Foundations of Data Science
Lecture 6

Rumi Chunara, PhD

CS3943/9223

So Far…

- What is Data Science?
- Intro to R
- Data cleaning, sampling, processing
- Intro to ML – what is it
- Two Basic Algorithms
  - kNN
  - Linear Regression
- Time-series Analyses
  - Regression and lagged data in R

Summary: Classification, so far

- Nearest-neighbor and k-nearest-neighbor classifiers
  - Euclidian distance, Cosine distance, Jaccard distance, etc.
- Support vector machines
  - Linear classifiers
  - Margin maximization
  - The kernel trick
  - Multi-class
- Of course, there are many other classifiers out there
  - Neural networks, boosting, decision trees, …
  - Deep neural networks, convolutional neural networks: jointly learning feature representation and classifiers
- Real world: exploit domain specific structure!

Machine Learning

There are two main categories of machine learning: **supervised learning** and **unsupervised learning**.

Unsupervised learning:

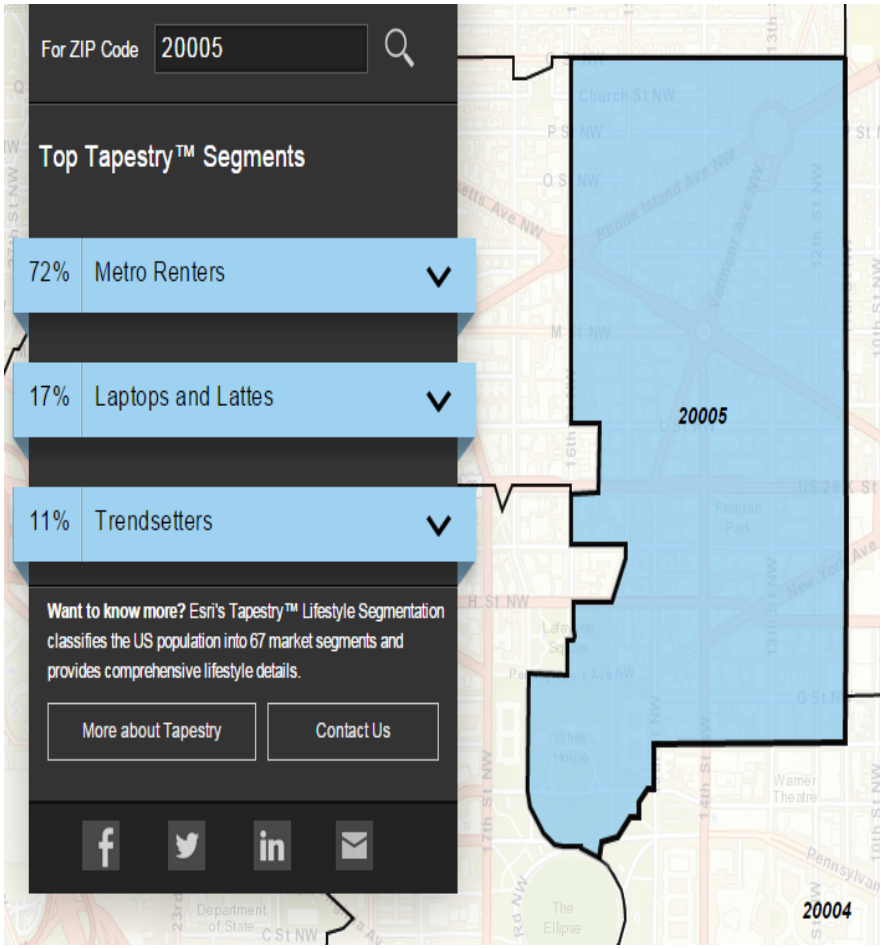
- Extracting structure from data
- Example: segment grocery store shoppers into “clusters” that exhibit similar behaviors
- Goal is “representation”

Supervised learning vs. Unsupervised learning

- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

Unsupervised Learning Example

Classify US residential neighborhoods into 67 unique segments based on demographic and socioeconomic characteristics



Metro Renters:

Young, mobile, educated, or still in school, we live alone or with a roommate in rented apartments or condos in the center of the city. Long hours and hard work don't deter us; we're willing to take risks to get to the top of our professions… We buy groceries at Whole Foods and Trader Joe's and shop for clothes at Banana Republic, Nordstrom, and Gap. We practice yoga, go skiing, and attend Pilates sessions.

Source: http://www.esri.com/landing-pages/tapestry/

Unsupervised Learning

Unsupervised learning has some clear differences from supervised learning. With **unsupervised learning:**

- There is no clear objective
- There is no “right anwser” (hard to tell how well you are doing)
- There is no response variable, just observations with features
- Labeled data is not required

Unsupervised Learning Example

Unsupervised learning example: Image clustering

- Input data: Images from Google
- Features: Numerical representations of the images
- Response: There isn't one (no hand-labeling required!)

1. Perform **unsupervised learning**
   - Cluster the images based on "similarity"
   - Might find a "dog cluster", might not
   - You're done!

Sometimes, unsupervised learning is used as a "preprocessing" step for supervised learning.

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, association rule mining is also unsupervised
- This chapter focuses on clustering.

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.

What is clustering for?

- Let us see some real-life examples
- Example 1: groups people of similar sizes together to make "small", "medium" and "large" T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- Example 2: In marketing, segment customers according to their similarities
  - To do targeted marketing.

What is clustering for? (cont…)

- Example 3: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- In fact, clustering is one of the most utilized data mining techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance ⇒ maximized
  - Intra-clusters distance ⇒ minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application.

Clustering – Why?

Clustering has one or more goals:

- Segment a large set of cases into small subsets that can be treated similarly - segmentation
- Generate a more compact description of a dataset - compression
- Model an underlying process that generates the data as a mixture of different, localized processes – representation

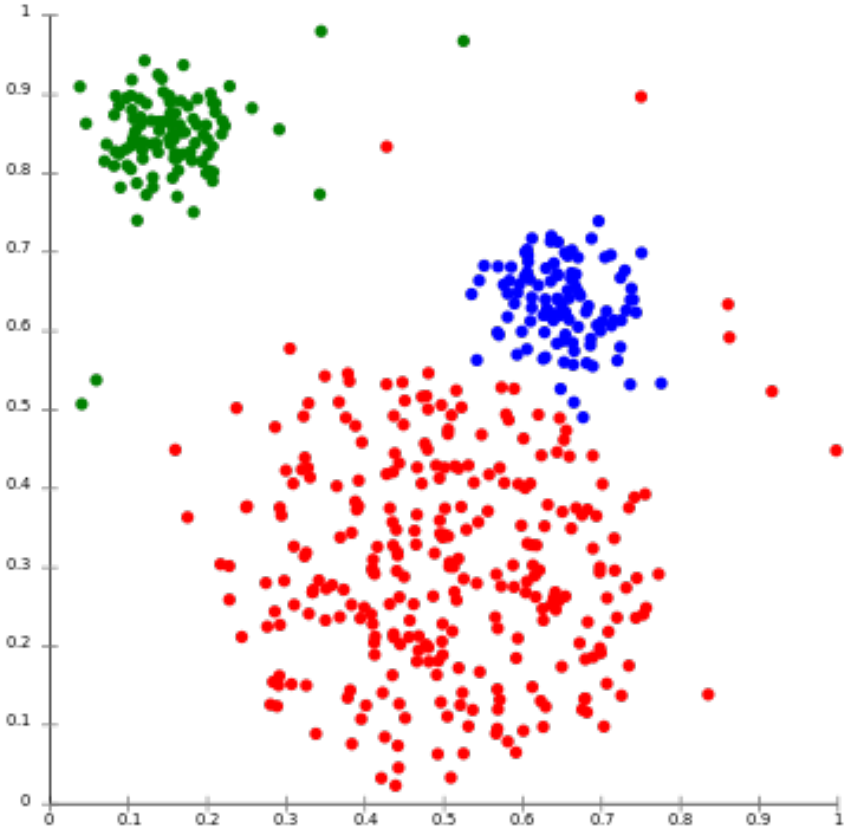
Clustering – Why?

