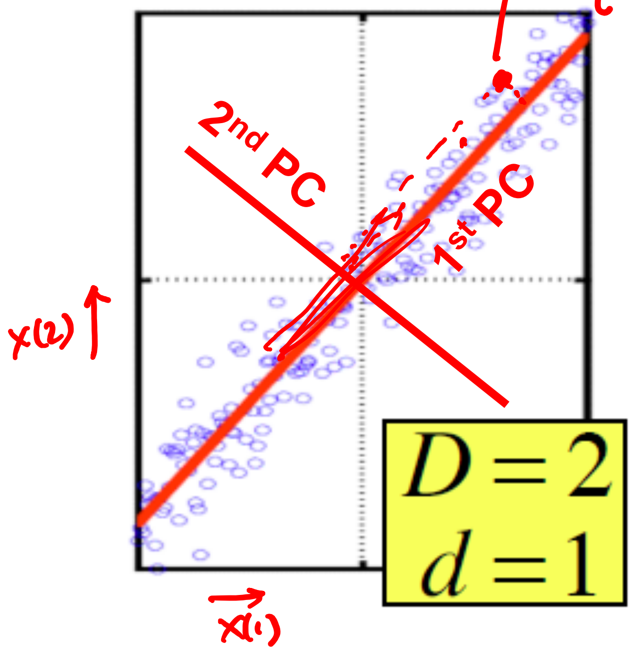
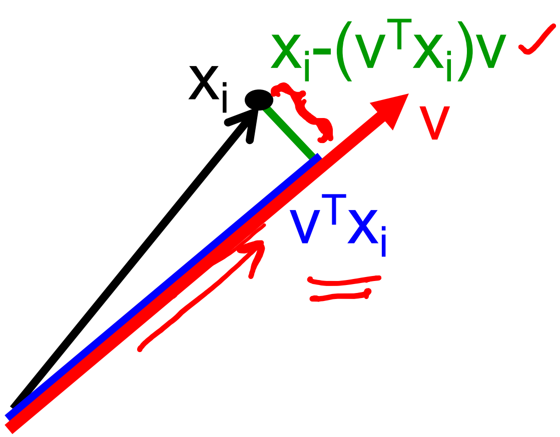
Principal Component Analysis (PCA)

$x_i = \begin{bmatrix} x_i(1) \\ x_i(2) \end{bmatrix}$ 2 dim $\longrightarrow$ $v^T x_i$ 1 dim

$\overset{\shortparallel}{D}$ $\quad\quad$ $\overset{\shortparallel}{d}$





Principal Components (PC) are orthogonal unit norm directions that capture most of the variance in the data

1st PC – direction of greatest variability in data

2nd PC – Next orthogonal (uncorrelated) direction of greatest variability

(remove all variability in first direction, then find next direction of greatest variability)

And so on …

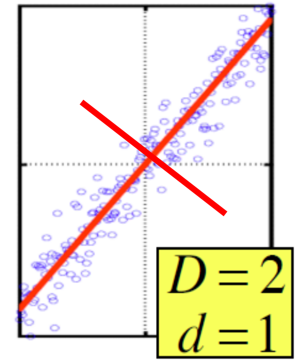
$D \rightarrow d << D$

Principal Component Analysis (PCA)

Let $v_1, v_2, \ldots, v_d$ denote the principal components

Orthogonal and unit norm $\quad v_i^T v_j = 0 \quad i \neq j$

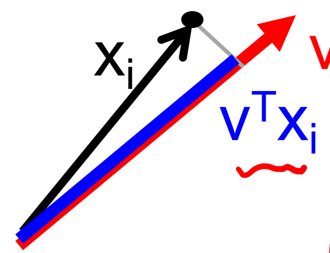
$v_i^T v_i = 1$



Find vector that maximizes sample variance of projection

$1 \times d \quad d \times d \quad d \times 1$

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \frac{\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}}{n}$$



$E[z^2]$

$E[(z-Ez)^2]$

$\mathrm{var}(z)$

$\frac{1}{n}\sum_{i=1}^{n} z_i^2$

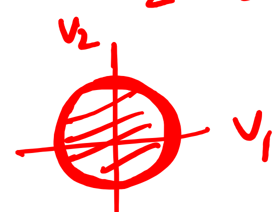
$$\max_{\mathbf{v}} \quad \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} \quad \text{s.t.} \quad \mathbf{v}^T\mathbf{v} = 1$$

convex $\qquad$ $\sum v_i^2 = 1$ $\qquad$ non-convex set

Poll:

$z^T M z \geq 0$

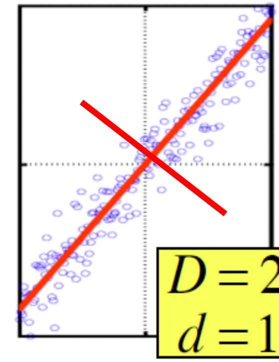
- Is this a convex optimization problem?



$\|v\| \leq 1$

$v^T v \leq 1$

Principal Component Analysis (PCA)



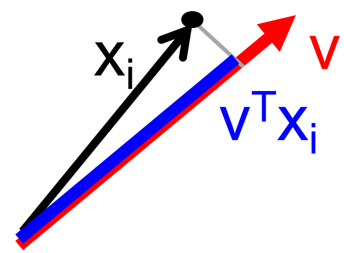
Let $v_1, v_2, \ldots, v_d$ denote the principal components

Orthogonal and unit norm $\quad v_i^T v_j = 0 \quad i \neq j$

$$v_i^T v_i = 1$$

Find vector that maximizes sample variance of projection

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \frac{\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}}{n}$$



$$\max_{\mathbf{v}} \ \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} \quad \text{s.t.} \quad \mathbf{v}^T\mathbf{v} = 1$$

λ

Lagrangian: $\max_{\mathbf{v}} \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} - \lambda\mathbf{v}^T\mathbf{v}$

Wrap constraints into the objective function

$2XX^Tv - 2\lambda v = 0$

evec(XX^T) =

$$\partial/\partial\mathbf{v} = 0 \qquad (\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0 \qquad \Rightarrow \boxed{(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}}$$

eval(XX^T)

→ Sample var when projecting on v

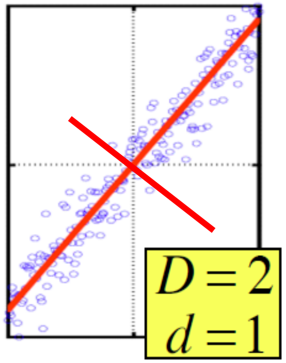
$v^T X X^T v = v^T(\lambda v) = \lambda v^T v = \lambda$

Principal Component Analysis (PCA)

1) center X

2) sample cov $\frac{XX^T}{n}$

$$(\mathbf{X}\mathbf{X}^T)\mathbf{v} = \lambda\mathbf{v}$$



Therefore, v is the eigenvector of sample covariance matrix XX^T

Sample variance of projection $= \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}^T\mathbf{v} = \lambda$

Thus, the eigenvalue λ denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > \ldots$

The 1st Principal component v_1 is the eigenvector of the sample covariance matrix XX^T associated with the largest eigenvalue λ_1

The 2nd Principal component v_2 is the eigenvector of the sample covariance matrix XX^T associated with the second largest eigenvalue λ_2

And so on …

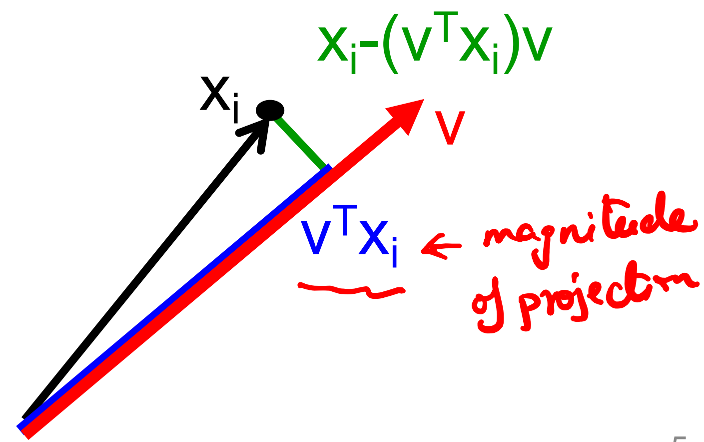
Another interpretation

Maximum Variance Subspace: PCA finds vectors v such that projections on to the vectors capture maximum variance in the data

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

Minimum Reconstruction Error: PCA finds vectors v such that projection on to the vectors yields minimum MSE reconstruction

