

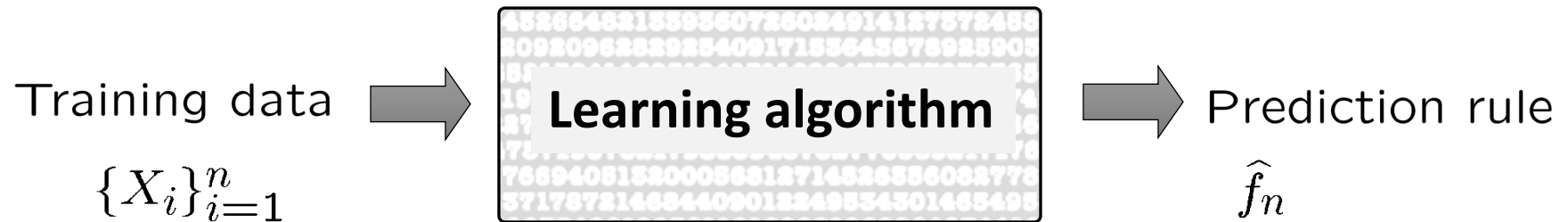
Unsupervised Learning

Prof. Pradeep Ravikumar

pradeepr@cs.cmu.edu

Unsupervised Learning

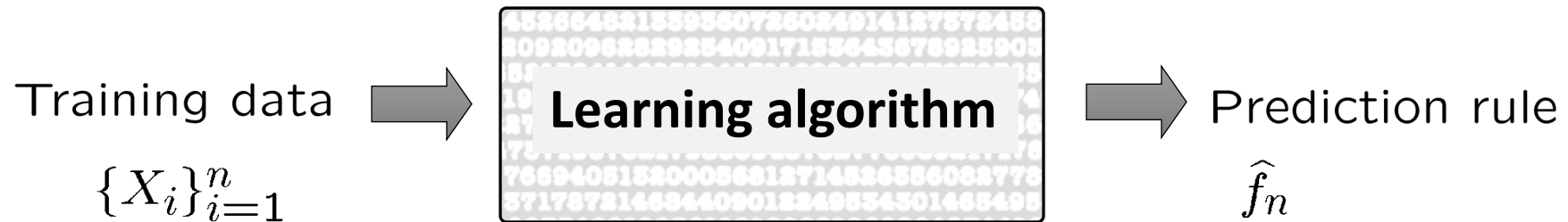
Learning from unlabeled/unannotated data (without supervision)



What can we predict from unlabeled data?

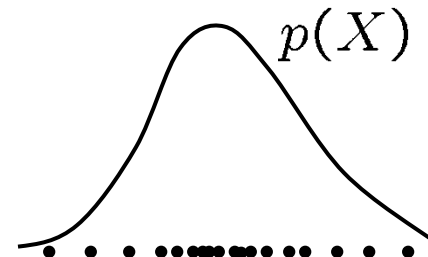
Unsupervised Learning

Learning from unlabeled/unannotated data (without supervision)



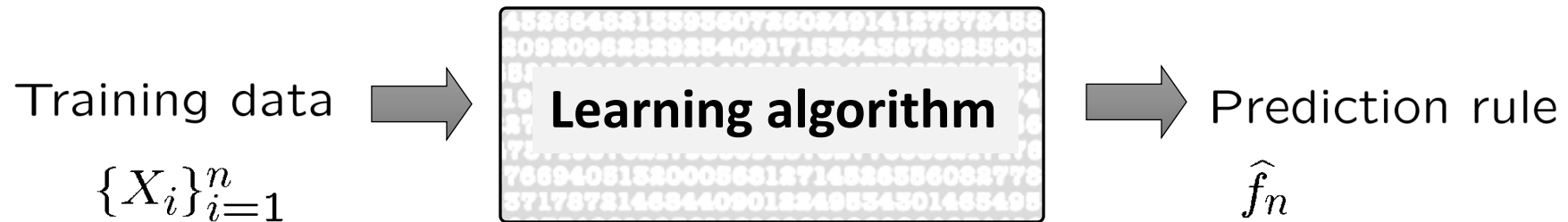
What can we predict from unlabeled data?

- Density estimation



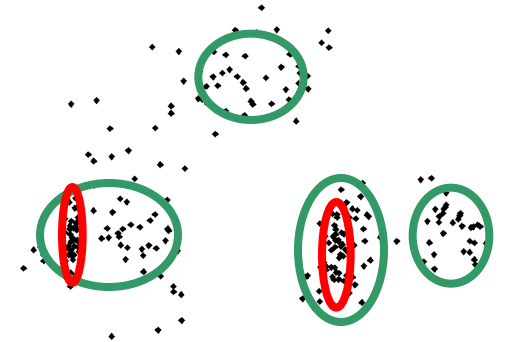
Unsupervised Learning

“Learning from unlabeled/unannotated data” (without supervision)



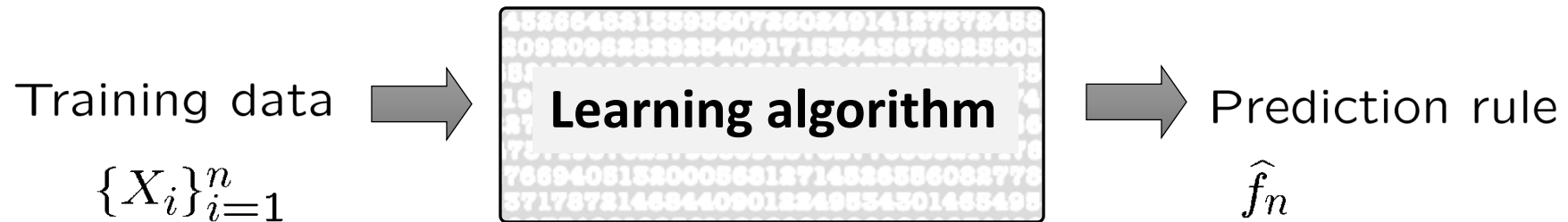
What can we predict from unlabeled data?

- Density estimation
- Groups or clusters in the data



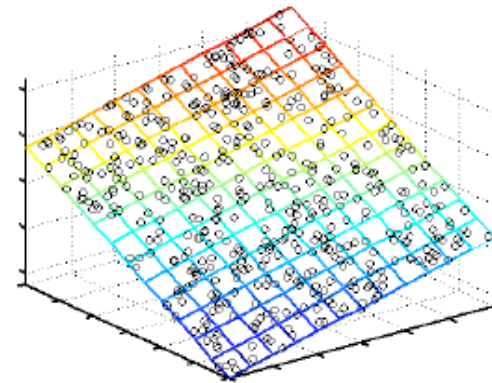
Unsupervised Learning

“Learning from unlabeled/unannotated data” (without supervision)



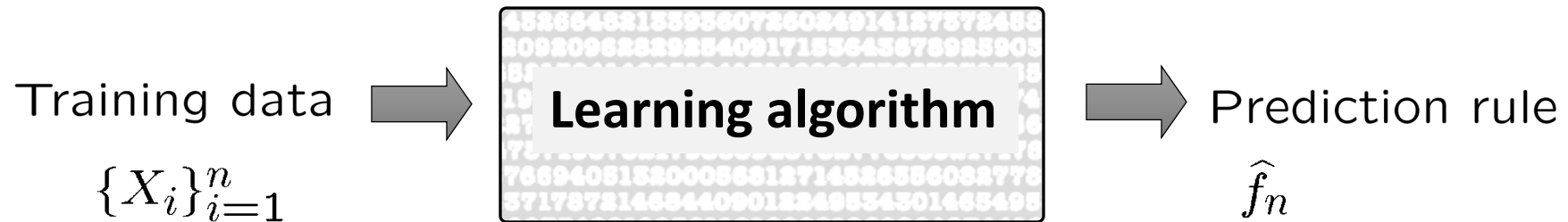
What can we predict from unlabeled data?

- Density estimation
- Groups or clusters in the data
- Low-dimensional structure
 - Principal Component Analysis (PCA) (linear)



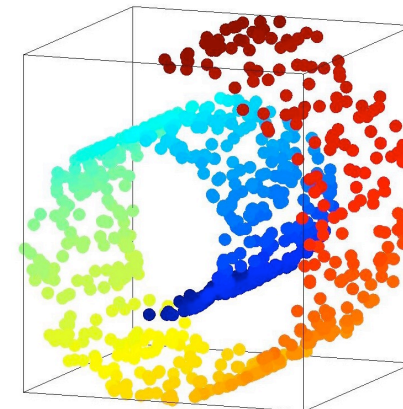
Unsupervised Learning

“Learning from unlabeled/unannotated data” (without supervision)



What can we predict from unlabeled data?

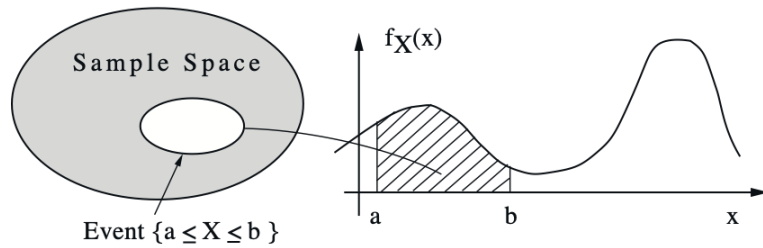
- Density estimation
- Groups or clusters in the data
- Low-dimensional structure
 - Principal Component Analysis (PCA) (linear)
 - Manifold learning (non-linear)



Density Estimation

Continuous Random Variables and PDFs

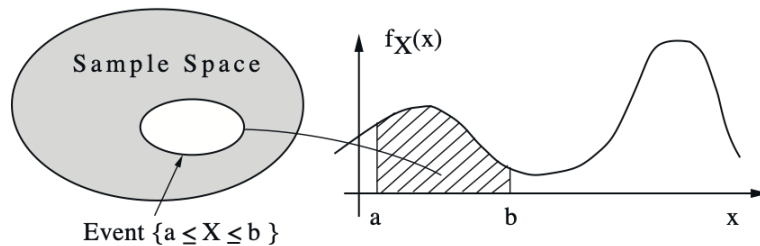
- **A continuous r.v.** is described by a **probability density function f_X**



$$\mathbf{P}(a \leq X \leq b) = \int_a^b f_X(x) dx$$

Continuous Random Variables and PDFs

- **A continuous r.v.** is described by a **probability density function f_X**

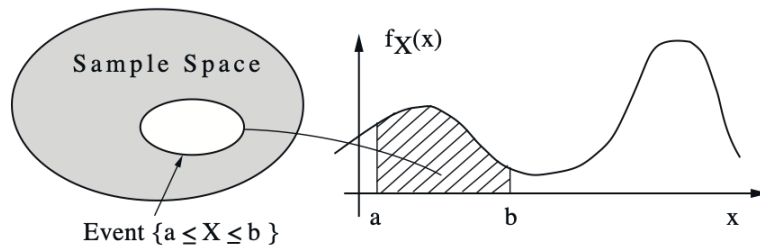


$$\mathbf{P}(a \leq X \leq b) = \int_a^b f_X(x) dx$$

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad \dots \text{ must satisfy the normalization equation}$$