Examples:

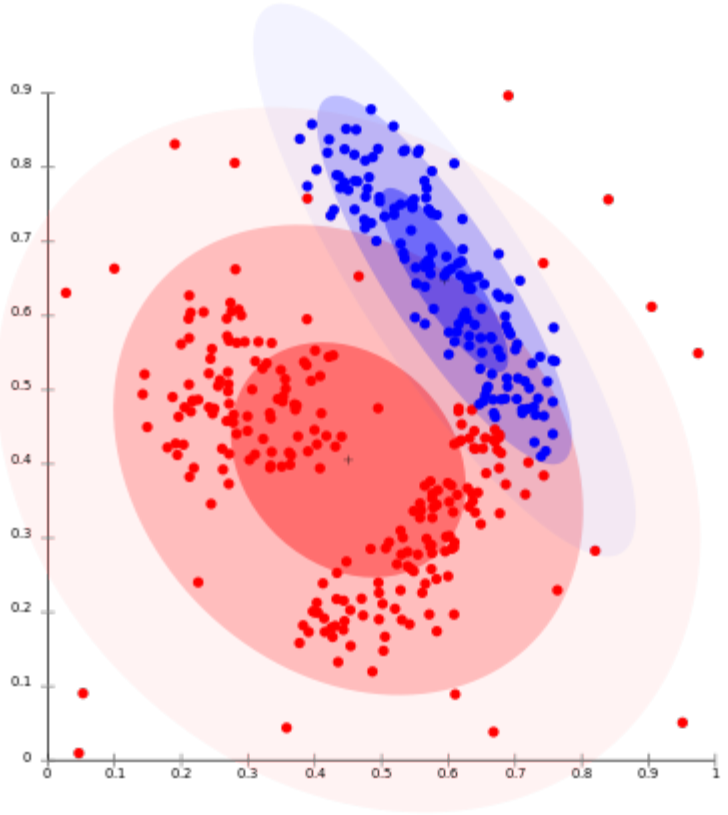
- Segment: image segmentation
- Compression: Cluster-based kNN, e.g. handwritten digit recognition.
- Underlying process: Accents people (because place of origin strongly influences the accent you have)

Stereotypical Clustering

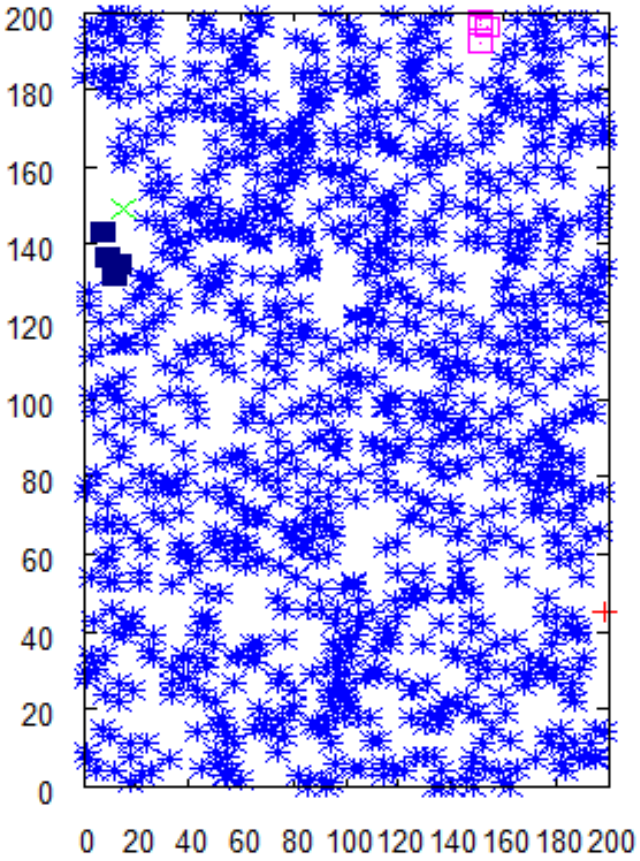
Note: Points are samples plotted in feature space, e.g. 10,000-dimensional space for 100x100 images.