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \underbrace{(\mathbf{v}^T\mathbf{x}_i)\mathbf{v}}_{\tilde{x}_i}\|^2$$

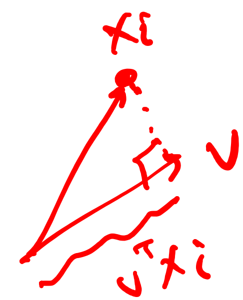


$$\arg\min_{\mathbf{v}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{v}^T\mathbf{x}_i)\mathbf{v}\|^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - (v^T x_i) v)^T (x_i - (v^T x_i) v)$$

$$= \frac{1}{n}\sum_{i=1}^{n} x_i^T x_i - 2(v^T x_i)^2 + \underbrace{v^T (v^T x_i)^2 v}_{(v^T x_i)^2 \cancel{v^T v}} \qquad v^T v \rightarrow 1 = \|v\|_2^2$$

$$= \arg\min_{v} \frac{1}{n}\sum_{i=1}^{n} -(v^T x_i)^2$$

$$= \arg\max_{v} \frac{1}{n}\sum_{i=1}^{n} \underbrace{(v^T x_i)^2}_{\text{projection}}$$

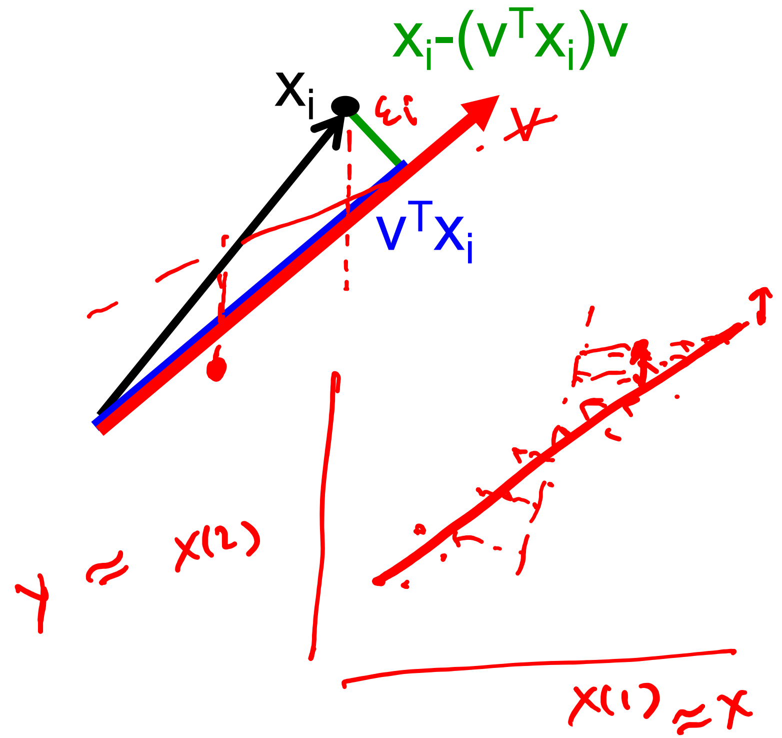


Another interpretation

Minimum Reconstruction Error: PCA finds vectors v such that projection on to the vectors yields minimum MSE reconstruction

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - (\mathbf{v}^T\mathbf{x}_i)\mathbf{v}\|^2$$

$$\frac{1}{n}\sum_{i=1}^{n}\underbrace{(y_i - v^T x_i)}_{\varepsilon_i}{}^2$$



Poll:

➢ Is the 1st PC same as the linear least square fit?

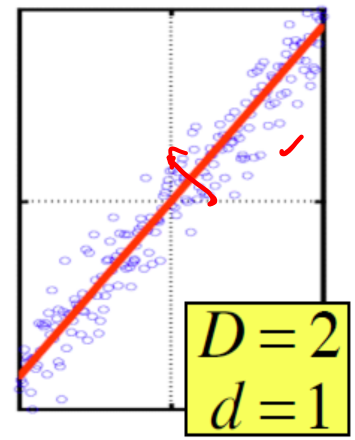
Dimensionality Reduction using PCA

$XX^T_{D \times D} \rightarrow v_1 \ldots v_D$

The eigenvalue λ denotes the amount of variability captured along that dimension.

Zero eigenvalues indicate no variability along those directions => data lies exactly on a linear subspace

Only keep data projections onto principal components with non-zero eigenvalues, say $v_1, \ldots, v_d$ where $d = \text{rank}(XX^T)$



Original Representation

data point

$x_i = [x_i^1, x_i^2, \ldots. x_i^D]^T$

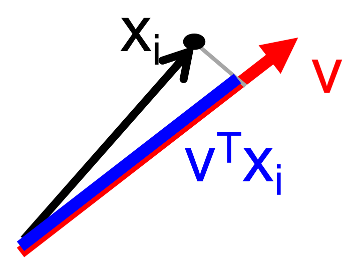
(D-dimensional vector)

Transformed representation

projections

$[v_1^T x_i, v_2^T x_i, \ldots v_d^T x_i]$

(d-dimensional vector)

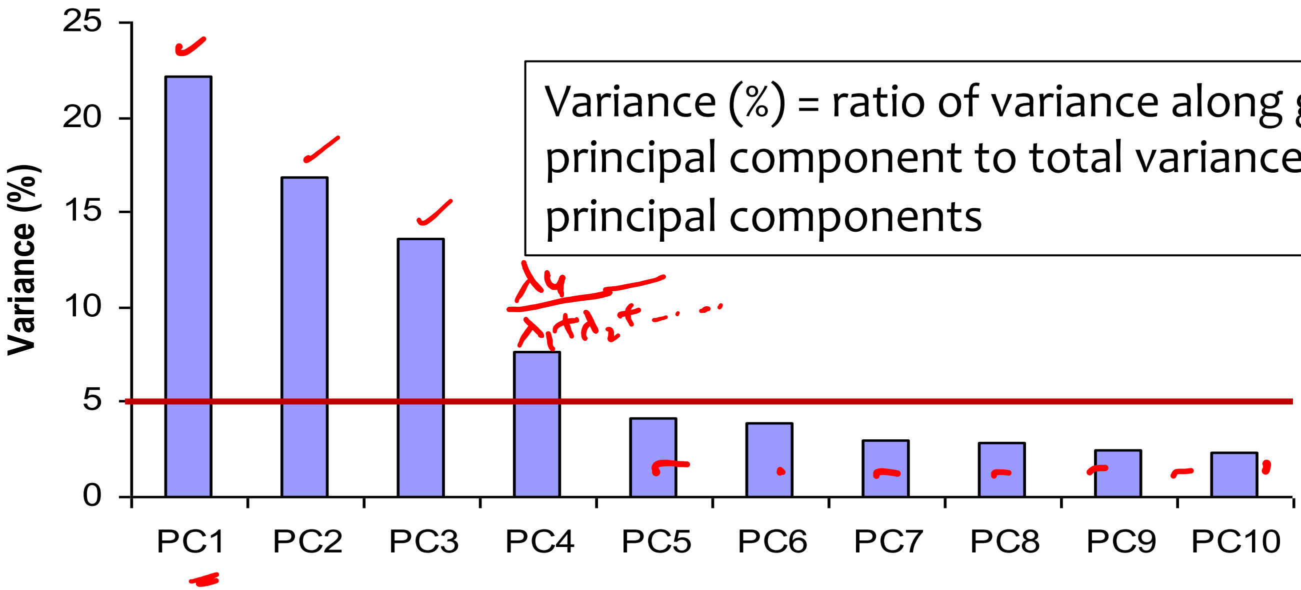


Dimensionality Reduction using PCA

In high-dimensional problem, data usually lies **near** a linear subspace, as noise introduces small variability