Continuous Random Variables and PDFs

- A continuous r.v. is described by a probability density function f_X

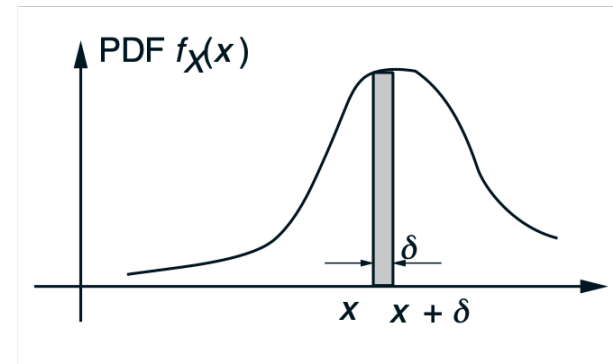


$$\mathbf{P}(a \leq X \leq b) = \int_a^b f_X(x) dx$$

$$\int_{-\infty}^{\infty} f_X(x) dx = 1$$

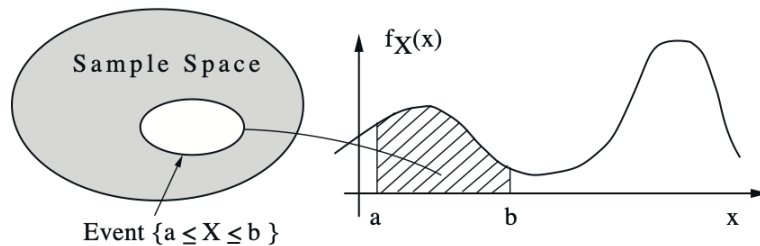
$$\mathbf{P}(x \leq X \leq x + \delta) = \int_x^{x+\delta} f_X(s) ds \approx f_X(x) \cdot \delta$$

$f_X(x)$ as “probability mass per unit length”



Continuous Random Variables and PDFs

- **A continuous r.v.** is described by a **probability density function** f_X



$$\mathbf{P}(a \leq X \leq b) = \int_a^b f_X(x) dx$$

$$\int_{-\infty}^{\infty} f_X(x) dx = 1$$

$$\mathbf{P}(x \leq X \leq x + \delta) = \int_x^{x+\delta} f_X(s) ds \approx f_X(x) \cdot \delta$$

$$\mathbf{P}(X \in B) = \int_B f_X(x) dx, \quad \text{for "nice" sets } B \quad \dots \text{ area under the curve}$$

Example

- A gambler spins a wheel of fortune, continuously calibrated between 0 and 1, and observes the resulting number. Assume that all subintervals of $[0,1]$ of the same length are equally likely. This experiment can be modeled in terms a random variable X with PDF

$$f_X(x) = \begin{cases} c & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$c =$$

Calculate the value of c from the normalization equation for the PDF

Example

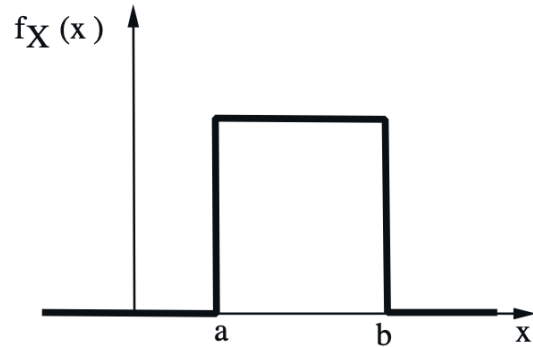
- A gambler spins a wheel of fortune, continuously calibrated between 0 and 1, and observes the resulting number. Assuming that all subintervals of $[0,1]$ of the same length are equally likely. This experiment can be modeled in terms a random variable X with PDF

$$f_X(x) = \begin{cases} c & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$c = 1$$

$$1 = \int_{-\infty}^{\infty} f_X(x) dx = \int_0^1 c dx = c \int_0^1 dx = c$$

Continuous Uniform RV



$$f_X(x) = \begin{cases} c & \text{if } a \leq x \leq b, \\ 0 & \text{otherwise,} \end{cases}$$

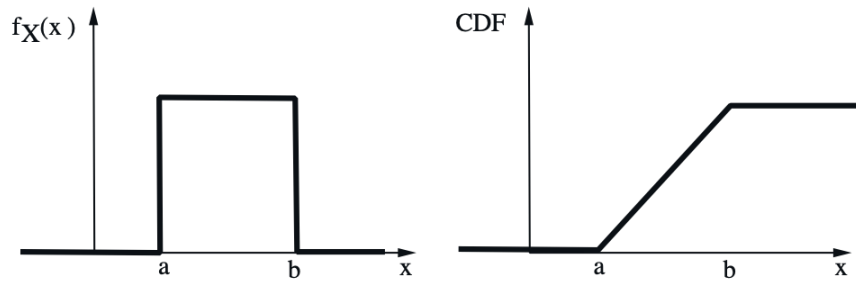
- In previous example, X took values in $[0,1]$, with all lengths equally likely.
- This is a generalization, with X taking values in an interval $[a,b]$
- Verify: $c = 1/(b-a)$

Cumulative Distribution Function

$$F_X(x) = \mathbf{P}(X \leq x) = \int_{-\infty}^x f_X(t) dt$$

Cumulative Distribution Function

$$F_X(x) = \mathbf{P}(X \leq x) = \int_{-\infty}^x f_X(t) dt$$



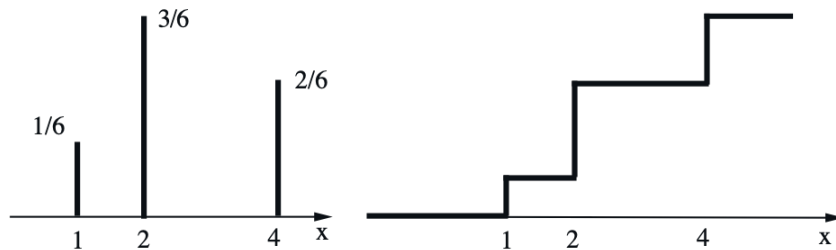
Uniform RV

Cumulative Distribution Function

$$F_X(x) = \mathbf{P}(X \leq x) = \int_{-\infty}^x f_X(t) dt$$

Discrete RV

$$F_X(x) = \mathbf{P}(X \leq x) = \sum_{k \leq x} p_X(k)$$



PDFs and CDFs

- If X is discrete and takes integer values, the PMF and the CDF are connected as follows:

$$F_X(k) = \sum_{i=-\infty}^k p_X(i),$$

PDFs and CDFs

- If X is discrete and takes integer values, the PMF and the CDF are connected as follows:

$$F_X(k) = \sum_{i=-\infty}^k p_X(i),$$

$$p_X(k) = \mathbf{P}(X \leq k) - \mathbf{P}(X \leq k - 1) = F_X(k) - F_X(k - 1),$$

PDFs and CDFs

- If X is continuous, the PDF and the CDF can be obtained from each other by

$$F_X(x) = \int_{-\infty}^x f_X(t) dt,$$

PDFs and CDFs

- If X is continuous, the PDF and the CDF can be obtained from each other by

$$F_X(x) = \int_{-\infty}^x f_X(t) dt,$$

$$f_X(x) = \frac{dF_X}{dx}(x).$$

Example

- You are allowed to take a certain test three times, and your final score will be the maximum of the test scores. Thus, $X = \max\{X_1, X_2, X_3\}$, where X_1, X_2, X_3 are the three test scores and X is the final score. Assume that your score in each test takes one of the values from 1 to 10 with equal probability $1/10$, independently of the scores in other tests. What is the PMF p_X of the final score?

Example

- You are allowed to take a certain test three times, and your final score will be the maximum of the test scores. Thus, $X = \max\{X_1, X_2, X_3\}$, where X_1, X_2, X_3 are the three test scores and X is the final score. Assume that your score in each test takes one of the values from 1 to 10 with equal probability $1/10$, independently of the scores in other tests. What is the PMF p_X of the final score?
 - Hint: Compute the CDF first

Example

- You are allowed to take a certain test three times, and your final score will be the maximum of the test scores. Thus, $X = \max\{X_1, X_2, X_3\}$, where X_1, X_2, X_3 are the three test scores and X is the final score. Assume that your score in each test takes one of the values from 1 to 10 with equal probability $1/10$, independently of the scores in other tests. What is the PMF p_X of the final score?
 - Hint: Compute the CDF first

$$\begin{aligned} F_X(k) &= \mathbf{P}(X \leq k) \\ &= \mathbf{P}(X_1 \leq k, X_2 \leq k, X_3 \leq k) \\ &= \mathbf{P}(X_1 \leq k) \mathbf{P}(X_2 \leq k) \mathbf{P}(X_3 \leq k) \\ &= \left(\frac{k}{10}\right)^3, \end{aligned}$$

Example

- You are allowed to take a certain test three times, and your final score will be the maximum of the test scores. Thus, $X = \max\{X_1, X_2, X_3\}$, where X_1, X_2, X_3 are the three test scores and X is the final score. Assume that your score in each test takes one of the values from 1 to 10 with equal probability $1/10$, independently of the scores in other tests. What is the PMF p_X of the final score?
 - Hint: Compute the CDF first

$$\begin{aligned} F_X(k) &= \mathbf{P}(X \leq k) \\ &= \mathbf{P}(X_1 \leq k, X_2 \leq k, X_3 \leq k) \\ &= \mathbf{P}(X_1 \leq k) \mathbf{P}(X_2 \leq k) \mathbf{P}(X_3 \leq k) \\ &= \left(\frac{k}{10}\right)^3, \end{aligned}$$

$$p_X(k) = \left(\frac{k}{10}\right)^3 - \left(\frac{k-1}{10}\right)^3, \quad k = 1, \dots, 10.$$