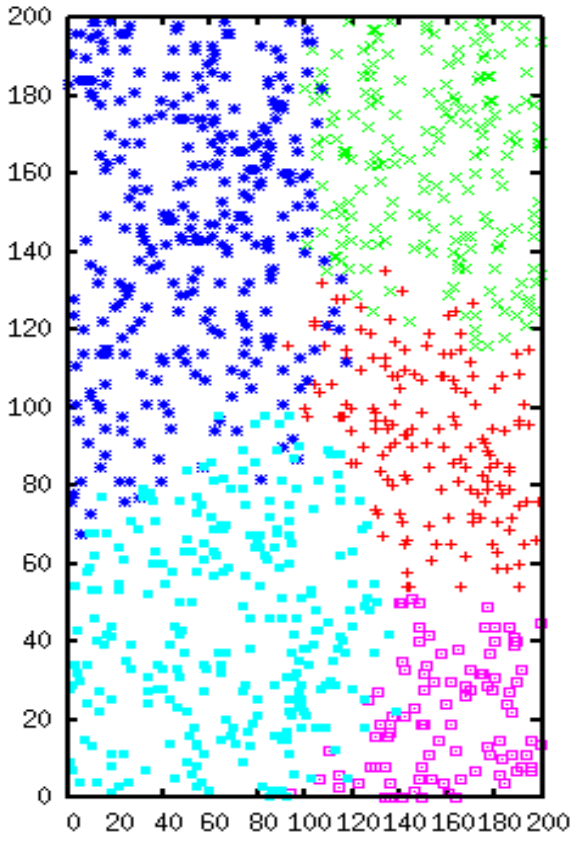


Model-based Clustering

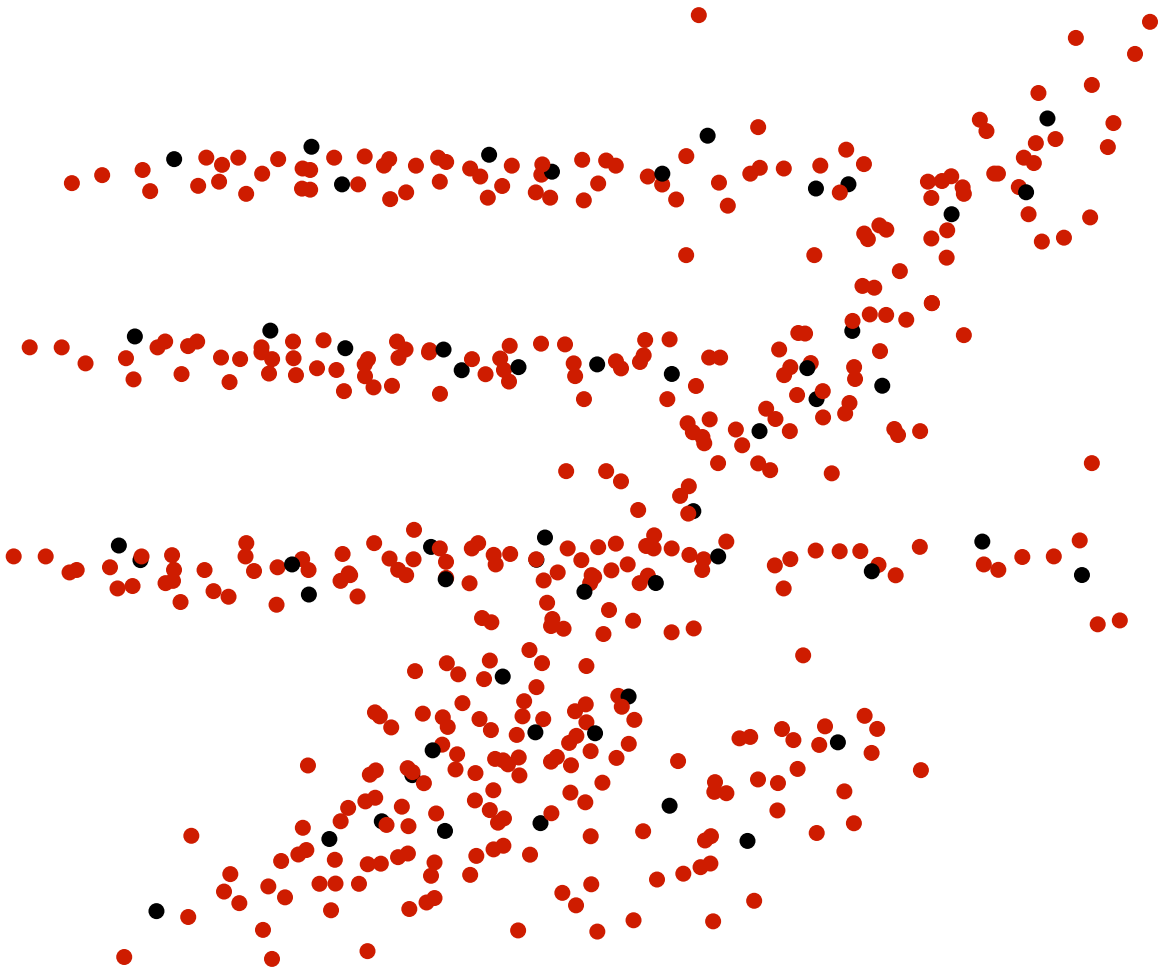


Clustering for Segmentation





Condensation/Compression



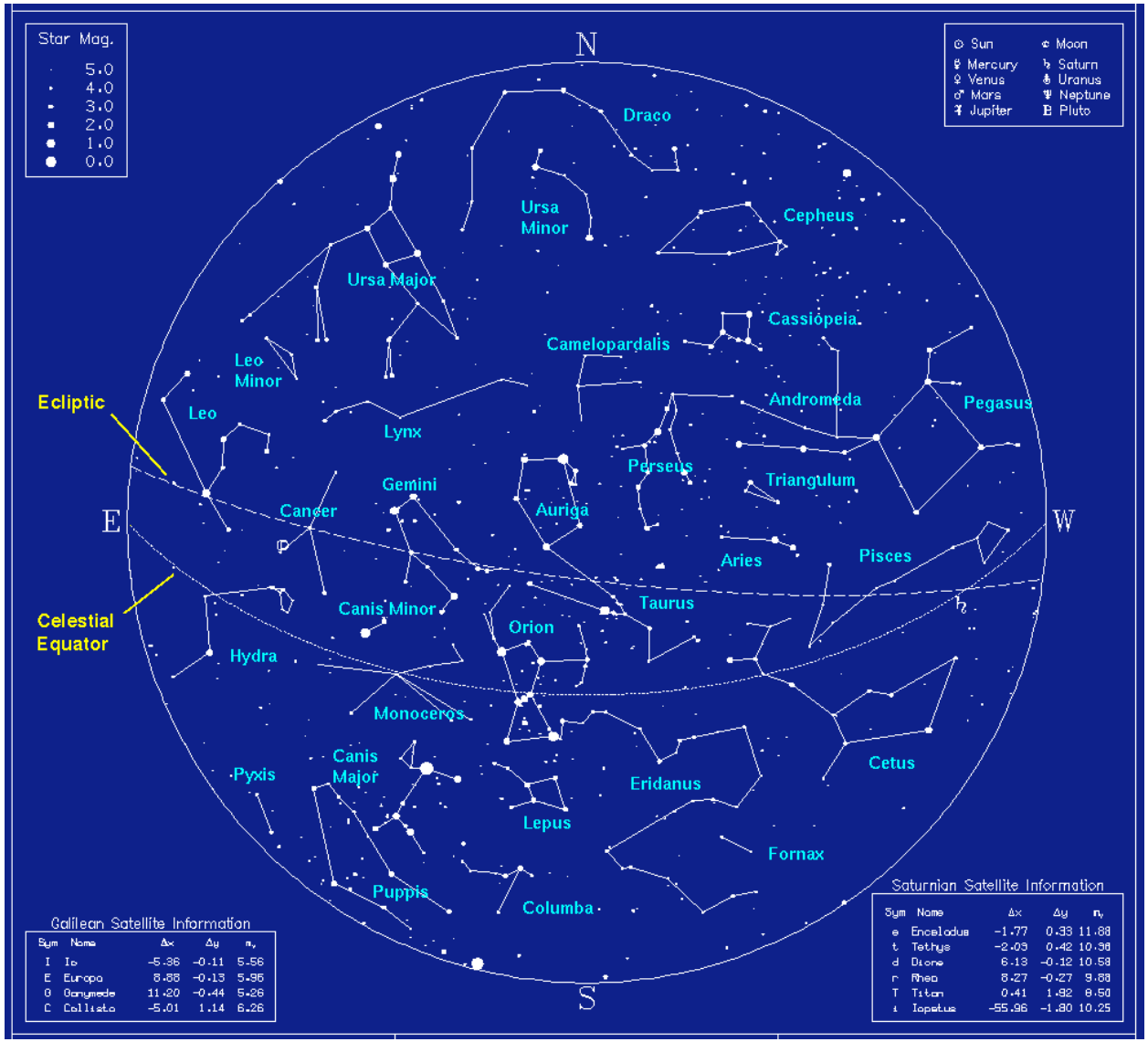
“Cluster Bias”

- Human beings conceptualize the world through categories represented as *examplars* (Rosch 1973, Estes 1994).