Only keep data projections onto principal components with **large** eigenvalues

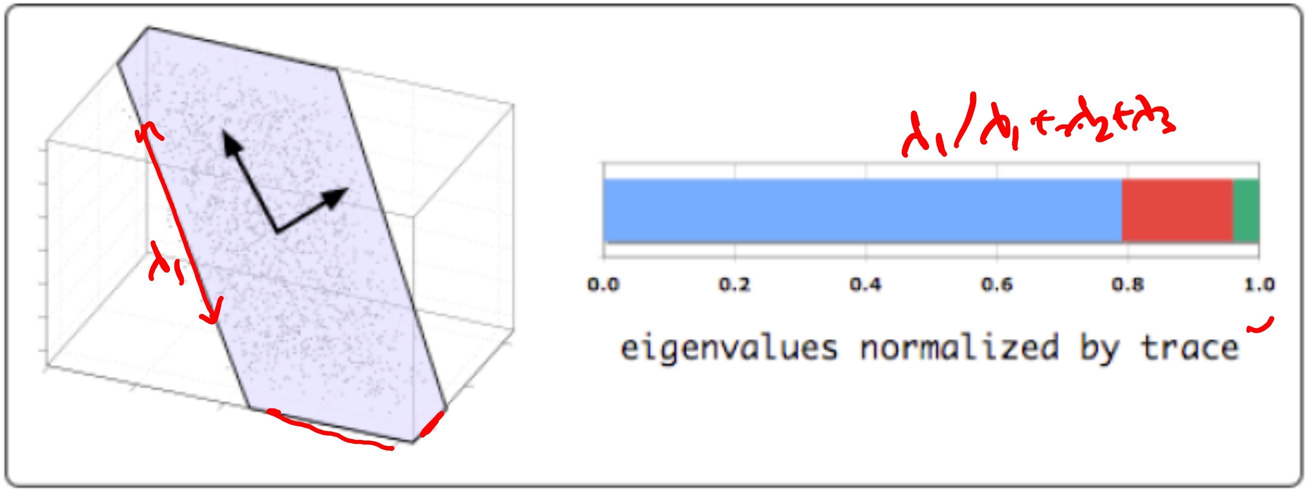
Can *ignore* the components of lesser significance.



Variance (%) = ratio of variance along given principal component to total variance of all principal components

You might lose some information, but if the eigenvalues are small, you don't lose much

Example of PCA



Eigenvectors and eigenvalues of covariance matrix for $n=1600$ inputs in $d=3$ dimensions.

$D = 256 \times 256$

$V_{D \times 1}$

Example: faces

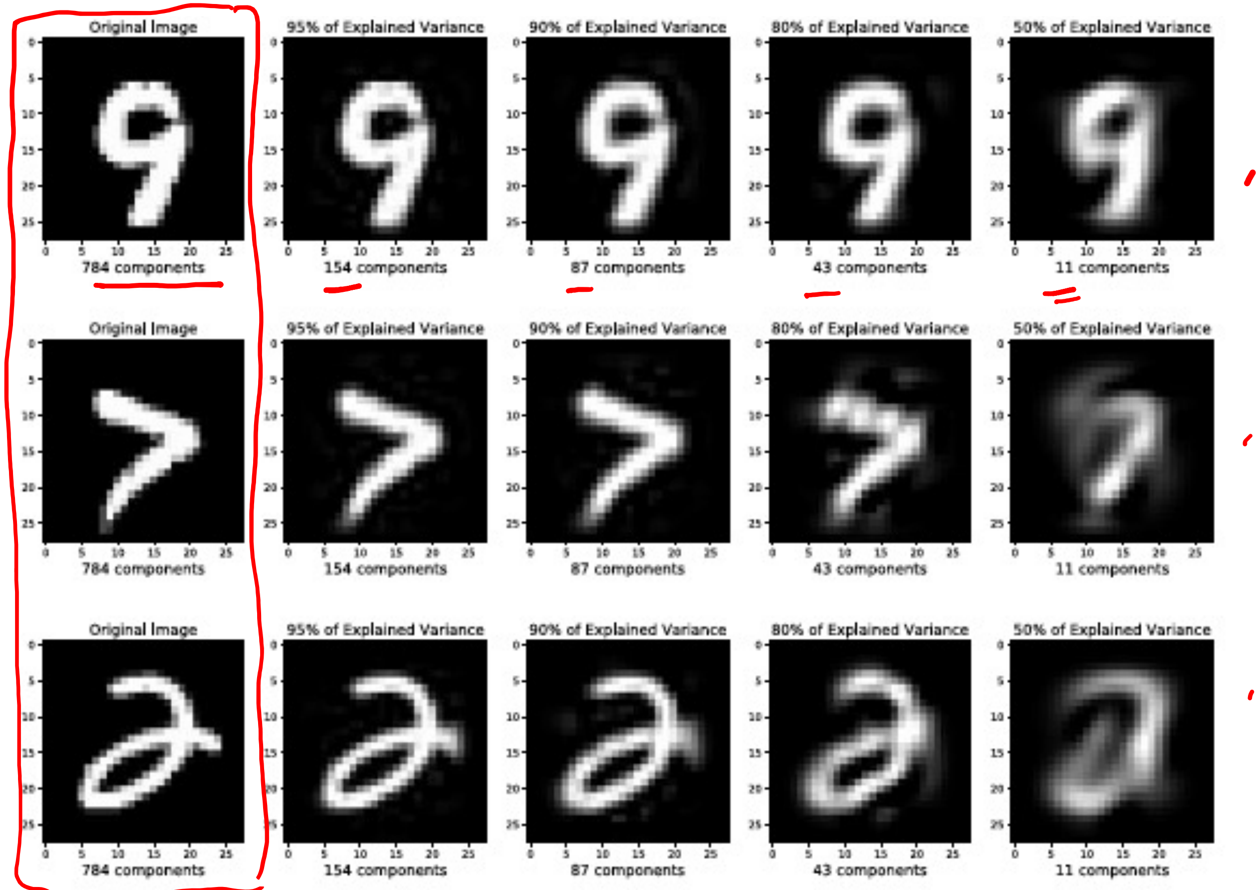
Eigenfaces from 7562 images:

top left image is linear combination of rest.

Sirovich & Kirby (1987)
Turk & Pentland (1991)

Example: MNIST digits

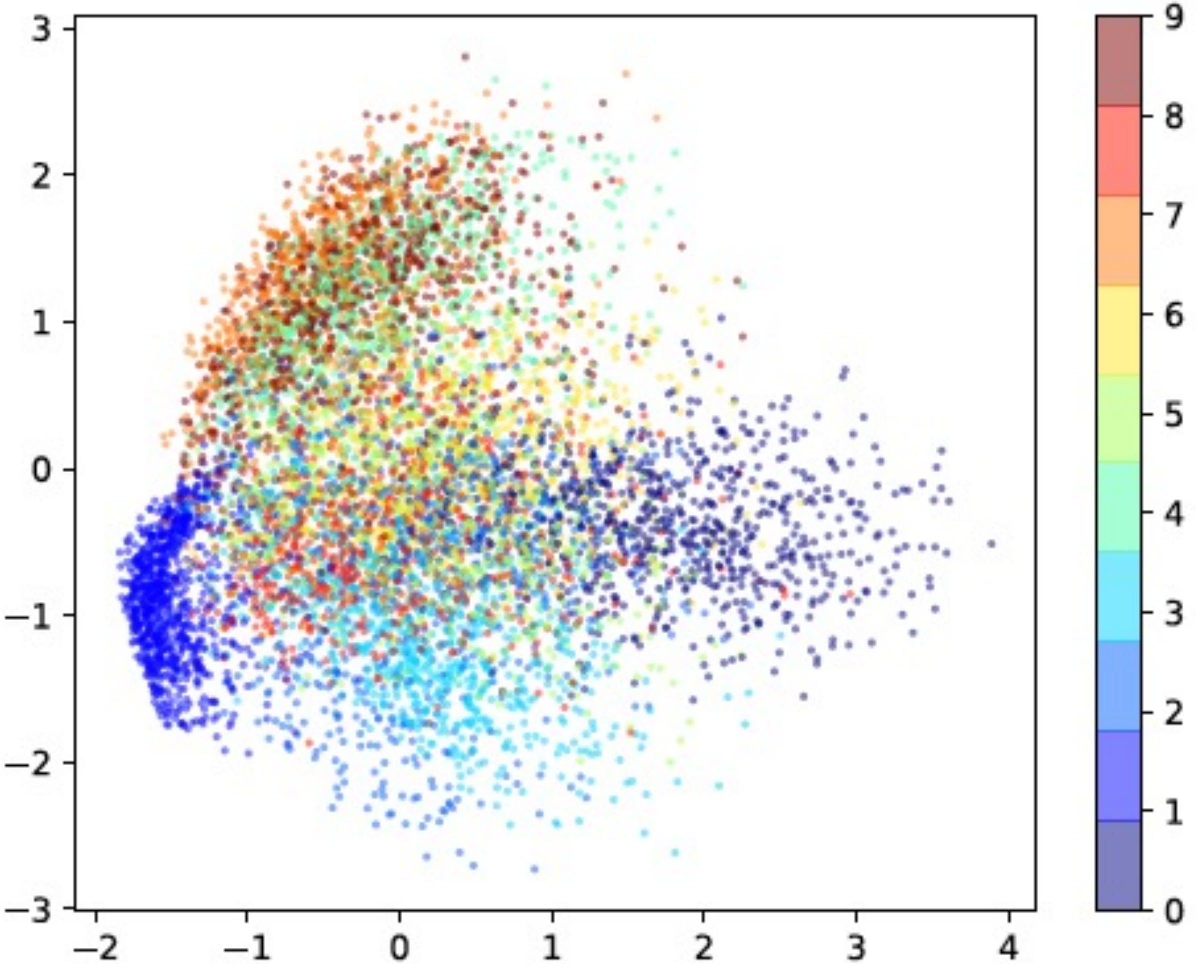
- 28x28 images = 784 PCA vectors
- Project to K dimensional space and then project back up