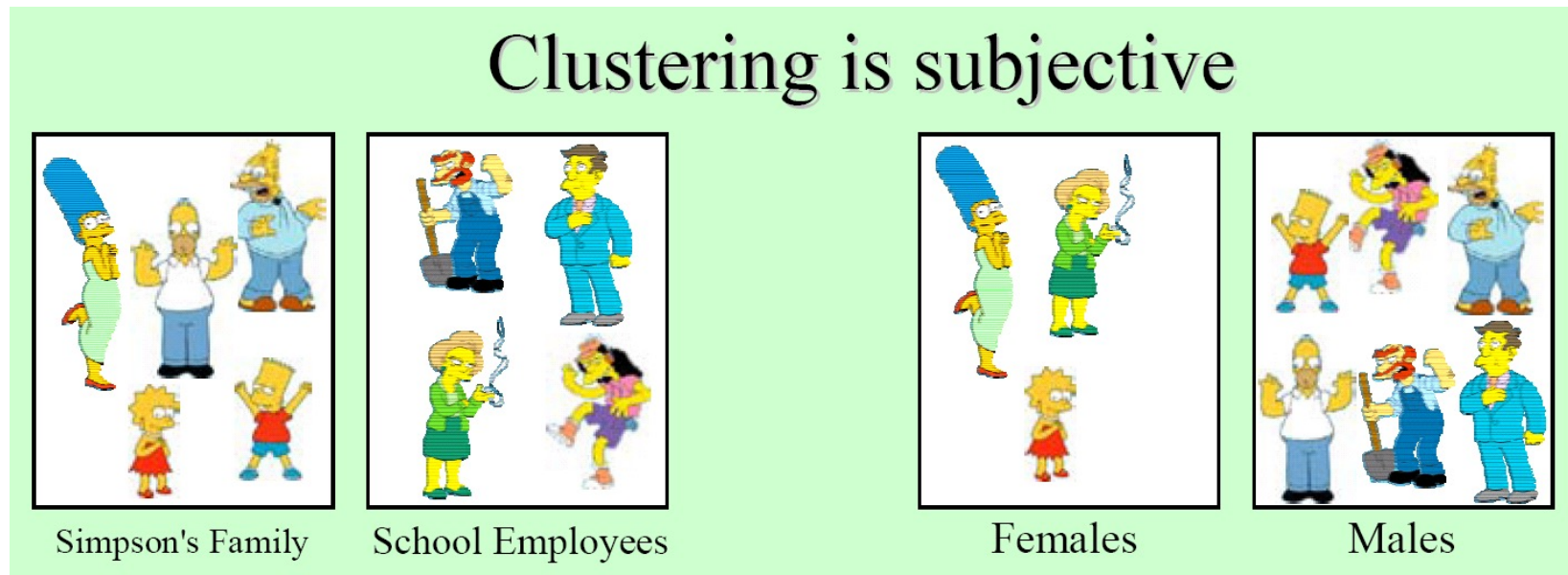
Density Estimation

- Use Frequentist Data Analysis (e.g. MLE) or Bayesian Data Analysis (e.g. MAP) approaches to learn distribution of data from samples
- More complex approaches:
 - Probabilistic Graphical Models
 - Generative Adversarial Networks (GANs)
 - Deep Generative Models

Clustering

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?

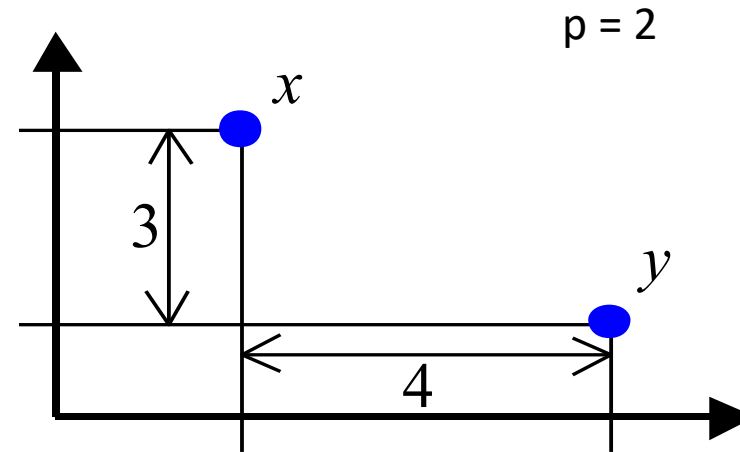


Hard to
define! But *we*
know it when
we see it

- 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

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



Euclidean distance

$$d(x, y) = \sqrt{\sum_{i=1}^p |x_i - y_i|^2}$$

5

Manhattan distance

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

7

Sup-distance

$$d(x, y) = \max_{1 \leq i \leq p} |x_i - y_i|$$

4

Correlation coefficient

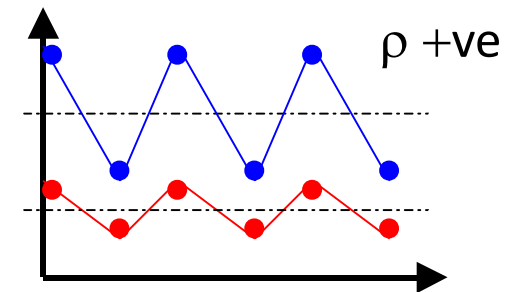
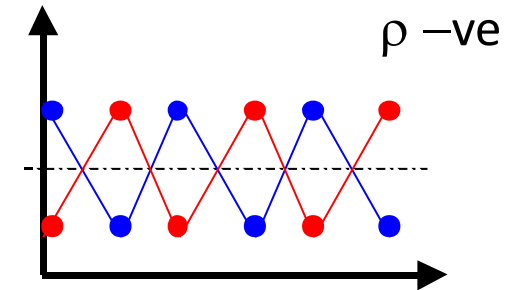
$$\mathbf{x} = (x_1, x_2, \dots, x_p)$$
$$\mathbf{y} = (y_1, y_2, \dots, y_p)$$

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

Pearson correlation coefficient

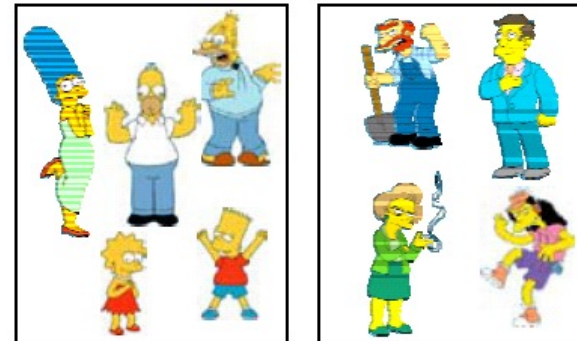
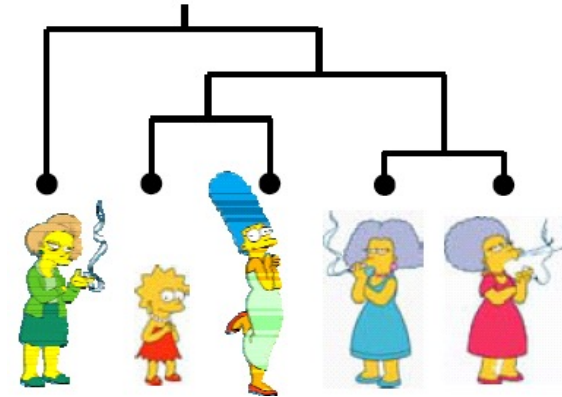
$$\rho(x, y) = \frac{\sum_{i=1}^p (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^p (x_i - \bar{x})^2 \times \sum_{i=1}^p (y_i - \bar{y})^2}}$$

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



Clustering Algorithms

- Hierarchical algorithms
 - Single-linkage
 - Average-linkage
 - Complete-linkage
 - Centroid-based
- Partition algorithms
 - K means clustering
 - Mixture-Model based clustering



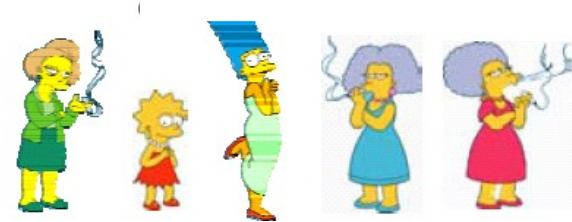
Hierarchical Clustering

- Bottom-Up Agglomerative Clustering

Starts with each object in a separate cluster, and repeat:

- Joins the most similar pair of clusters,
- Update the similarity of the new cluster to others until there is only one cluster.

Greedy – less accurate but simple to implement

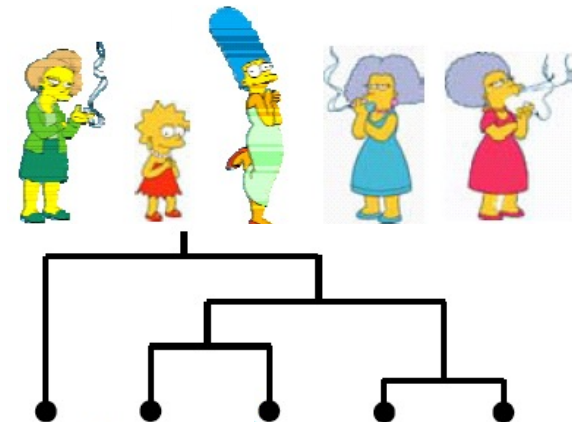


- Top-Down divisive

Starts with all the data in a single cluster, and repeat:

- Split each cluster into two using a partition algorithm
- Until each object is a separate cluster.

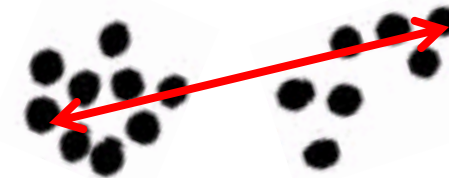
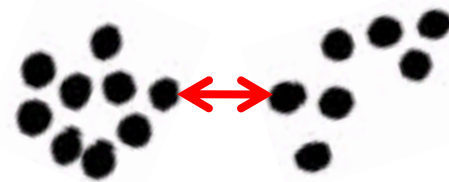
More accurate but complex to implement



Bottom-up Agglomerative clustering

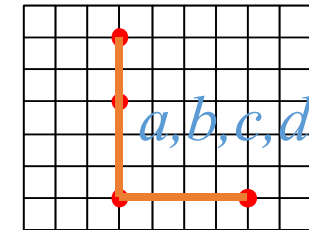
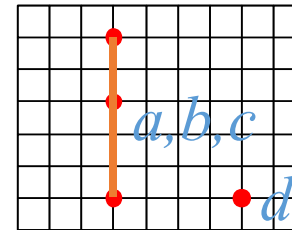
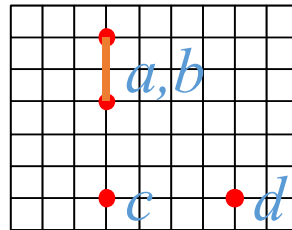
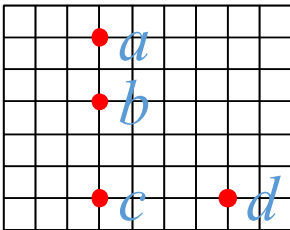
Different algorithms differ in how the similarities are defined (and hence updated) between two clusters

- Single-Linkage
 - Nearest Neighbor: similarity between their closest members.
- Complete-Linkage
 - Furthest Neighbor: similarity between their furthest members.
- Centroid
 - Similarity between the centers of gravity
- Average-Linkage
 - Average similarity of all cross-cluster pairs.