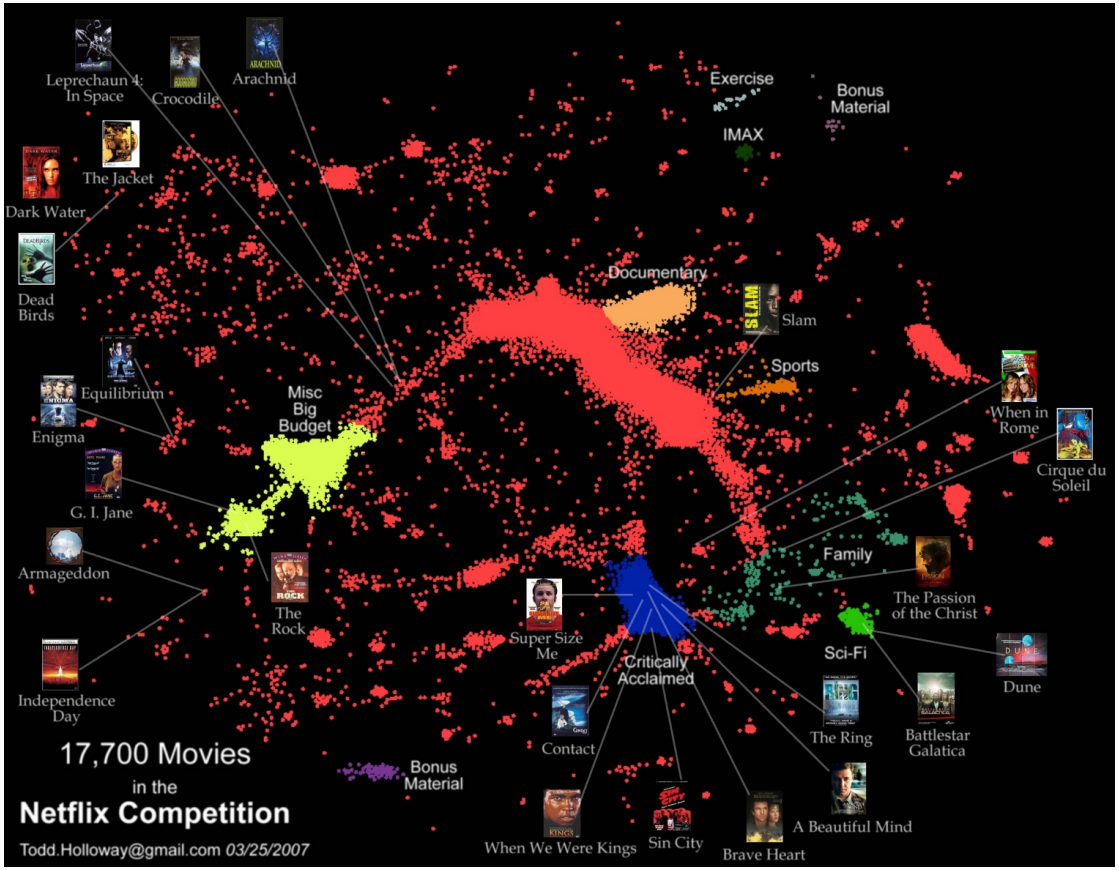
- We tend to see cluster structure whether it is there or not.
- Works well for dogs, but…

Cluster Bias



Netflix

- More of a continuum than discrete clusters
- Factor models, kNN do much better than discrete cluster models.



“Cluster Bias”

Upshot:

- **Clustering is used more than it should be**, because people assume an underlying domain has discrete classes in it.
- This is especially true for characteristics of people, e.g. Myers-Briggs personality types like “ENTP”.
- In reality the underlying data is usually **continuous**.
- Just as with Netflix, continuous models (dimension reduction, kNN) tend to do better.

Terminology

- **Hierarchical clustering:** clusters form a hierarchy. Can be computed bottom-up or top-down.
- **Flat clustering:** no inter-cluster structure.

- **Hard clustering:** items assigned to a unique cluster.
- **Soft clustering:** cluster membership is a real-valued function, distributed across several clusters.

K-means clustering

The standard k-means algorithm is based on **Euclidean distance**.

The cluster quality measure is an

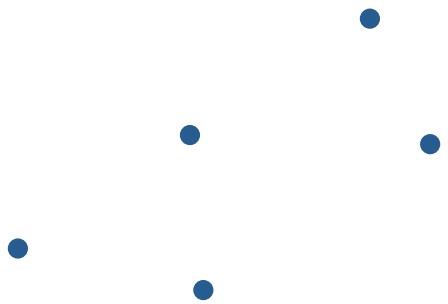
intra-cluster measure only

A simple greedy algorithm locally optimizes this measure (usually called Lloyd's algorithm):

- **Find the closest cluster center** for each item, and assign it to that cluster.
- **Recompute the cluster centroid** as the mean of items, for the newly-assigned items in the cluster.

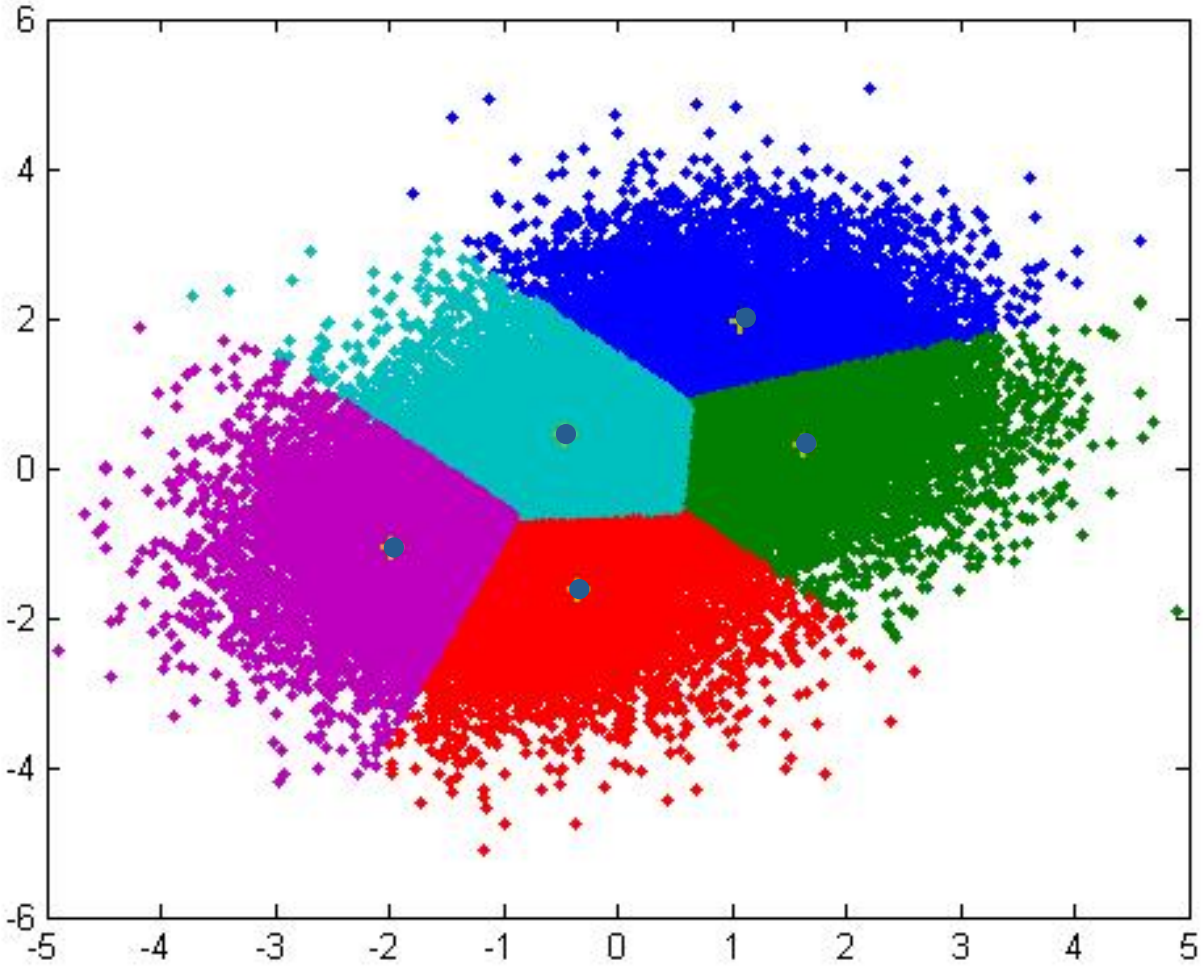
K-means clustering

Cluster centers – can pick by sampling the input data.