Projecting MNIST digits

D=28×28

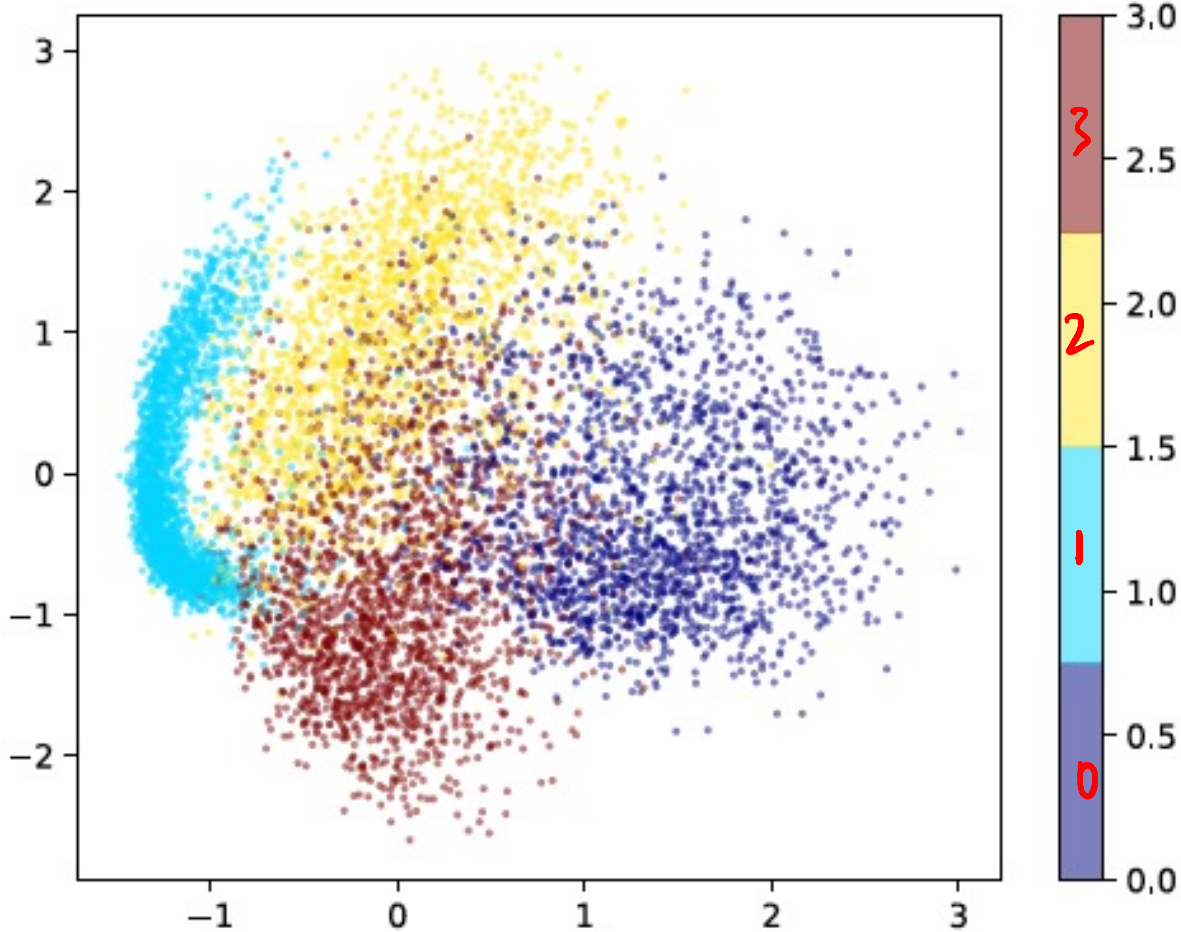
d=2



x(2)

x(1)

Projecting MNIST digits



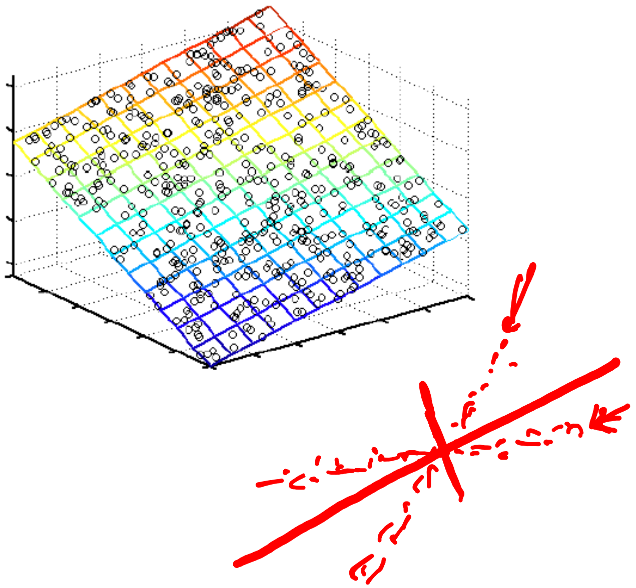
Unsupervised Dimensionality Reduction

- Linear

    Principal Component Analysis (PCA)

    Factor Analysis

    Independent Component Analysis (ICA)



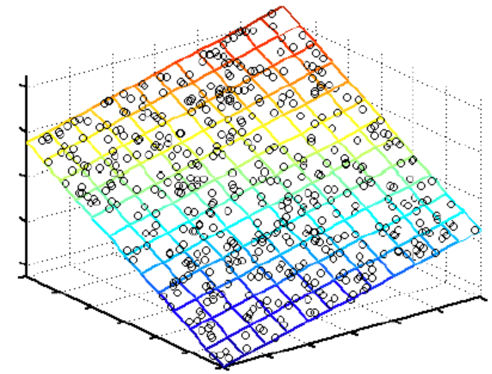
Unsupervised Dimensionality Reduction

- Linear

  Principal Component Analysis (PCA)

  Factor Analysis

  Independent Component Analysis (ICA)