Single-Linkage Method

Euclidean Distance



(1)

(2)

(3)

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

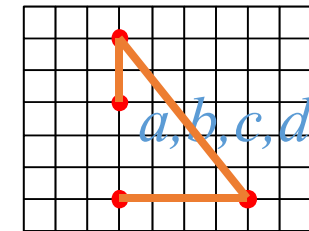
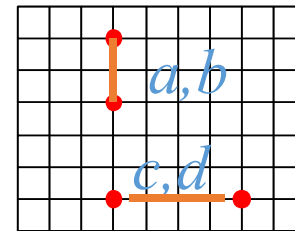
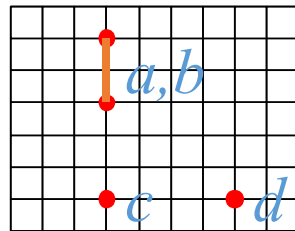
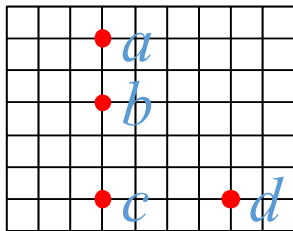
	<i>c</i>	<i>d</i>
<i>a, b</i>	3	5
<i>c</i>		4

	<i>d</i>
<i>a, b, c</i>	4

Distance Matrix

Complete-Linkage Method

Euclidean Distance



(1)

(2)

(3)

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

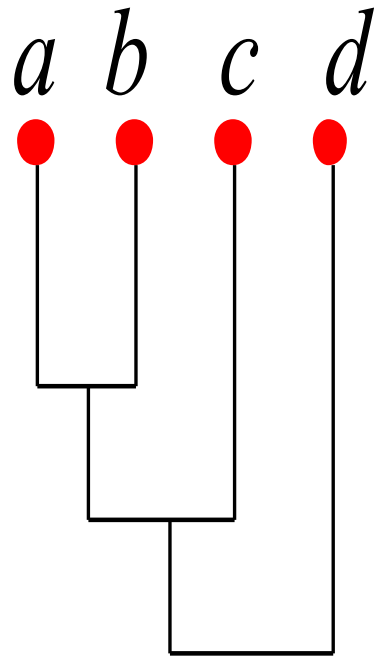
	<i>c</i>	<i>d</i>
<i>a, b</i>	5	6
<i>c</i>		4

	<i>c, d</i>
<i>a, b</i>	6

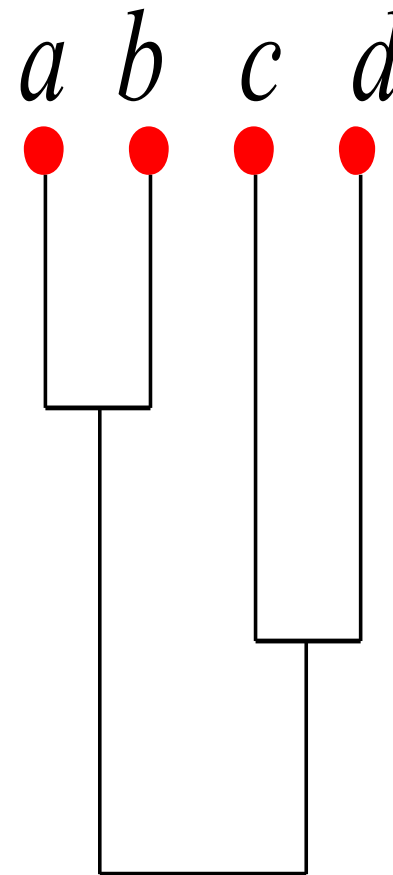
Distance Matrix

Dendrograms

Single-Linkage



Complete-Linkage



0

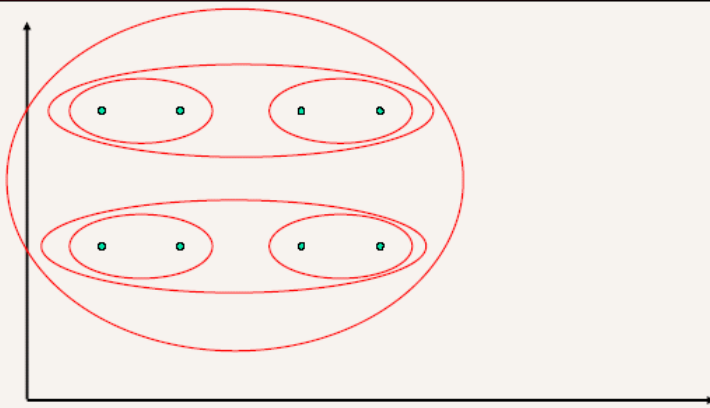
2

4

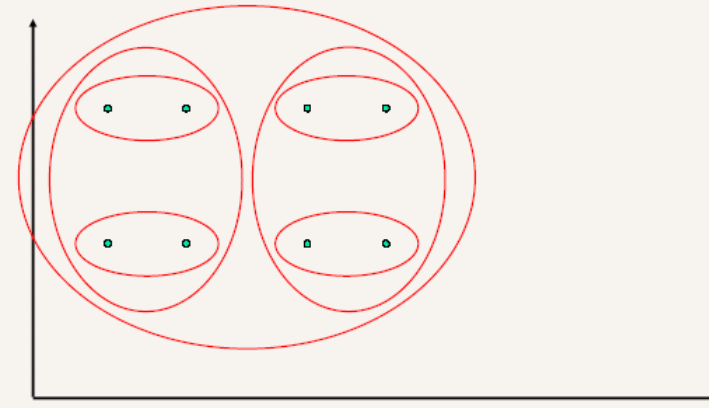
6

Another Example

Single Link Example



Complete Link Example



Single vs. Complete Linkage

Shape of clusters

Single-linkage

allows anisotropic and
non-convex shapes

Complete-linkage

assumes isotropic, convex
shapes



Computational Complexity

- All hierarchical clustering methods need to compute similarity of all pairs of n individual instances which is $O(n^2)$.
- At each iteration,
 - Find largest of the set of similarities $O(n^2)$
 - Update similarity between merged cluster and other clusters ... $O(n)$
 - Maximum no. of iterations ... $O(n)$
- So we get time complexity of $O(n^3)$
 - could be reduced with more complicated data structures such as heaps which however come with greater storage complexity

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
- 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 –

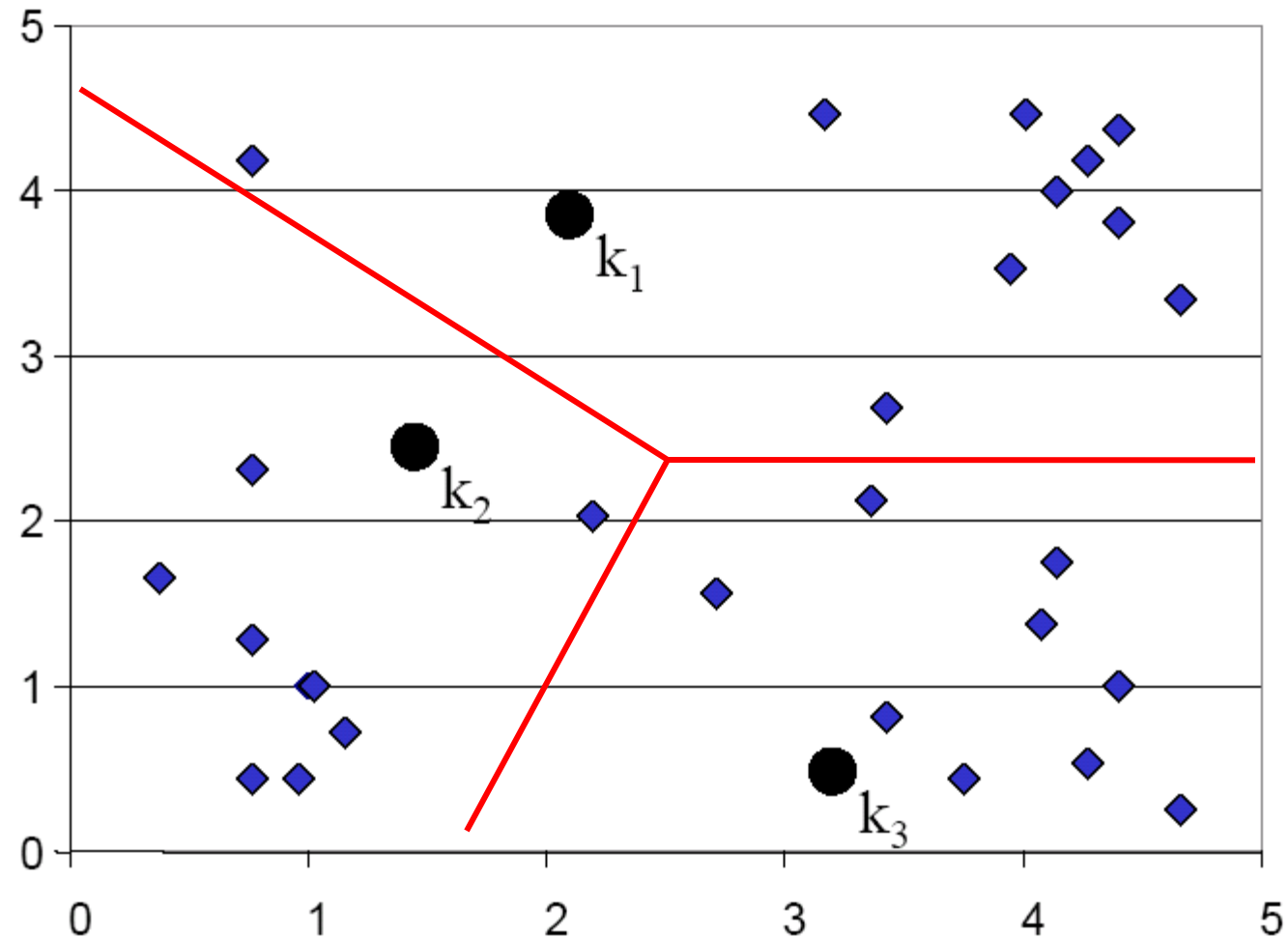
1. Assign points to the nearest cluster centers
2. Re-estimate the k cluster centers (aka the **centroid** or **mean**), by assuming the memberships found above are correct.