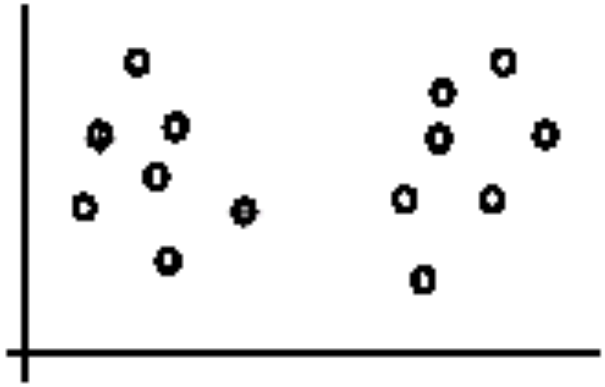


K-means clustering

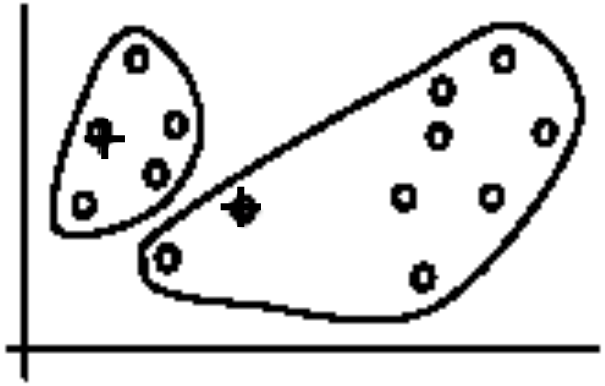
Assign points to closest center



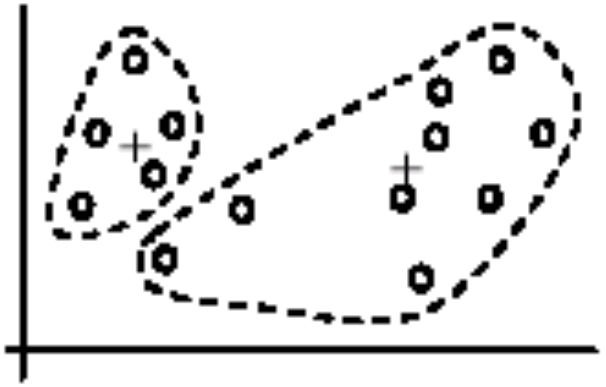
An example



(A). Random selection of k centers

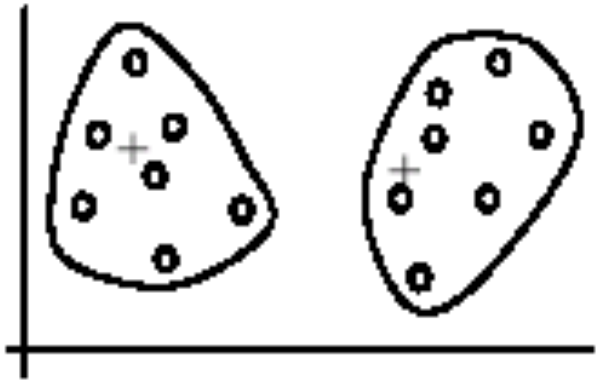


Iteration 1: (B). Cluster assignment

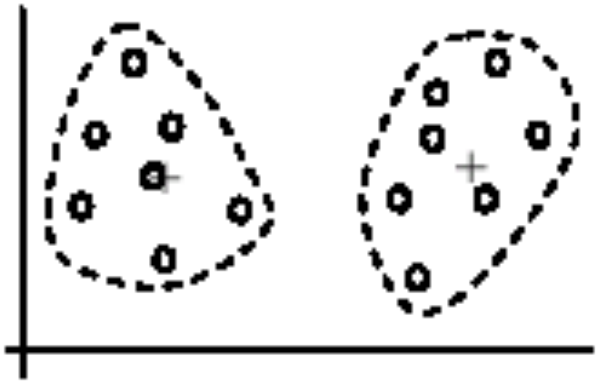


(C). Re-compute centroids

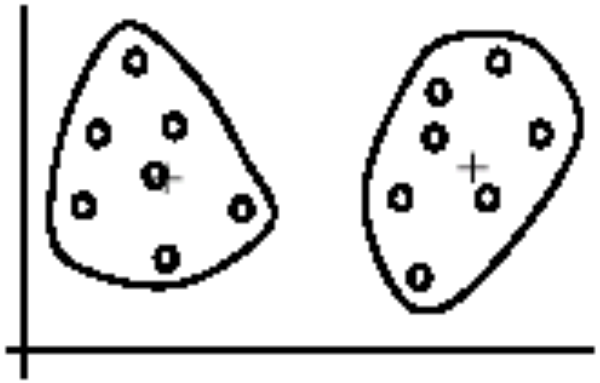
An example (cont …)



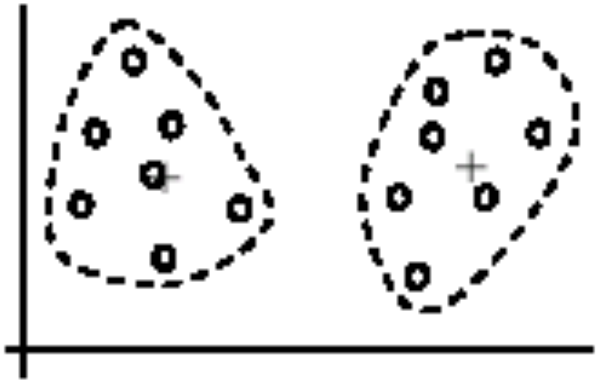
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

K-means clustering

Iterate (until – stoping criteria):

- For fixed number of iterations
- Until no change in assignments
- Until small change in quality

K-means properties

- It's a greedy algorithm with random setup – **solution isn't optimal** and varies significantly with different initial points.
- Very simple convergence proofs.
- **Performance is O(nk) per iteration**, not bad and can be heuristically improved.
  n = total features in the dataset, k = number clusters
- As a "local" clustering method, it works well for data condensation/compression.

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

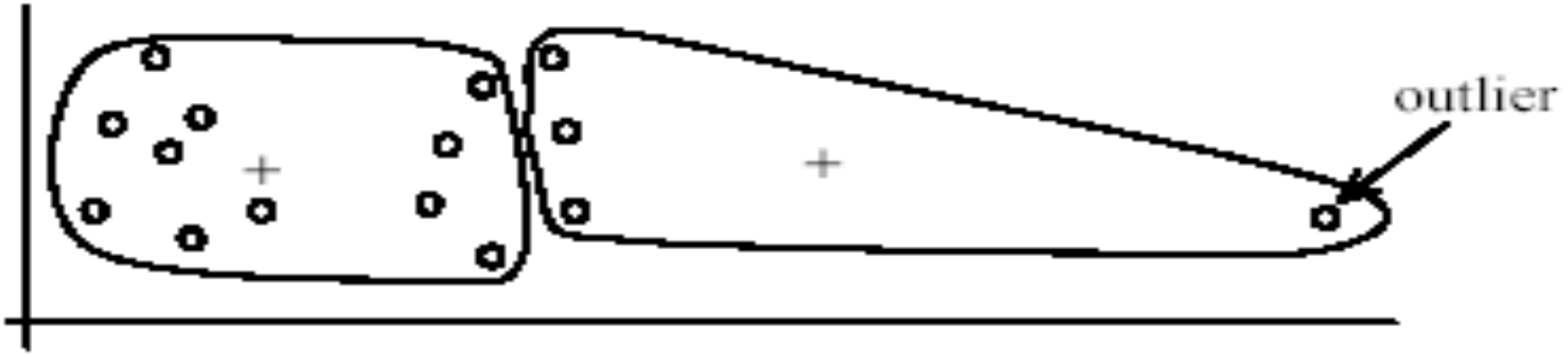
    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