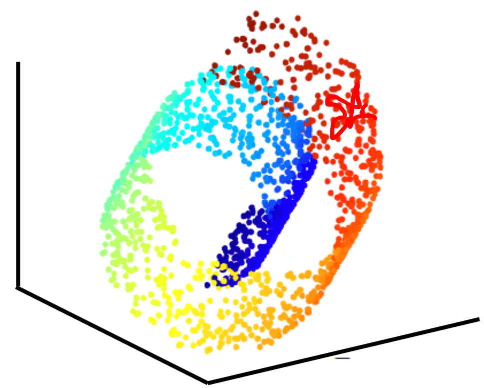


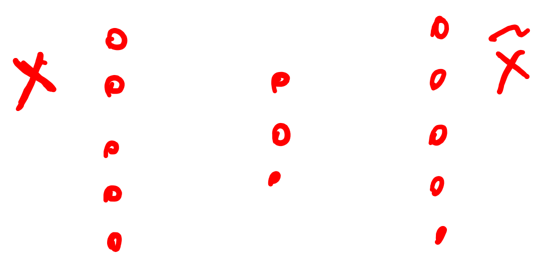
- Nonlinear

  Kernel PCA

  Laplacian Eigenmaps, ISOMAP, LLE

  Autoencoders

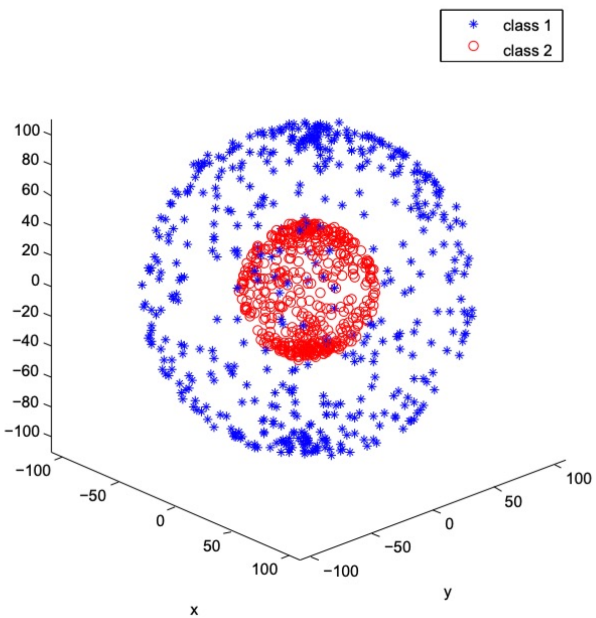




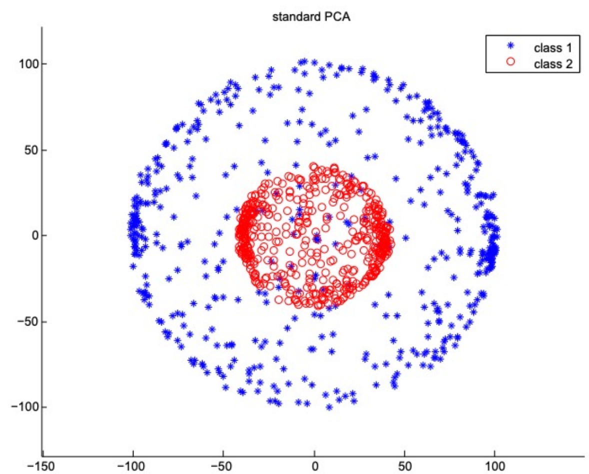
$\to (\tilde{x}-x)^2$

Kernel PCA

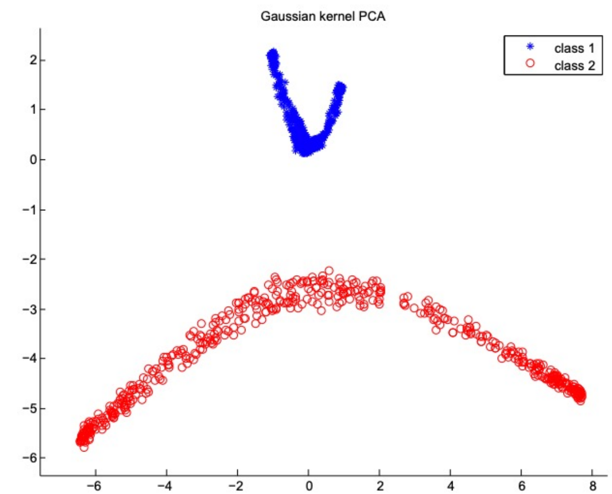
Latent features: linear in $\phi(x)$ where $\phi(x).\ \phi(x') = K(x,x')$ that capture maximum variance or minimum reconstruction error



Original data points



PCA



Kernel PCA
Gaussian/RBF kernel

Src: ArXiv 1207.3538

Kernel PCA

Latent features: linear in $\phi(x)$ where $\phi(x).\ \phi(x') = K(x,x')$ that capture maximum variance or minimum reconstruction error

PCA:

Top d eigenvectors (each D dimensional) of sample covariance $\underbrace{XX^T}_{D \times D}$

Low d-dimensional embedding of a point: $[v_1^T x_i,\ v_2^T x_i,\ \ldots\ v_d^T x_i]$

$d \times 1$

Kernel PCA:

$\phi(x)^T \phi(x)$

Top d eigenvectors (each n dimensional) of kernel matrix $\underbrace{K(X,X)}_{n \times n} = X^T X$

Low d-dimensional embedding of a point: $[v_1(i),\ v_2(i),\ \ldots,\ v_d(i)]$

Eigenvectors are not PCs but projections of data points

PCA Summary

- PCA finds latent features linear in original features x that capture
  - Maximum variance amongst all linear features
  - Minimum reconstruction error when recovering points from PC projections

- Non-convex problem with simple solution:

  PCs = eigenvectors of sample covariance matrix

  Lower ($d < D$) dimensional embedding of data point = projection of data point onto d PCs

- Kernel PCA: latent features linear in $\phi(x)$ where $\phi(x).\ \phi(x') = K(x,x')$ that capture maximum variance or minimum reconstruction error
  - Directly get projections of data points

Clustering

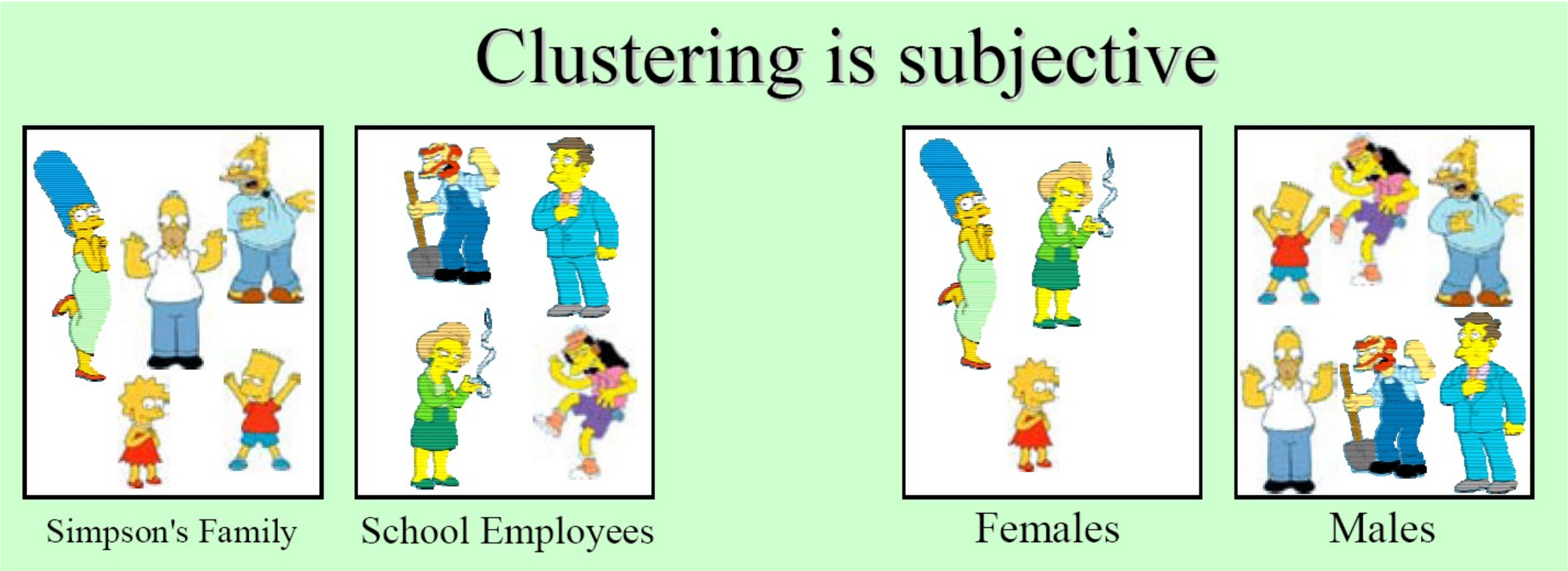
Aarti Singh

Machine Learning 10-701
Apr 24, 2023

Some slides courtesy of Eric Xing, Carlos Guestrin

What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - high intra-class similarity
  - low inter-class similarity
  - It is the most common form of unsupervised learning



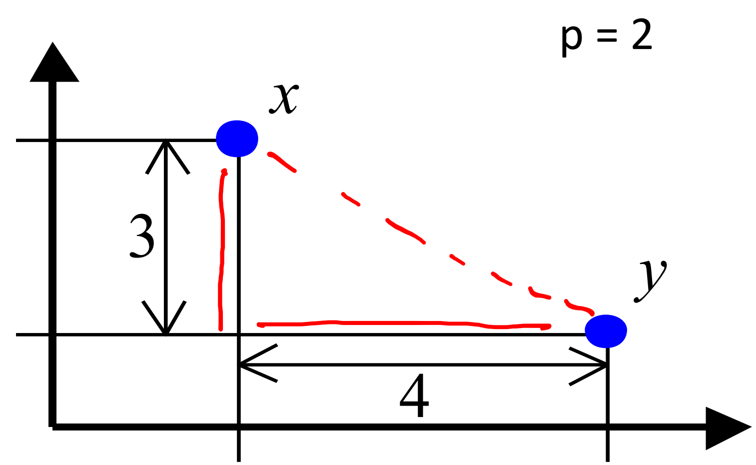
What is Similarity?