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

Termination –

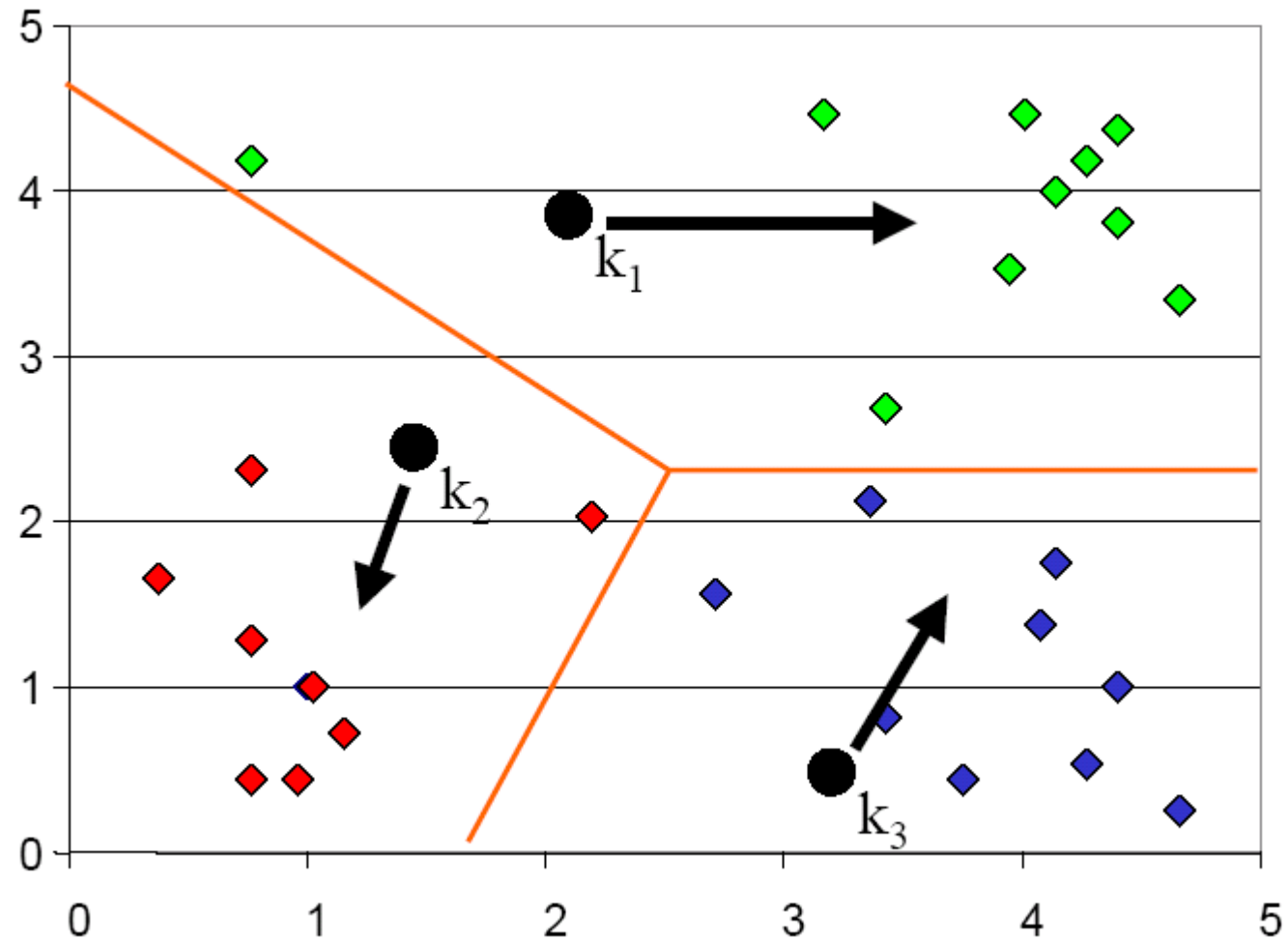
If none of the objects changed membership in the last iteration, exit. Otherwise go to 1.