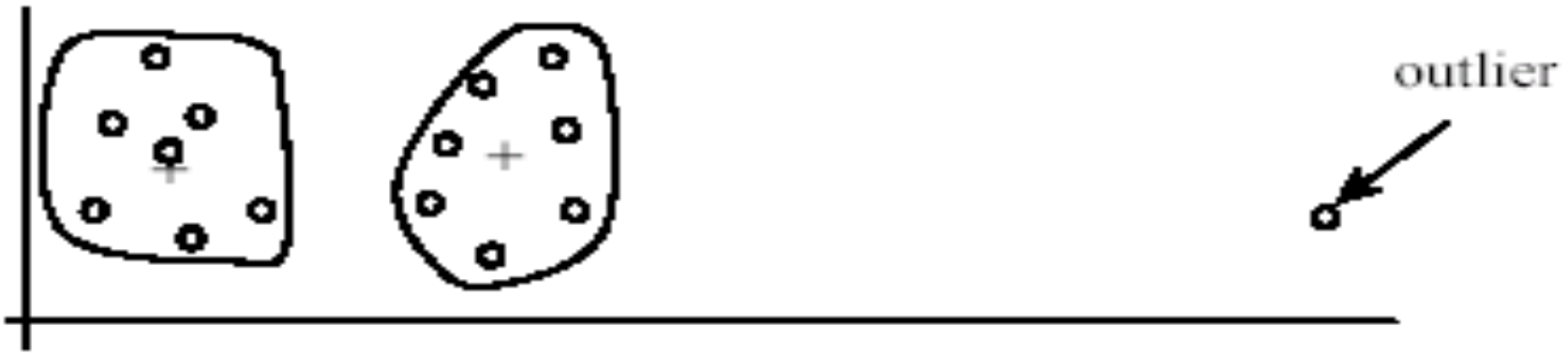
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



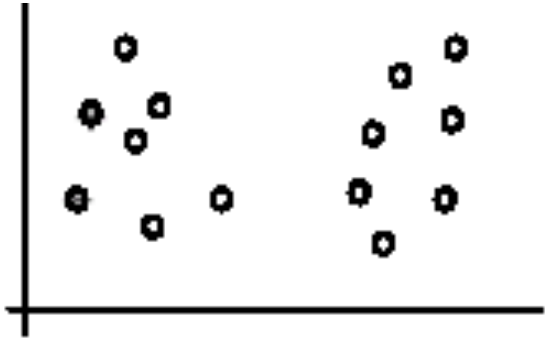
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

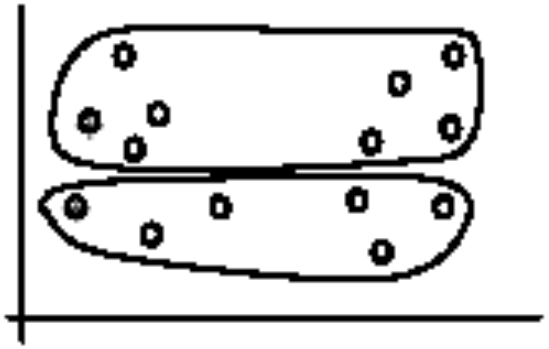
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont …)

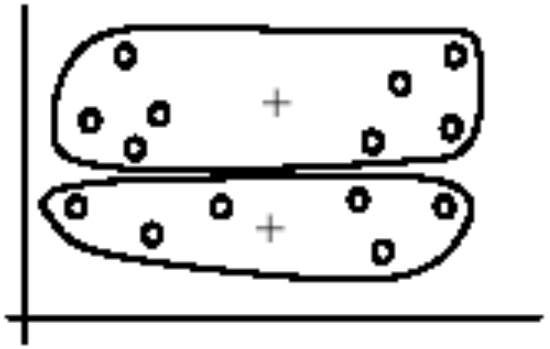
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



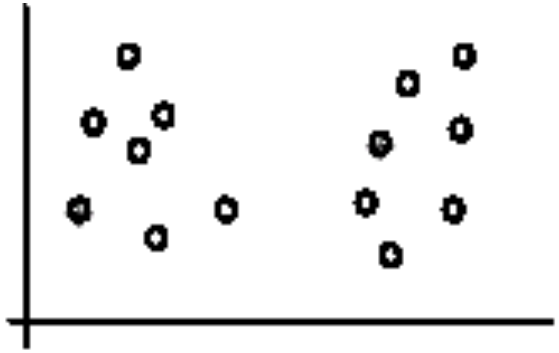
(B). Iteration 1



(C). Iteration 2

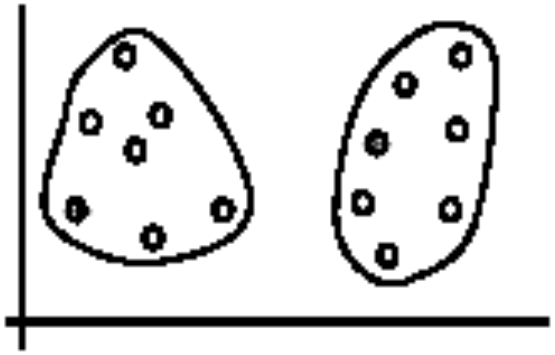
Weaknesses of k-means (cont …)

- If we use different seeds: good results

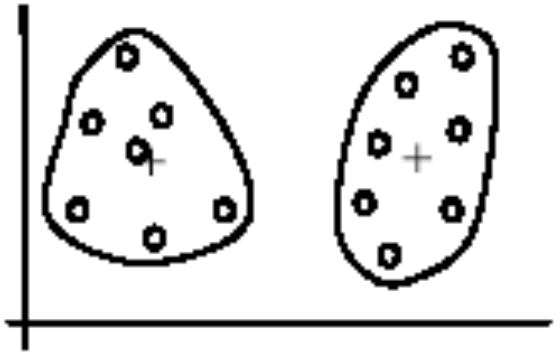


There are some methods to help choose good seeds

(A). Random selection of k seeds (centroids)



