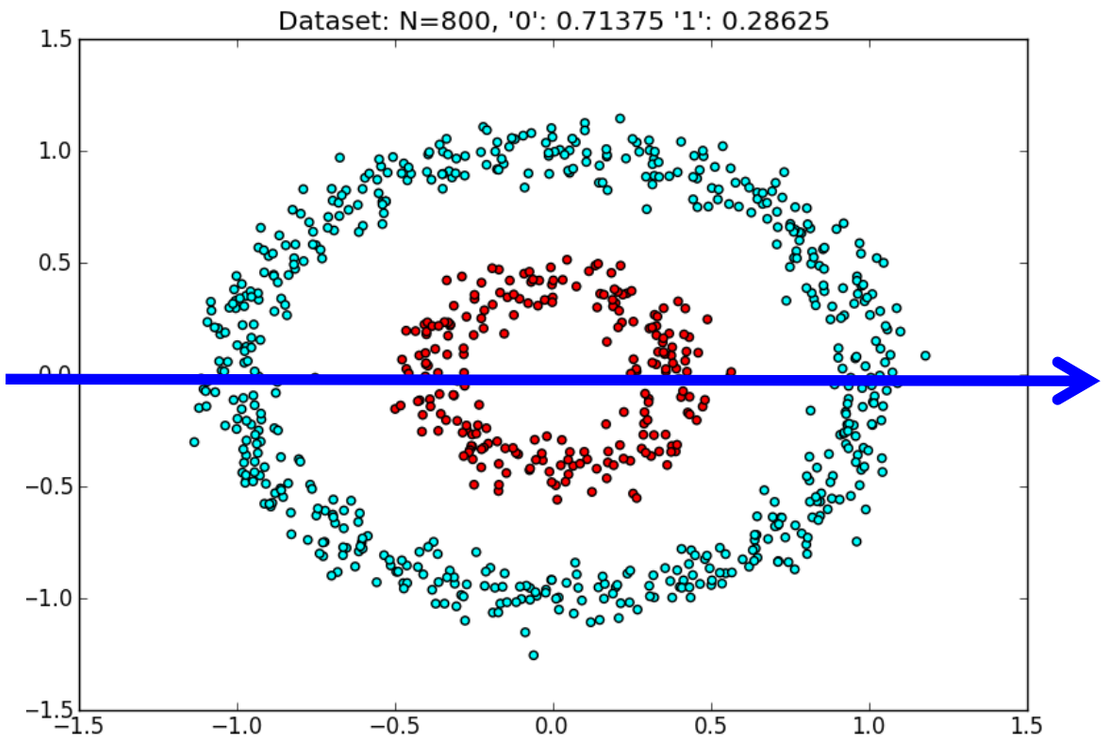


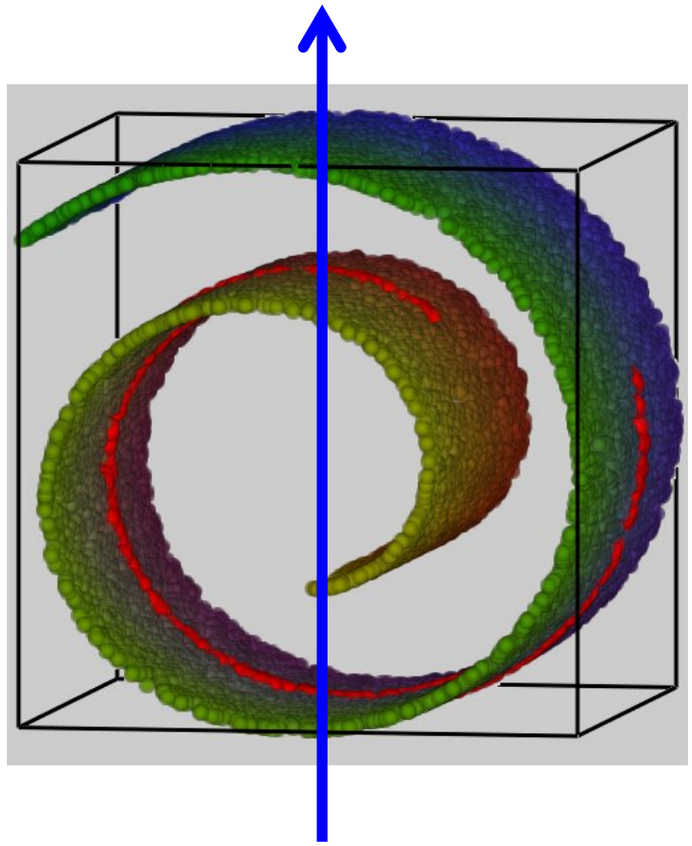

- Not suitable when nonlinear structures are present



# What's wrong with PCA

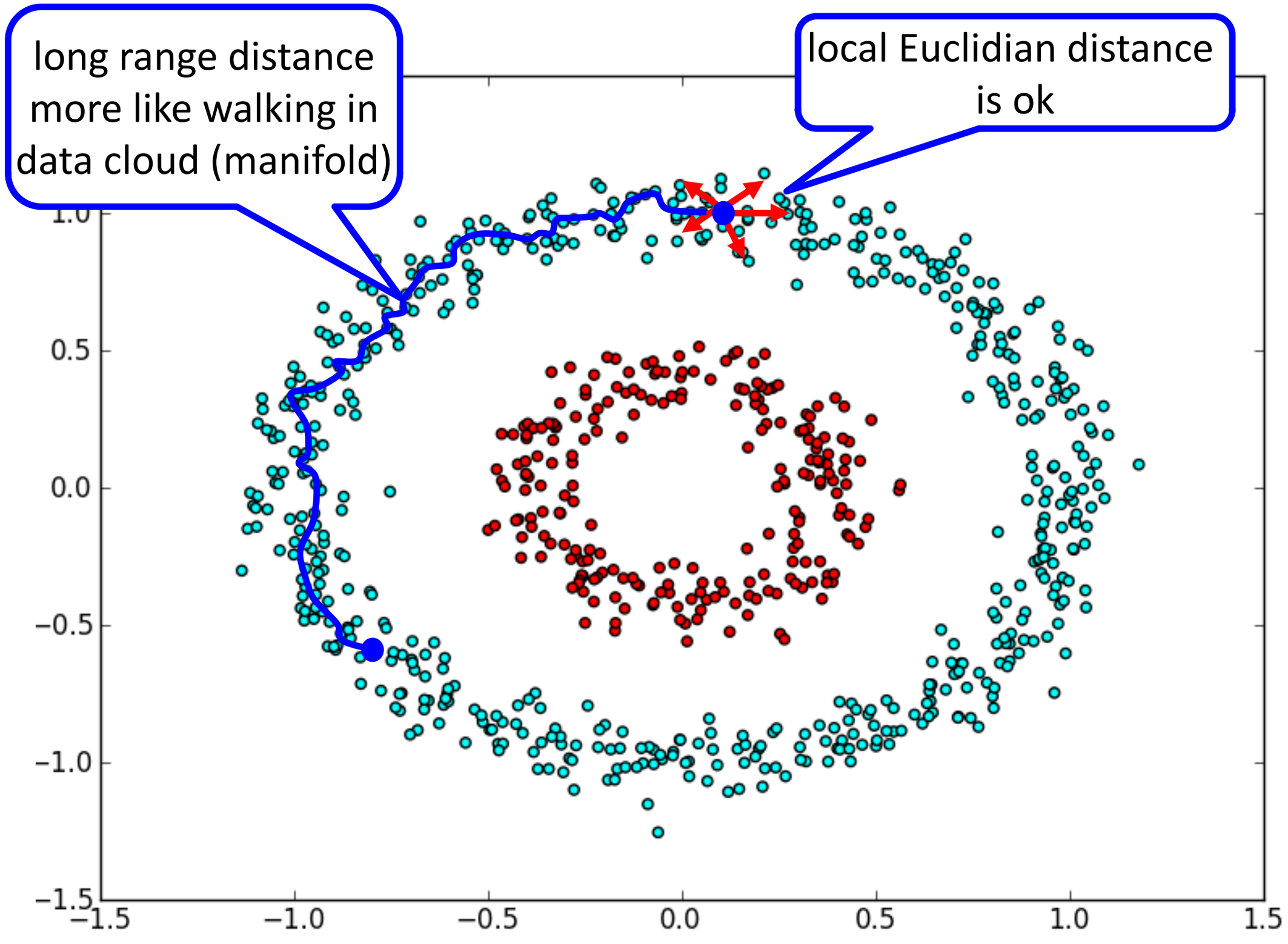


- PCA uses linear projection $w^\top x$, implicitly assuming Euclidean distance is the dissimilarity (distance) measure

- When there are nonlinear structure, Euclidean distance is not the right distance measure globally

# What's a reasonable distance measure



# Isomap

- Key idea: produce low dimensional representation which preserves "walking-distance" over the data cloud (manifold)

  - Find neighbors $N(i)$ of each data point, $x^i$, within distance $\epsilon$ and let $A$ be the adjacency matrix recording neighbor Euclidean distance

  - Find shortest path distance matrix $D$ between each pairs of points, $x^i$ and $x^j$, based on $A$

  - Find low dimensional representation which preserves the distances information in $D$