K-means Clustering: Step 1

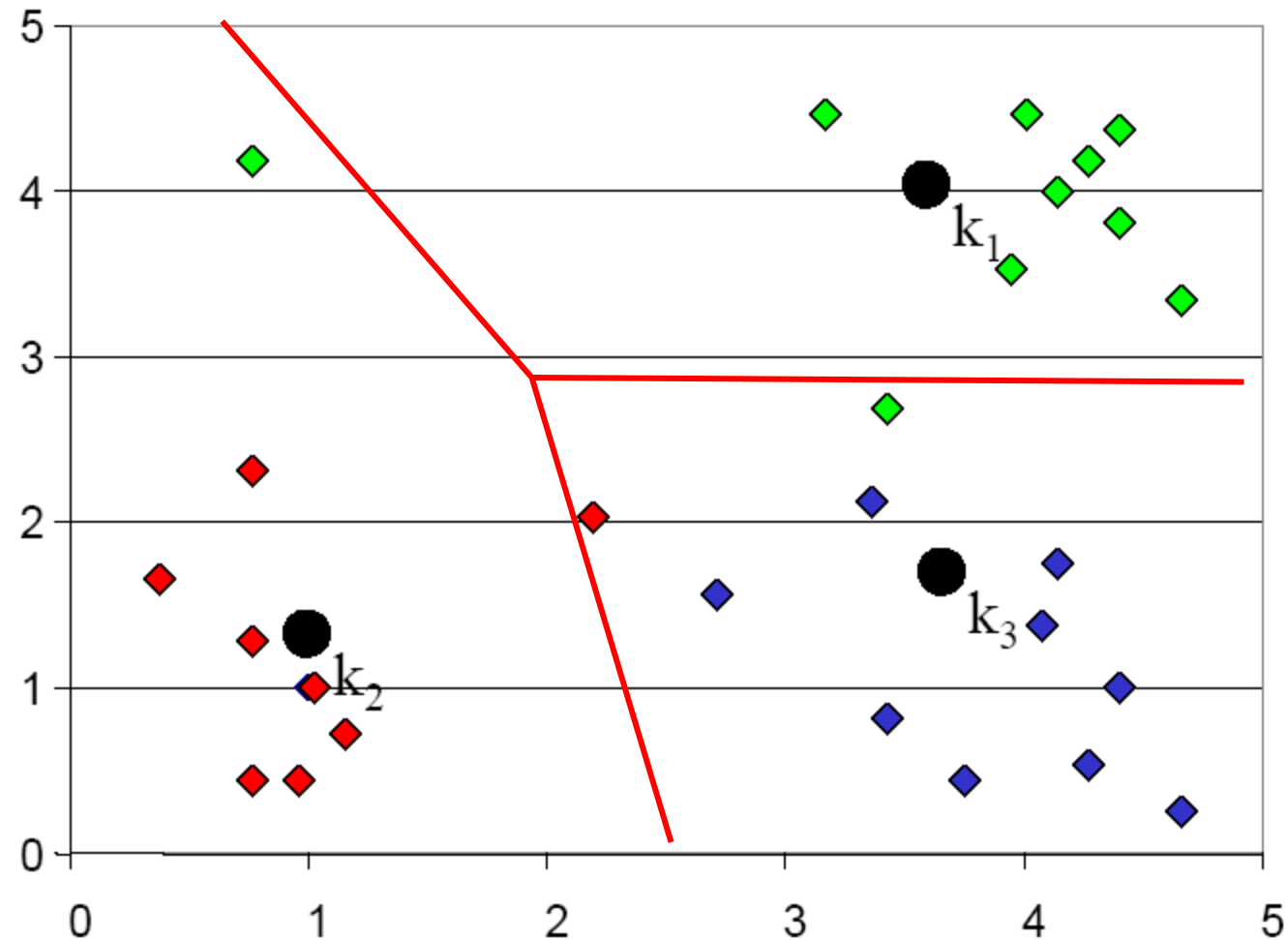


**Voronoi
diagram**

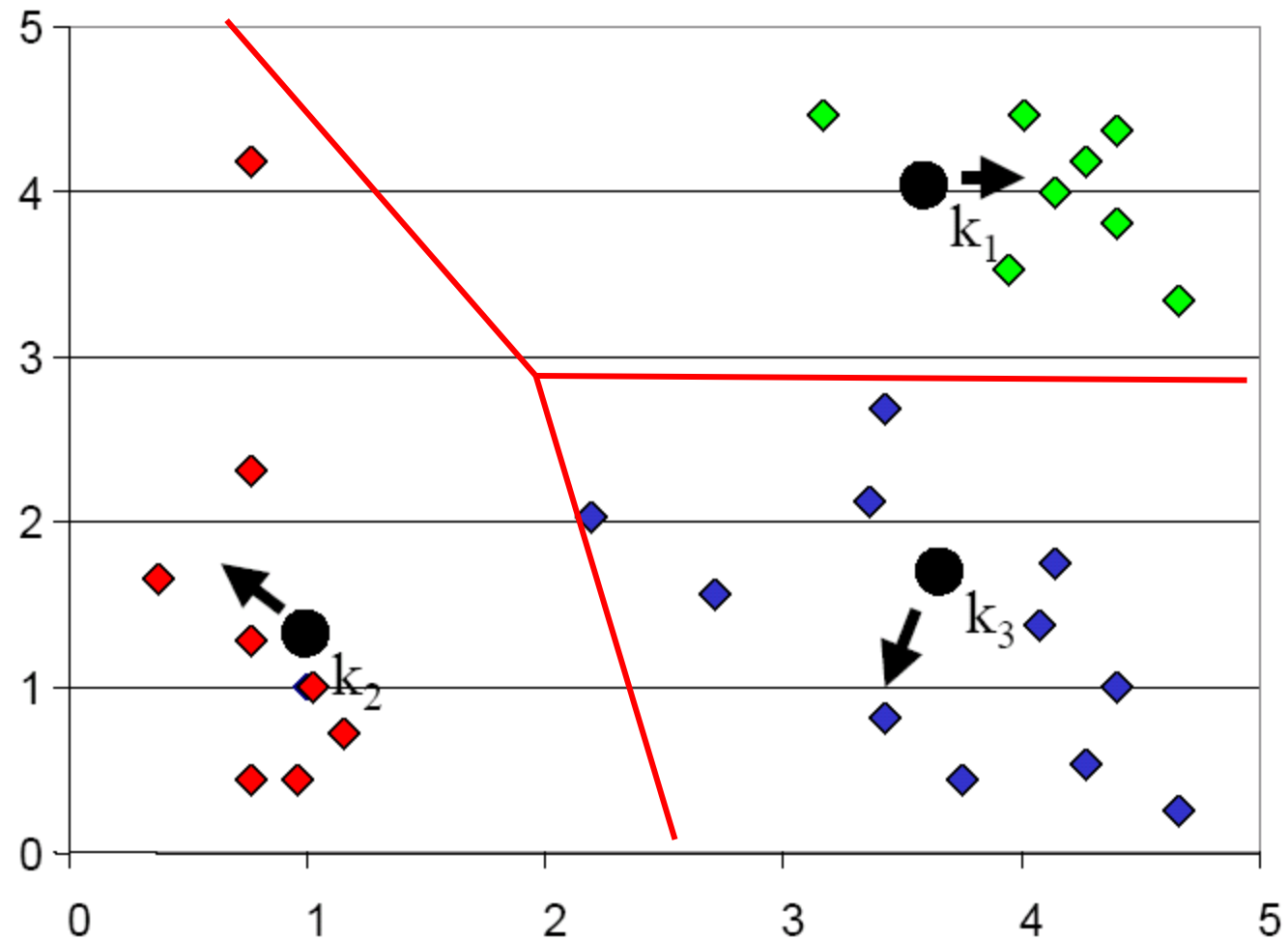
K-means Clustering: Step 2



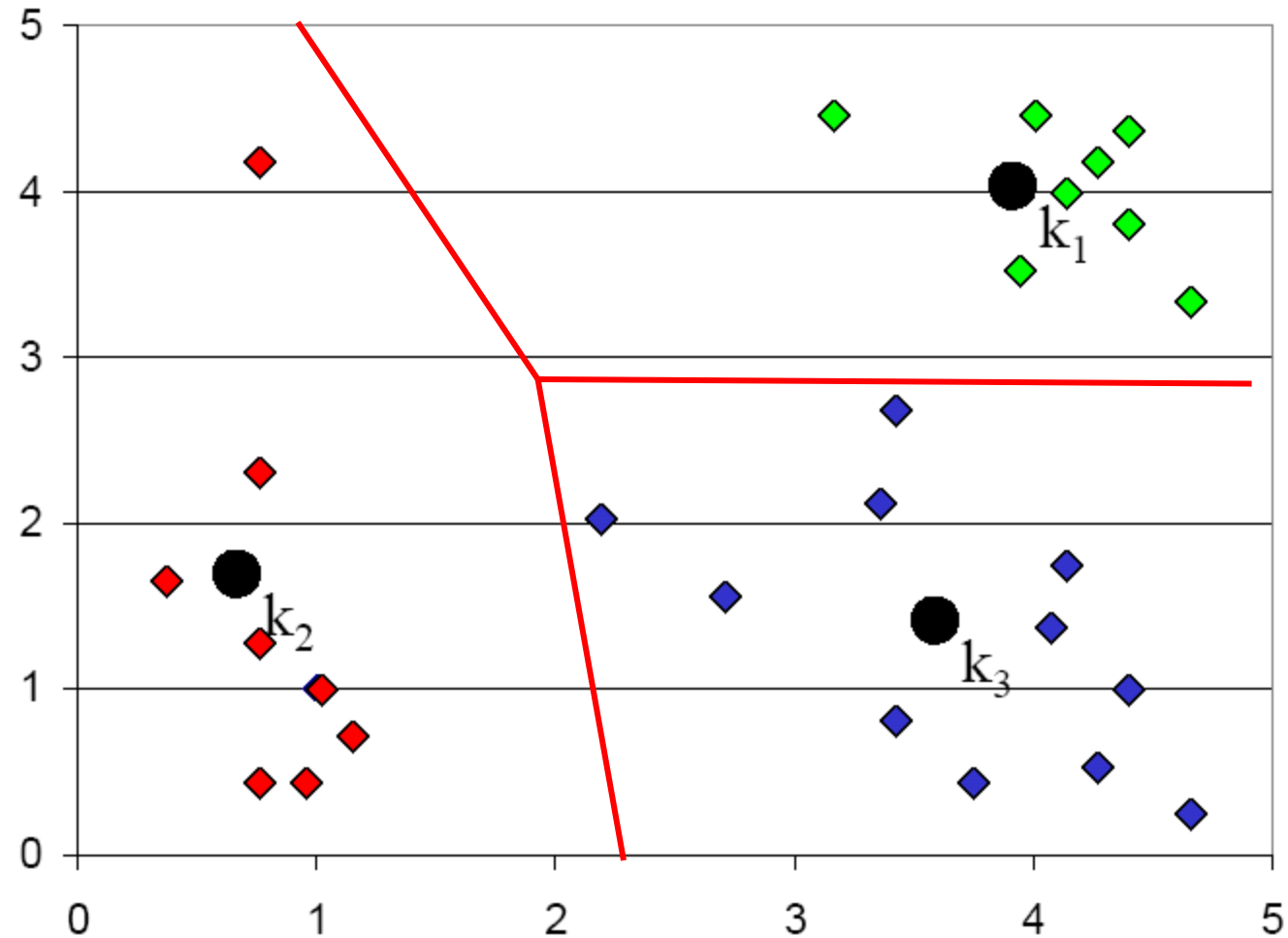
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



K-means Recap ...

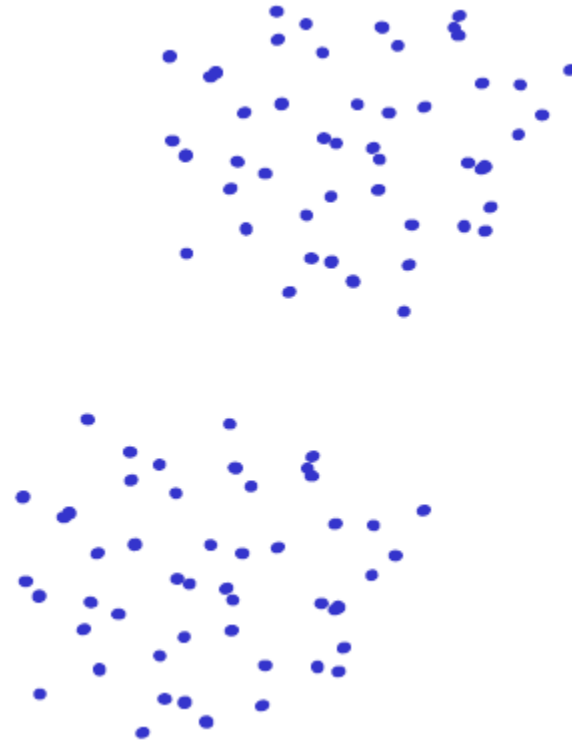
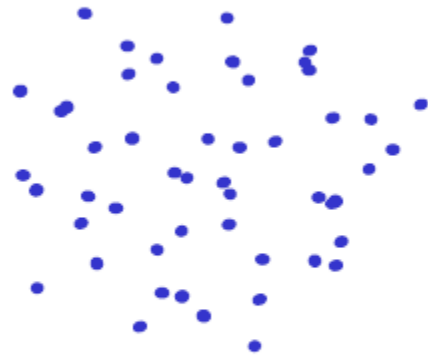
- Randomly initialize k centers
 - $\mu^{(0)} = \mu_1^{(0)}, \dots, \mu_k^{(0)}$
- **Classify:** Assign each point $j \in \{1, \dots, m\}$ to nearest center:
 - $C^{(t)}(j) \leftarrow \arg \min_{i=1, \dots, k} \|\mu_i^{(t)} - x_j\|^2$
- **Recenter:** μ_i becomes centroid of its points:
 - $\mu_i^{(t+1)} \leftarrow \arg \min_{\mu} \sum_{j: C^{(t)}(j)=i} \|\mu - x_j\|^2 \quad i \in \{1, \dots, k\}$
 - Equivalent to $\mu_i \leftarrow$ average of its points!

Computational Complexity

- At each iteration,
 - Computing distance between each of the n objects and the K cluster centers is $O(Kn)$.
 - Computing cluster centers: Each object gets added once to some cluster: $O(n)$.
- Assume these two steps are each done once for l iterations: $O(lKn)$.

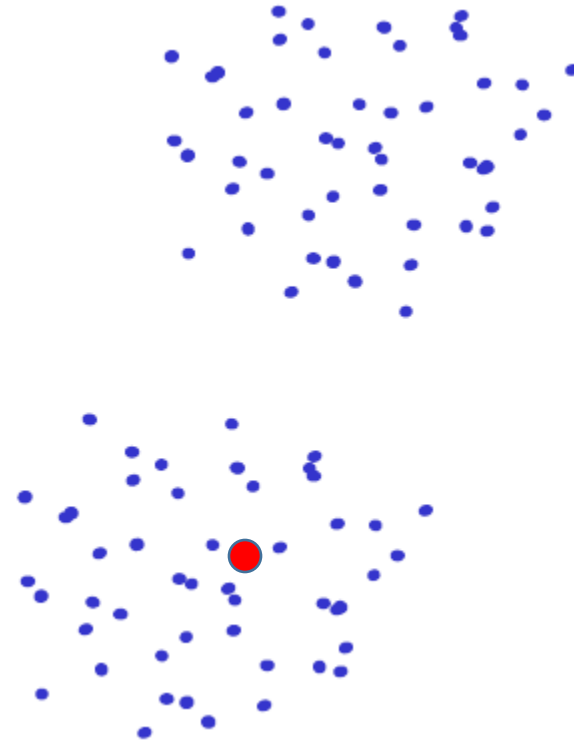
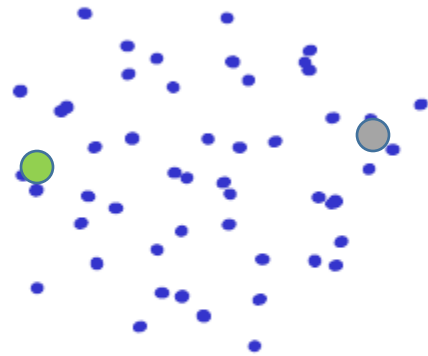
Seed Choice

- Results are quite sensitive to seed selection.