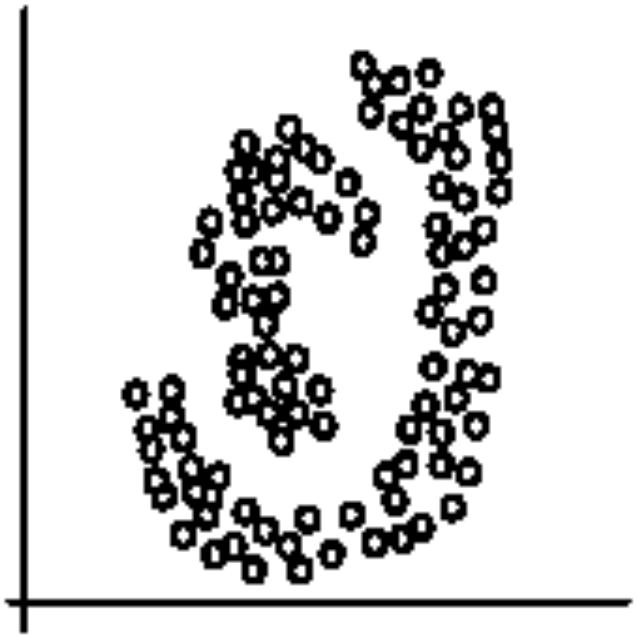
(B). Iteration 1



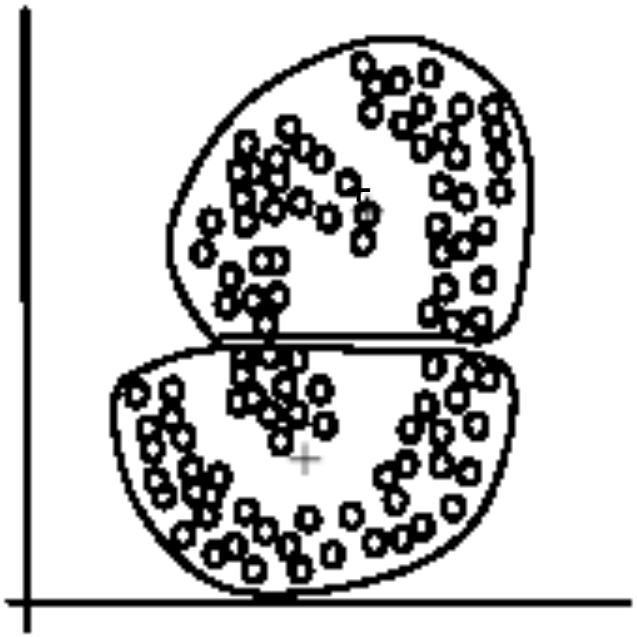
(C). Iteration 2

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

K-means summary

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and
  - other clustering algorithms have their own lists of weaknesses.
- No clear evidence that any other clustering algorithm performs better in general
  - although they may be more suitable for some specific types of data or applications.
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Choosing clustering dimension

- AIC or Akaike Information Criterion:

$$\textbf{AIC:} \qquad K = \arg\min_{K}[-2L(K) + 2q(K)]$$

- K=dimension, L(K) is the likelihood (could be RSS) and q(K) is a measure of model complexity (cluster description complexity).

- AIC favors more compact (fewer clusters) clusterings.

- For sparse data, AIC will incorporate the number of non-zeros in the cluster spec. Lower is better.

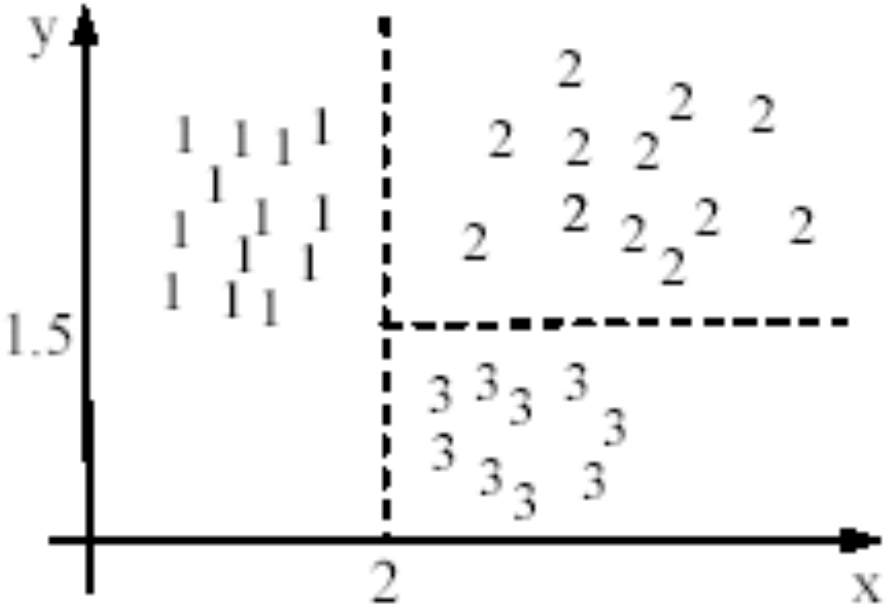
Common ways to represent clusters

- Use the centroid of each cluster to represent the cluster.
  - compute the radius and
  - standard deviation of the cluster to determine its spread in each dimension

  - The centroid representation alone works well if the clusters are of the hyper-spherical shape.
  - If clusters are elongated or are of other shapes, centroids are not sufficient

Using classification model

- All the data points in a cluster are regarded to have the same class label, e.g., the cluster ID.
  - run a supervised learning algorithm on the data to find a classification model.



$x \leq 2 \rightarrow$ cluster 1

$x > 2, y > 1.5 \rightarrow$ cluster 2

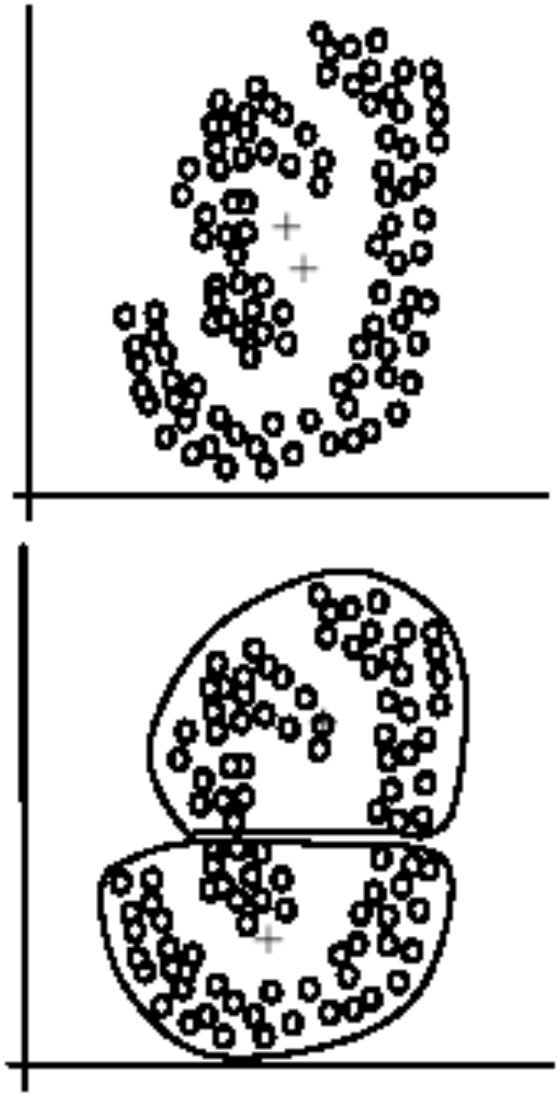
$x > 2, y \leq 1.5 \rightarrow$ cluster 3

Use frequent values to represent cluster

- This method is mainly for clustering of categorical data (e.g., *k*-modes clustering).
- Main method used in text clustering, where a small set of frequent words in each cluster is selected to represent the cluster.

Clusters of arbitrary shapes

- Hyper-elliptical and hyper-spherical clusters are usually easy to represent, using their centroid together with spreads.
- Irregular shape clusters are hard to represent. They may not be useful in some applications.
  - Using centroids are not suitable (upper figure) in general
  - K-means clusters may be more useful (lower figure), e.g., for making 2 size T-shirts.



Principal Components Analysis

- Data Visualization
- Data Compression
- Noise Reduction
- Data Classification
- Trend Analysis
- Factor Analysis

Data Visualization

Example:

- Given 53 blood and urine samples (features) from 65 people.

- How can we visualize the measurements?