Hard to define! But *we know it when we see it*

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach - think in terms of a distance (rather than similarity) between vectors or correlations between random variables.

Distance metrics

$x = (x_1, x_2, \ldots, x_p)$
$y = (y_1, y_2, \ldots, y_p)$



Euclidean distance $$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$ **5**

Manhattan distance $$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$ **7**

Sup-distance $$d(x,y) = \max_{1 \le i \le p} |x_i - y_i|$$ **4**

Correlation coefficient

$x = (x_1, x_2, \ldots, x_p)$
$y = (y_1, y_2, \ldots, y_p)$

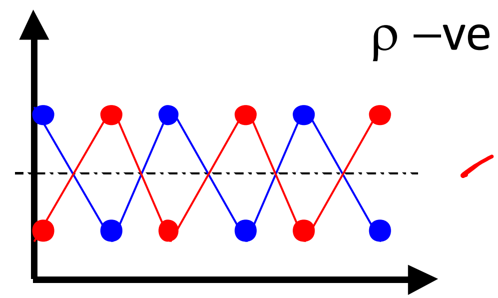
Random vectors (e.g. expression levels of two genes under various drugs)

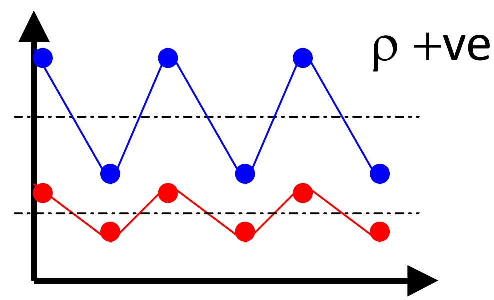
$I(X,Y)$

Pearson correlation coefficient

$$\rho(x,y) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \times \sum_{i=1}^{p}(y_i - \bar{y})^2}}$$

$$\text{where } \bar{x} = \frac{1}{p}\sum_{i=1}^{p} x_i \text{ and } \bar{y} = \frac{1}{p}\sum_{i=1}^{p} y_i.$$

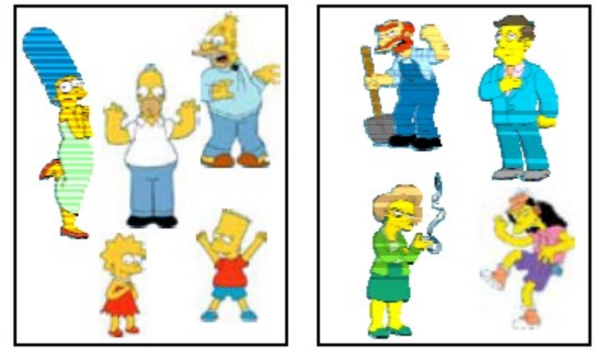




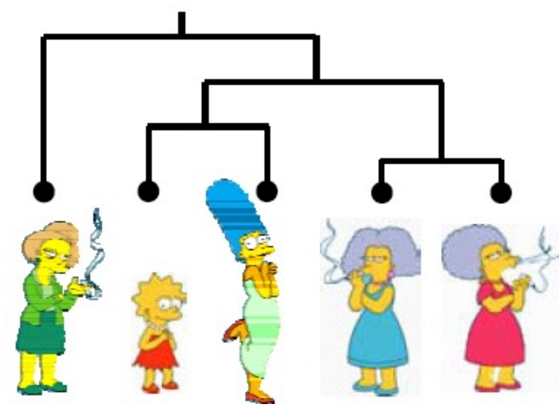
Clustering Algorithms

- **Partition algorithms**
  - K means clustering
  - Mixture-Model based clustering

p(x)



- Hierarchical algorithms
  - Single-linkage
  - Average-linkage
  - Complete-linkage
  - Centroid-based



Partitioning Algorithms

- Partitioning method: Construct a partition of *n* objects into a set of *K* clusters
- Given: a set of objects and the number *K*

$$\binom{n}{K} \simeq n^K$$

- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic method: K-means algorithm

K-Means

Algorithm

Input – Desired number of clusters, *k*

Initialize – the *k* cluster centers (randomly if necessary)

Iterate –

1. Assign points to the nearest cluster centers

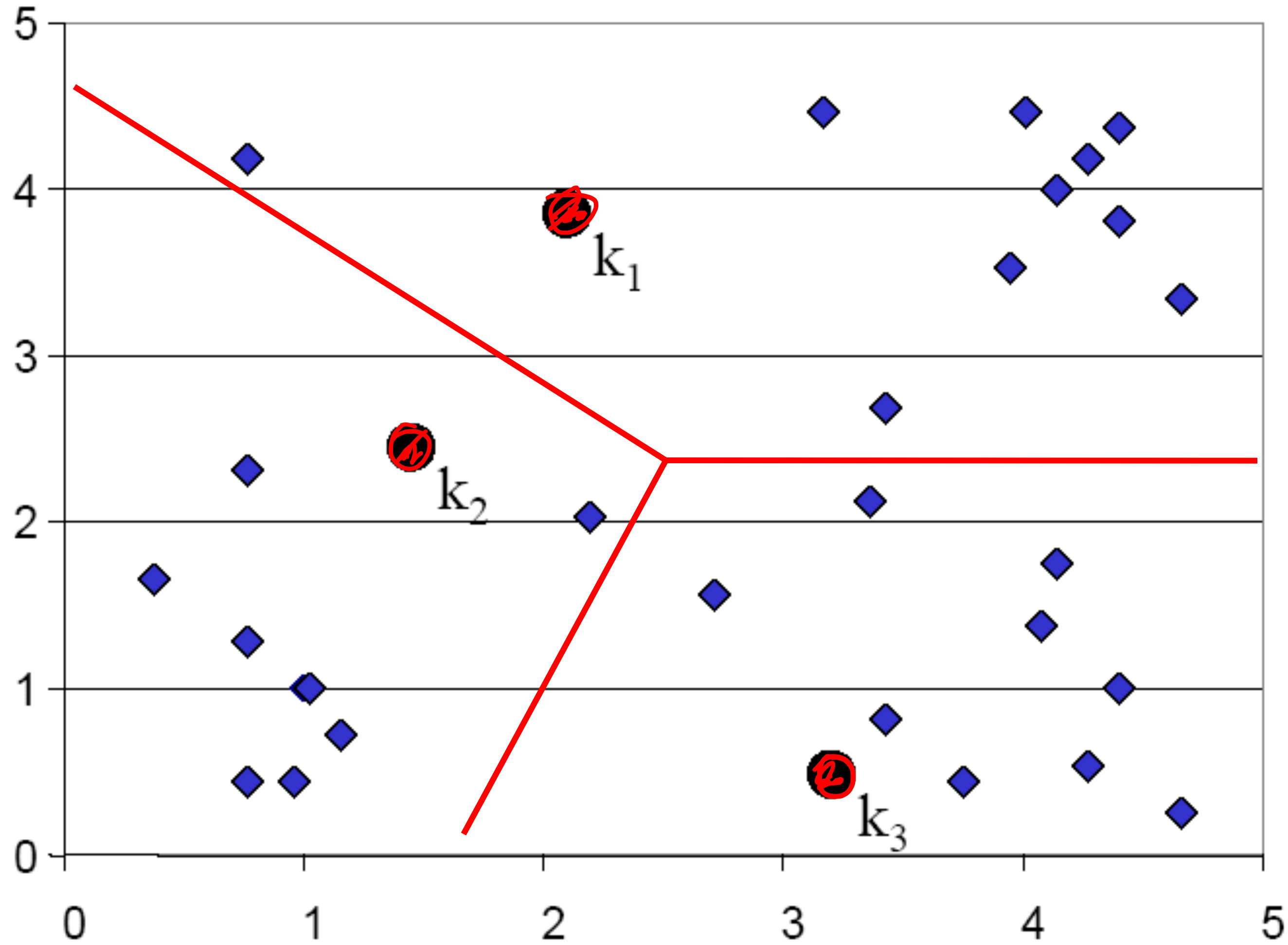
2. Re-estimate the *k* cluster centers (aka the centroid or mean), by assuming the memberships found above are correct.