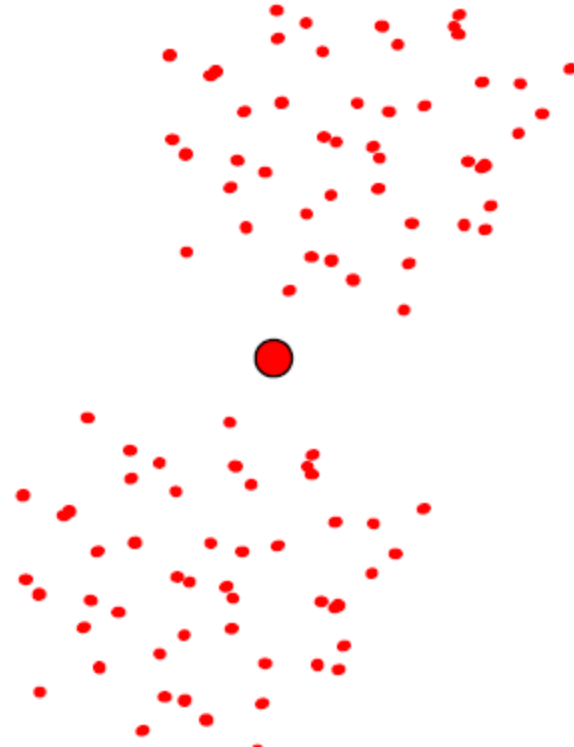
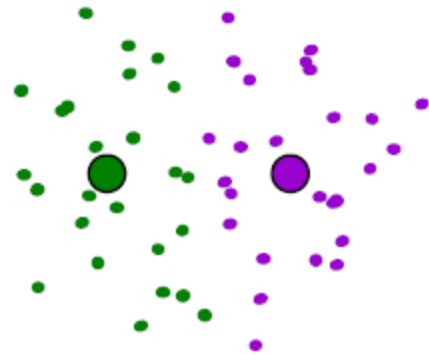
Seed Choice

- Results are quite sensitive to seed selection.



Seed Choice

- Results are quite sensitive to seed selection.



Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clustering.
- Select good seeds using a heuristic (e.g., object least similar to any existing mean)
- k-means ++ algorithm of Arthur and Vassilvitskii
 - key idea: choose centers that are far apart
 - probability of picking a point as cluster center proportional to distance from nearest center picked so far
- Try out multiple starting points (very important!!!)
- Initialize with the results of another method.

Other Issues

- Shape of clusters
 - Assumes isotropic, equal variance, convex clusters
- Sensitive to Outliers
 - use K-medoids

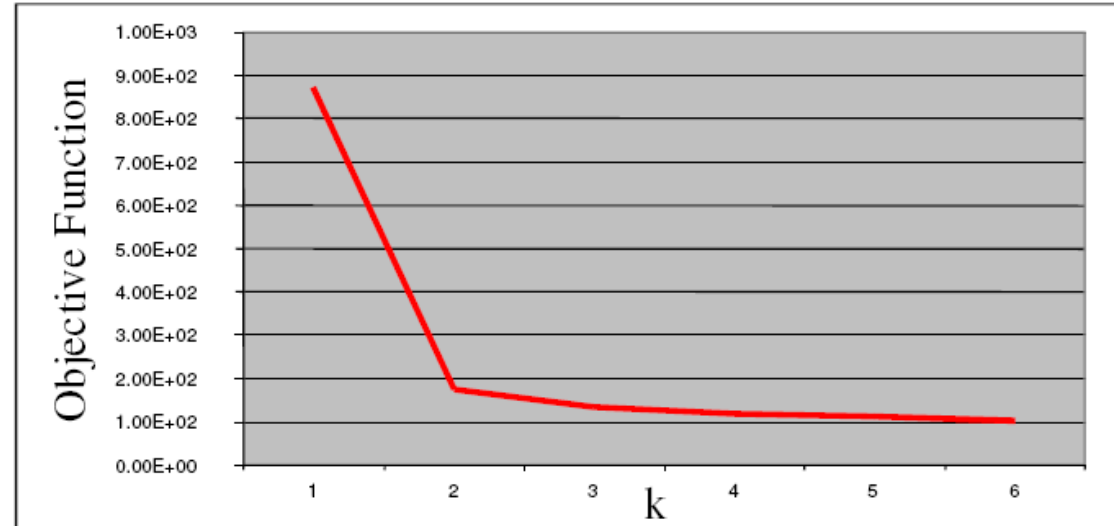


Other Issues

- Number of clusters K
 - SSE: sum of squared errors

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

- Look for “Knee” in SSE



Low Dimensional Structure

Low-Dimensional Structure

- When data has too many features, a lot of machine learning algorithms “break down”
- Extracting “low-dimensional” structure from data can help with:
 - Dimensionality Reduction
 - Data Visualization
 - Data Compression
- Approaches:
 - Principal Component Analysis (Linear)
 - Manifold Learning (Non-Linear)

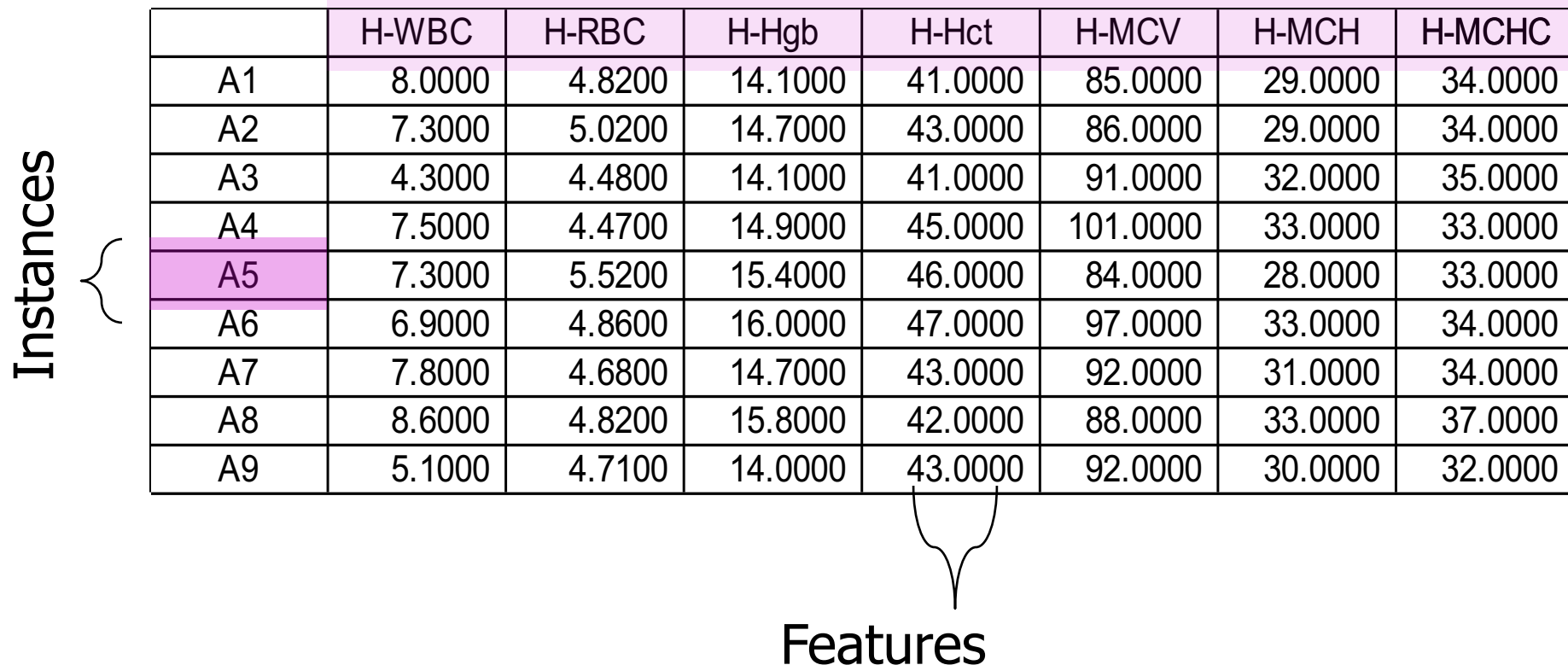
Data Visualization

Example:

- Given 53 blood samples (features) from 65 people.
- How can we visualize the measurements?

Data Visualization

- Matrix format (65x53)



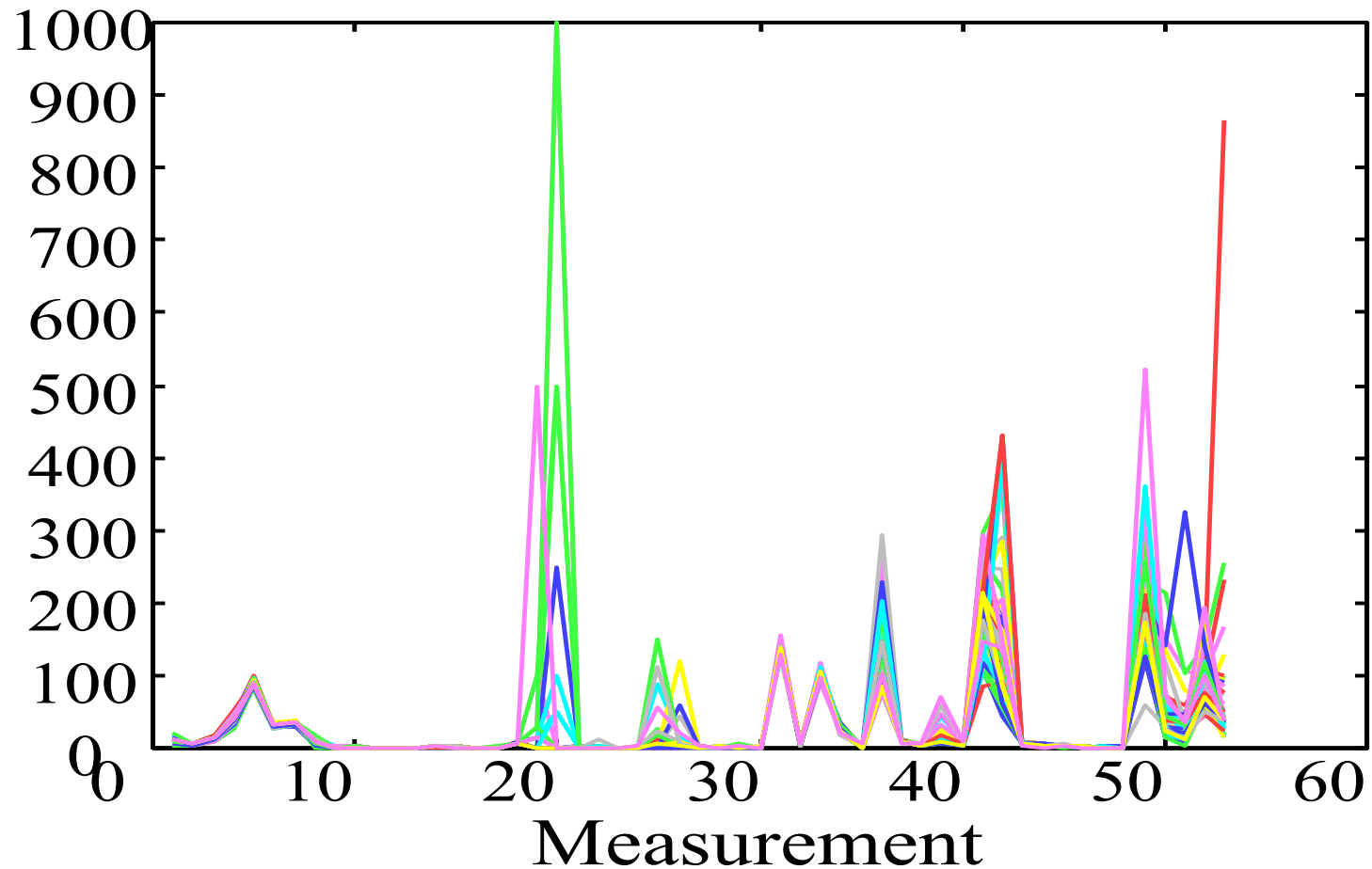
The diagram shows a data matrix with 9 rows and 8 columns. The first column contains instance labels A1 through A9. The subsequent columns contain numerical feature values. A bracket on the left side of the matrix is labeled 'Instances', and a bracket at the bottom is labeled 'Features'. The row for instance A5 is highlighted in pink, and the header row is also highlighted in pink.