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

Termination –

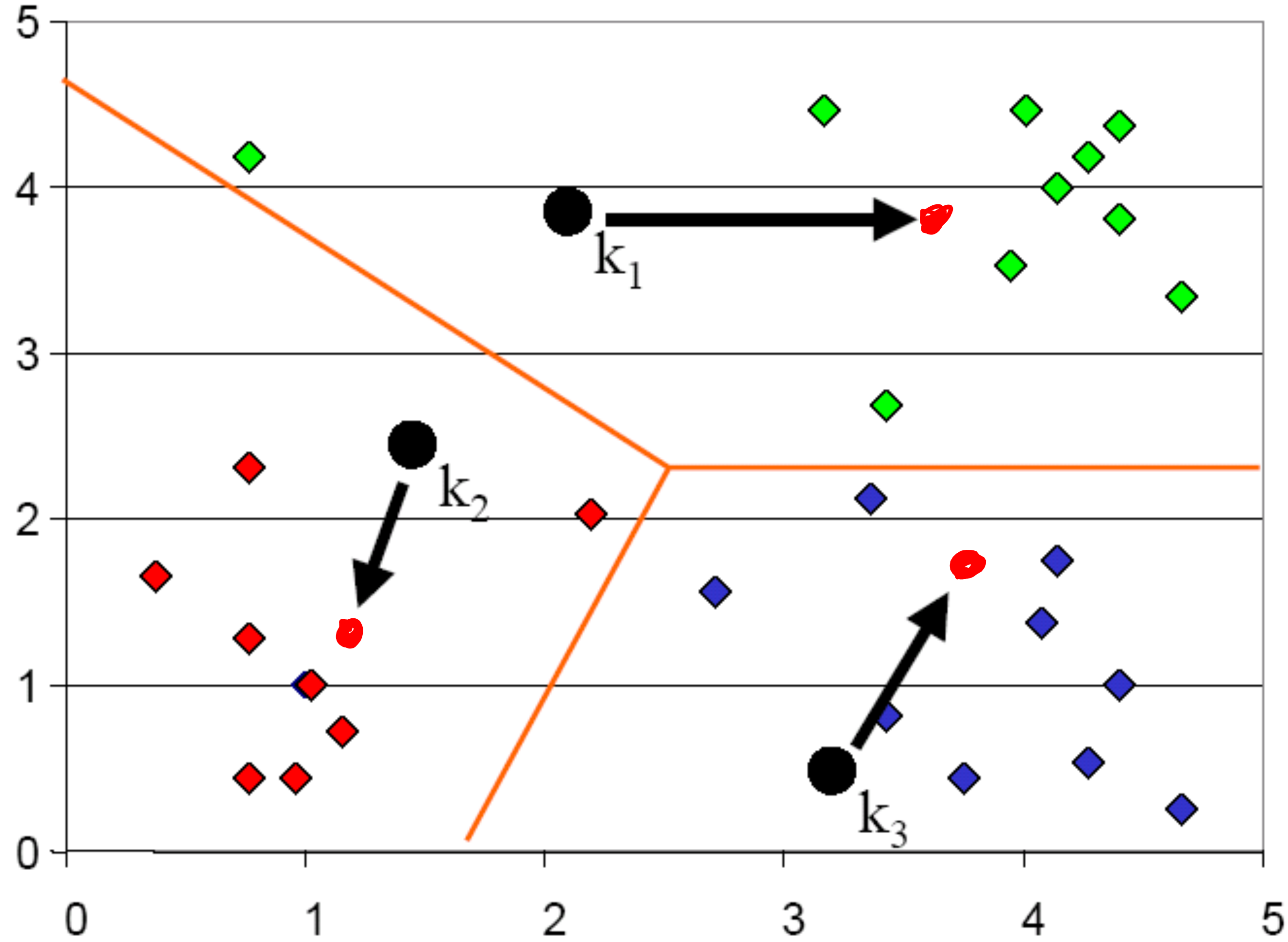
If none of the objects changed membership in the last iteration, exit. Otherwise go to 1.