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

Difficult to see the correlations between the features...

Data Visualization

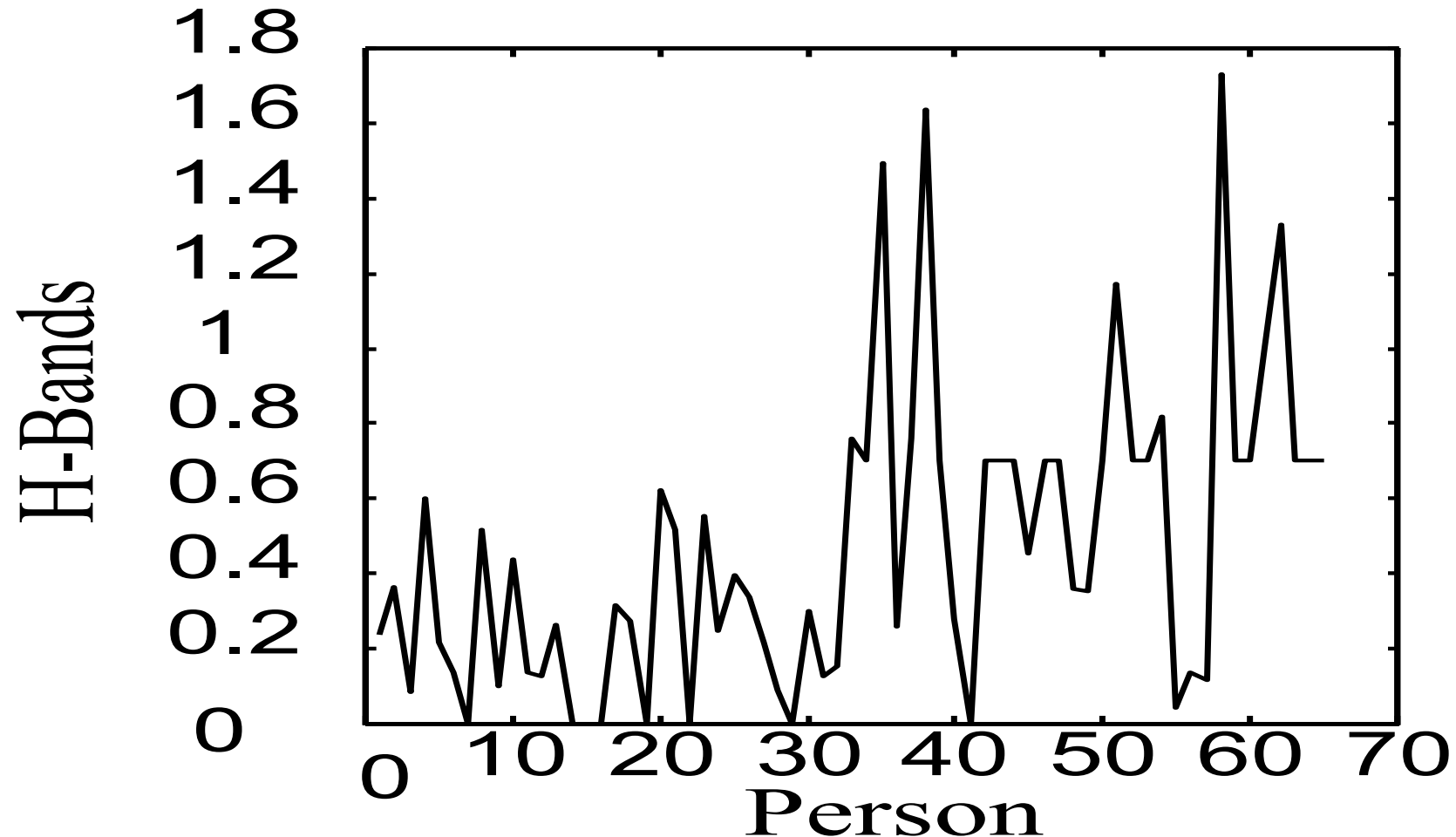
- Curves (65 curves, one for each person)



Difficult to compare the different patients...

Data Visualization

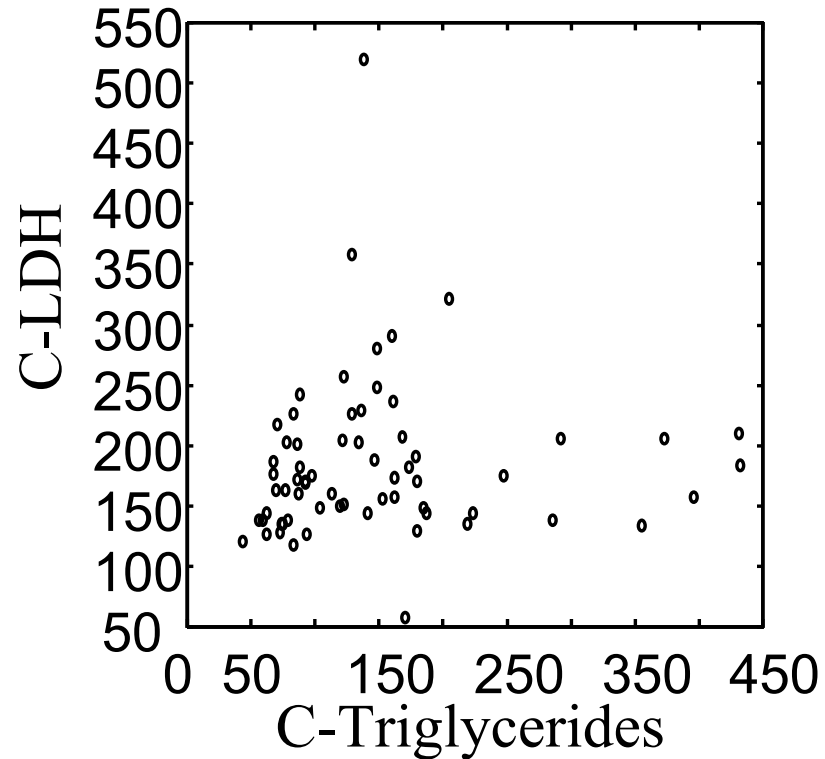
- Curves (53 pictures, one for each feature)



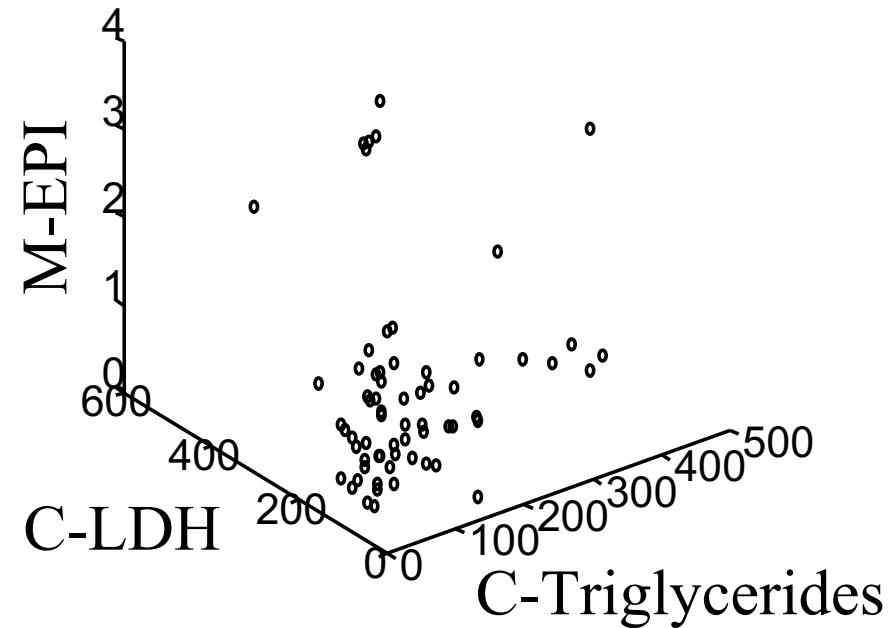
Difficult to see the correlations between the features...

Data Visualization

Bi-variate



Tri-variate



How can we visualize the other variables???

... difficult to see in 4 or higher dimensional spaces...

Original Image



- ❑ Divide the original 372x492 image into patches:
 - Each patch is an instance that contains 12x12 pixels on a grid
- ❑ Consider each as a 144-D vector

Compression: each patch 144D \rightarrow 60D



Low-Dimensional Structure

- When data has too many features, a lot of machine learning algorithms “break down”
- Extracting “low-dimensional” structure from data can help with:
 - Dimensionality Reduction
 - Data Visualization
 - Data Compression
- Approaches:
 - Principal Component Analysis (Linear)
 - Manifold Learning (Non-Linear)