K-means Clustering: Step 1

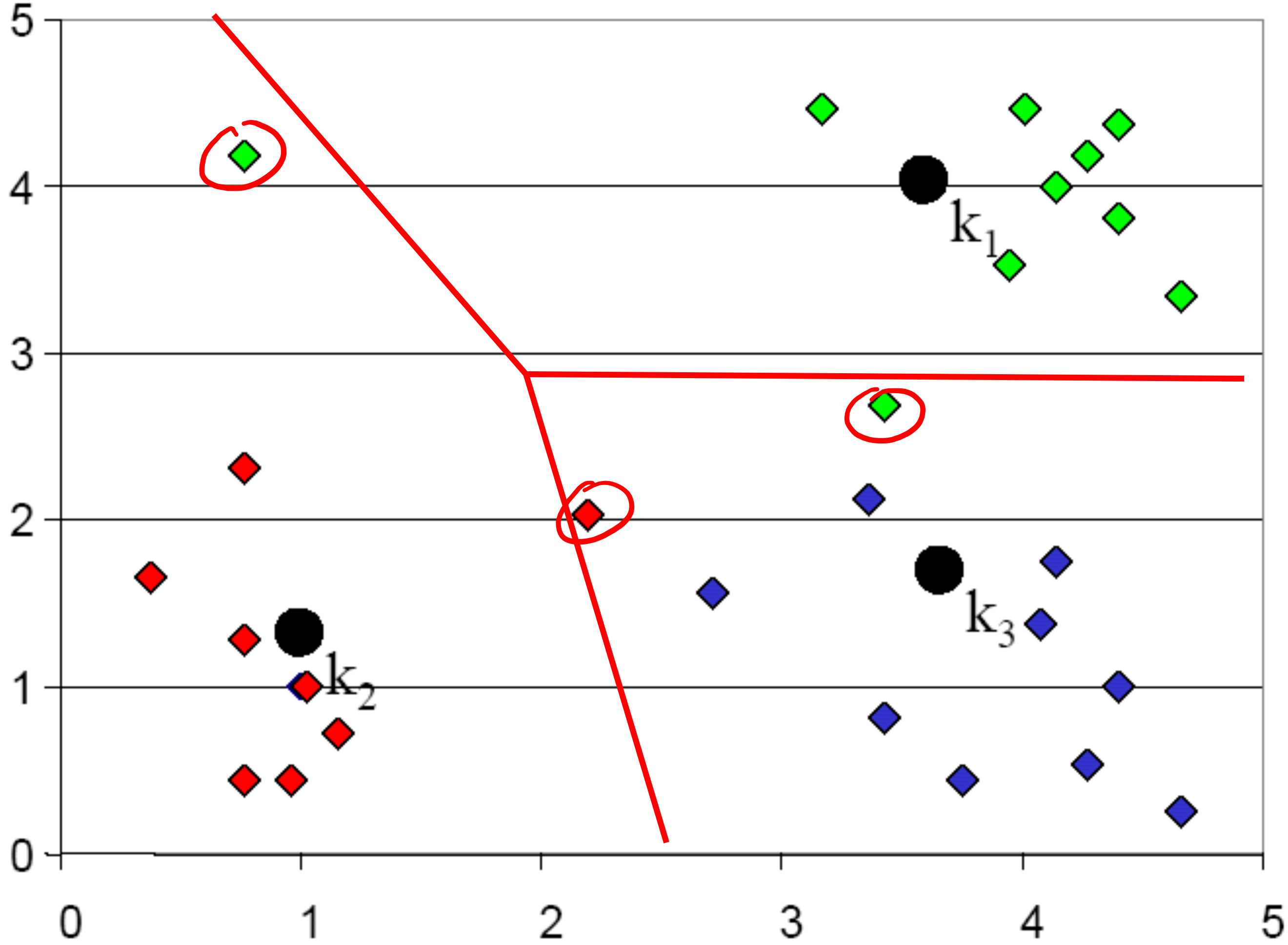


Voronoi diagram

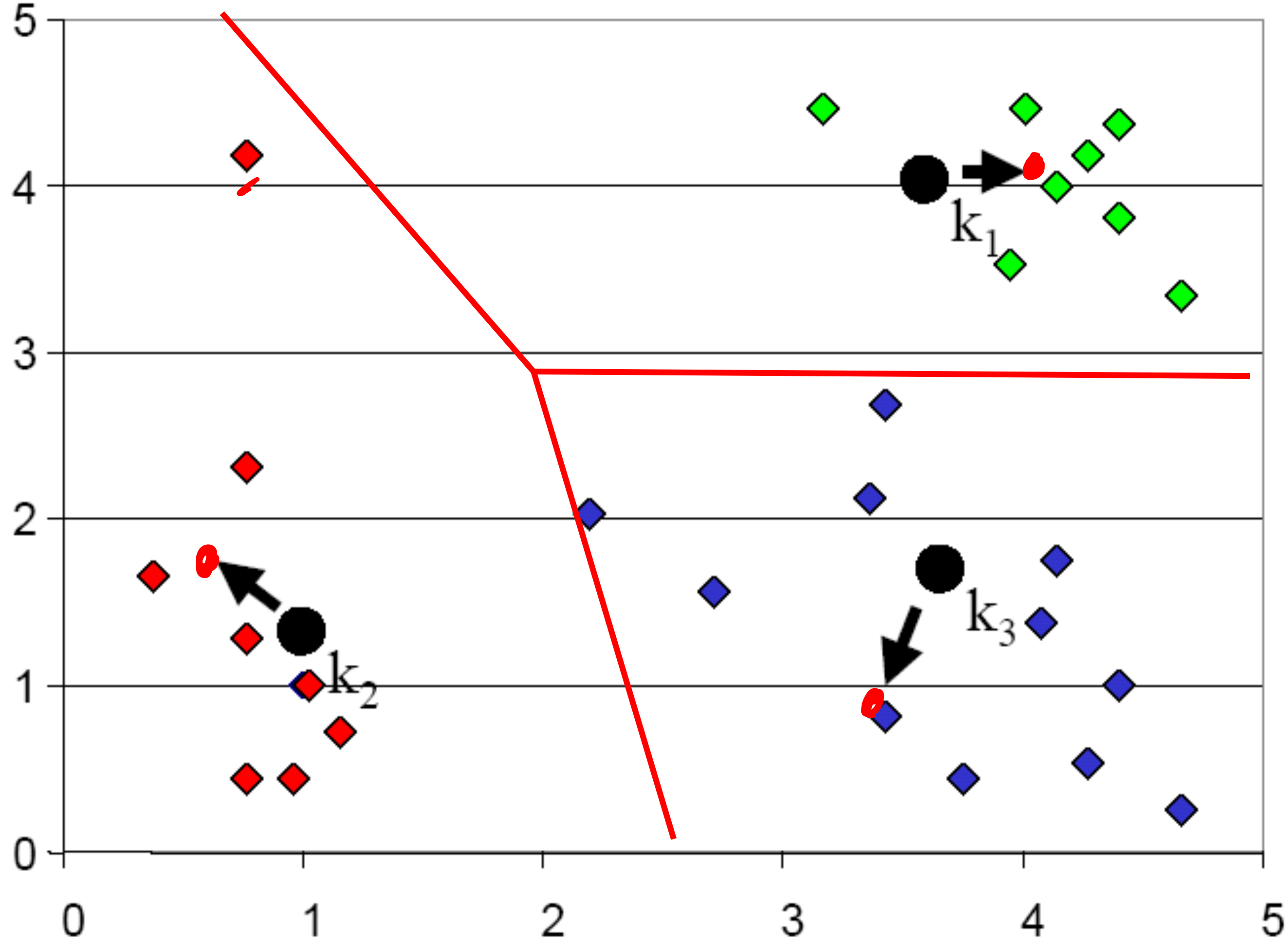
K-means Clustering: Step 2



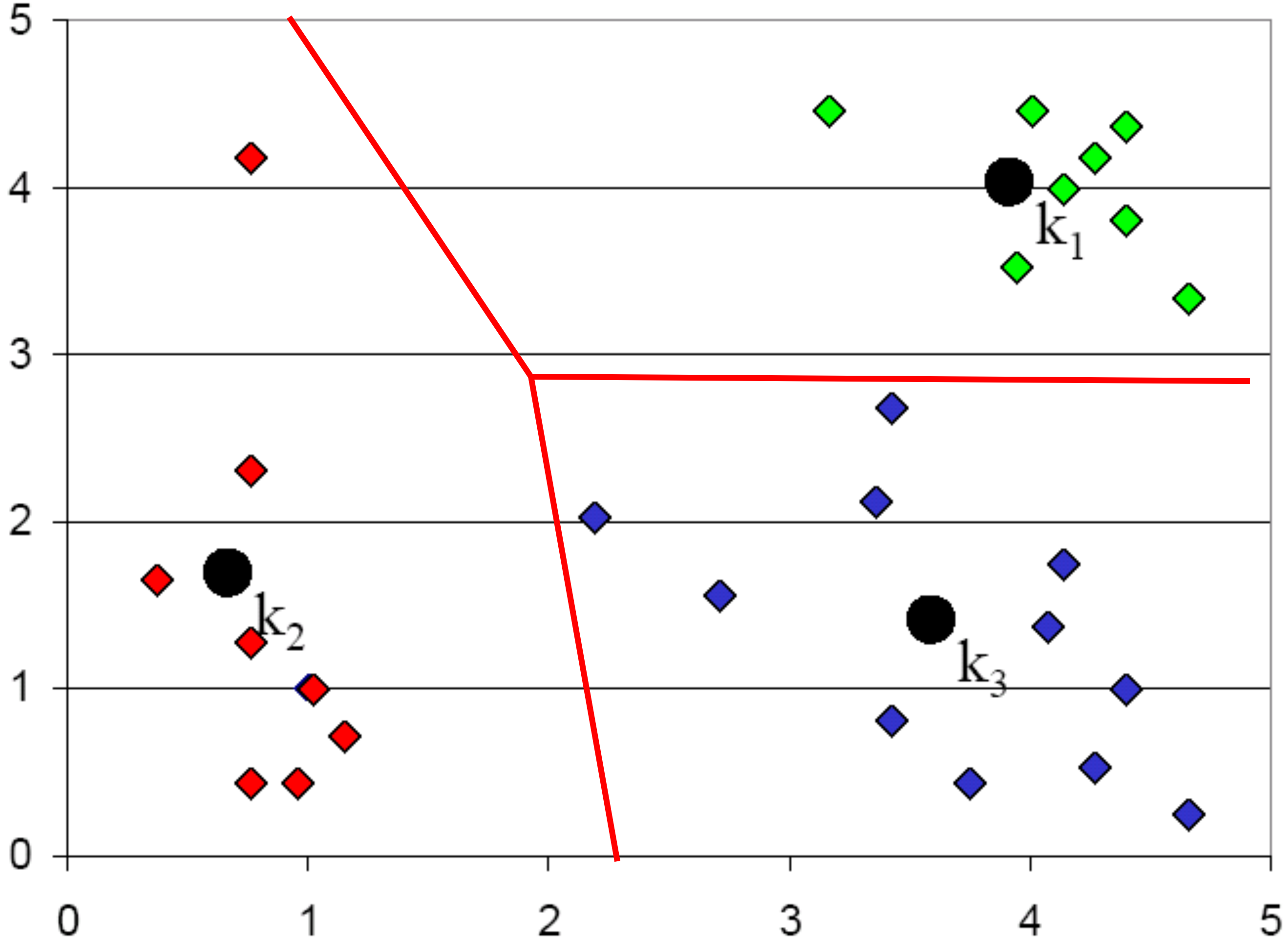
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



K-means Recap …

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

K-means Recap …

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

Iterate t = 0, 1, 2, …

- **Classify**: Assign each point j∈{1,…m} to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_{i=1,\ldots,k} \|\mu_i^{(t)} - x_j\|^2$

K-means Recap …

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

Iterate t = 0, 1, 2, …

- **Classify**: Assign each point j∈{1,…m} to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_{i=1,\ldots,k} \|\mu_i^{(t)} - x_j\|^2$
- **Recenter**: μ_i becomes centroid of its points**:**
  - $\mu_i^{(t+1)} \leftarrow \arg\min_{\mu} \sum_{j:C^{(t)}(j)=i} \|\mu - x_j\|^2 \qquad i \in \{1, \ldots, k\}$
  - Equivalent to $\mu_i \leftarrow$ average of its points!

What is K-means optimizing?

- Potential function F(**μ**,C) of centers **μ** and point allocations C:

$$F(\mu, C) = \sum_{j=1}^{m} ||\mu_{C(j)} - x_j||^2$$

$$= \sum_{i=1}^{k} \boxed{\sum_{j:C(j)=i} ||\mu_i - x_j||^2} \quad C(j)=i$$

j

- Optimal K-means:
  - $\min_{\mu} \min_{C} F(\mu, C)$

➢ Is the K-means objective convex?

K-means algorithm

- Optimize potential function:

$$\min_{\mu}\min_{C} F(\mu, C) = \min_{\mu}\min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:** (coordinate descent on F)

(1) Fix μ, optimize C — **Expected** cluster assignment

(2) Fix C, optimize μ — **Maximum** likelihood for center

Similar to EM/Baum Welch algorithm for learning HMM parameters