Data Visualization

- Matrix format (65x53)

	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

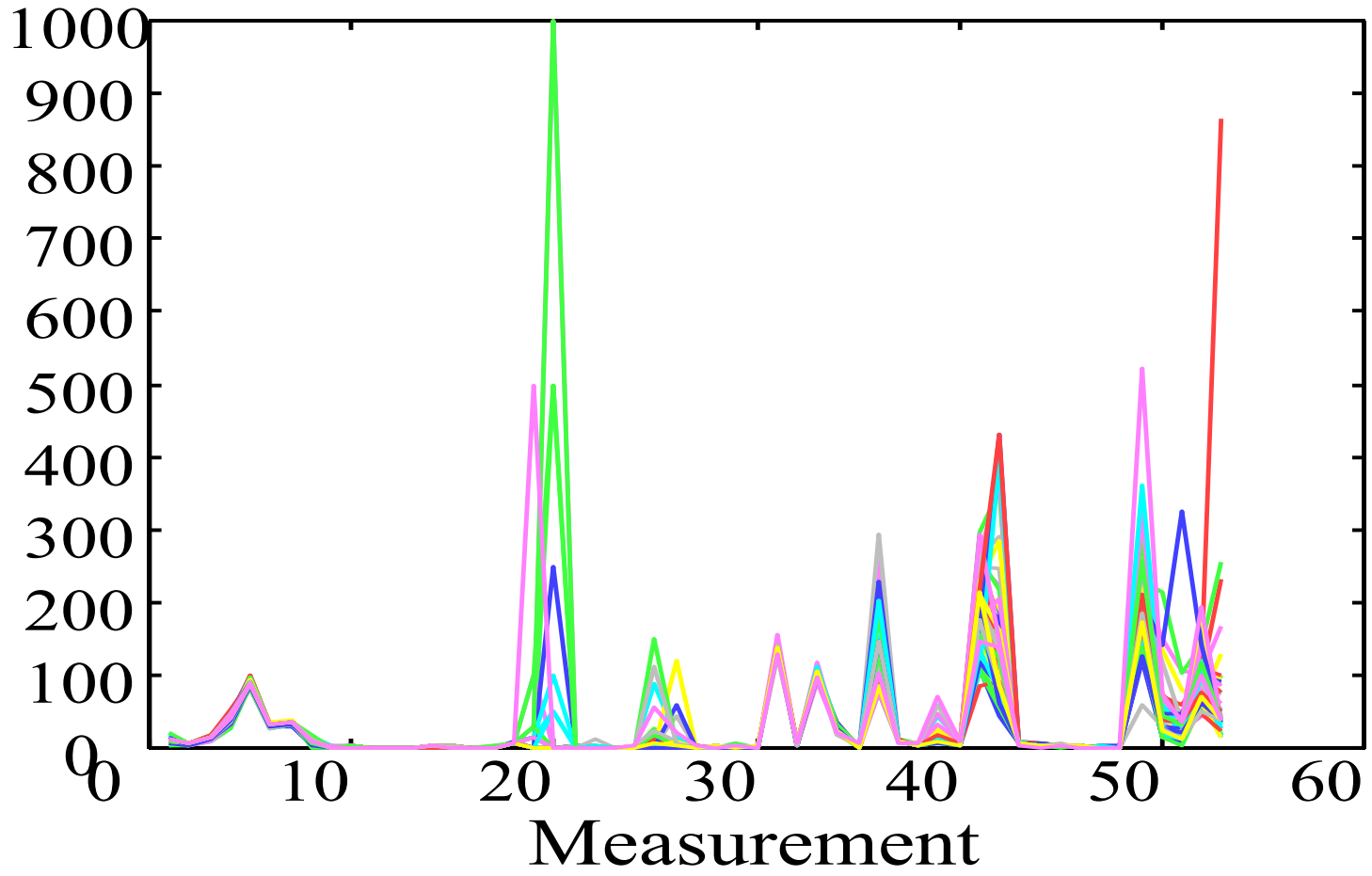
Instances

Features

Difficult to see the correlations between the features...

Data Visualization

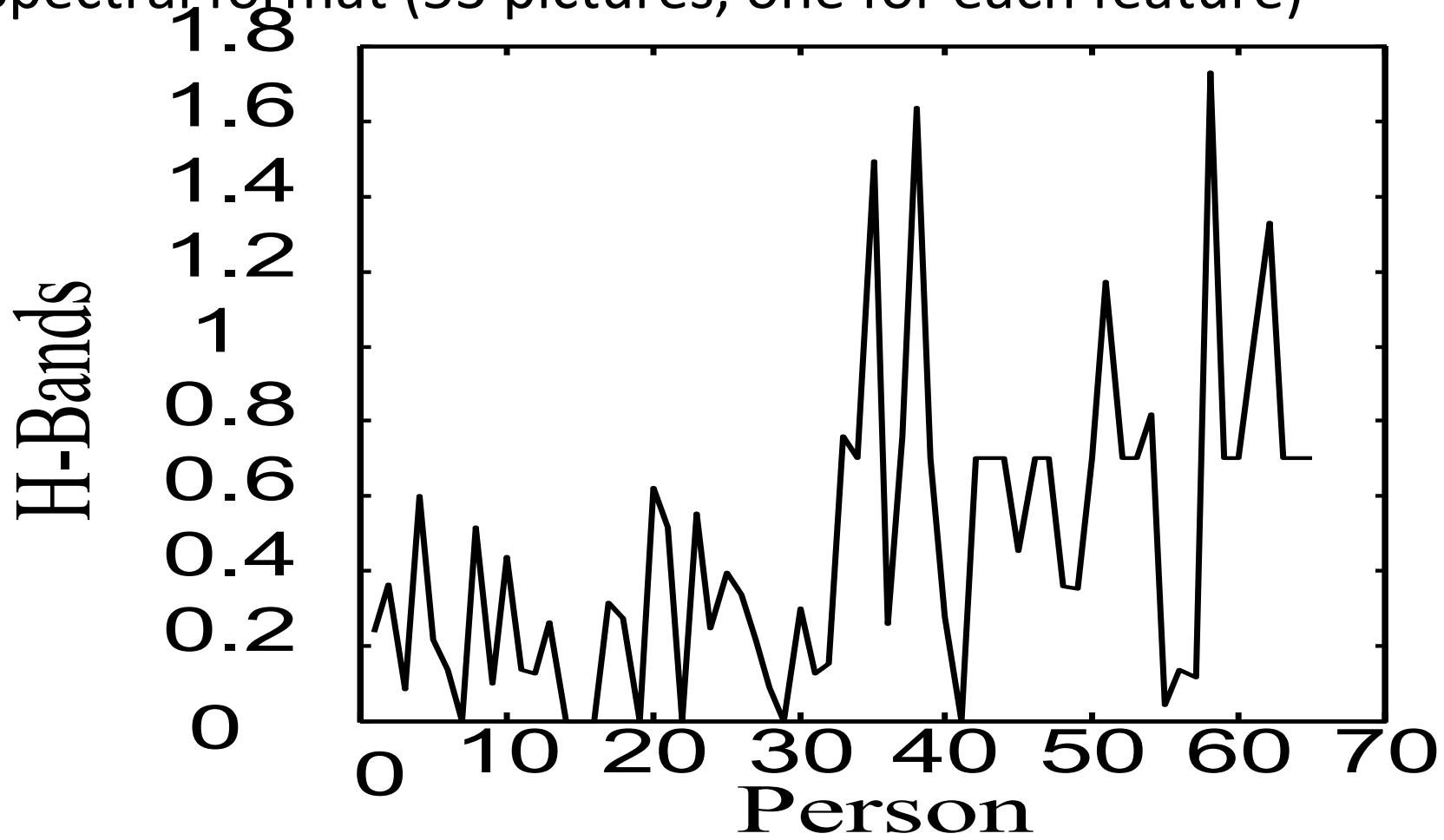
- Spectral format (65 pictures, one for each person)



Difficult to compare the different patients...

Data Visualization

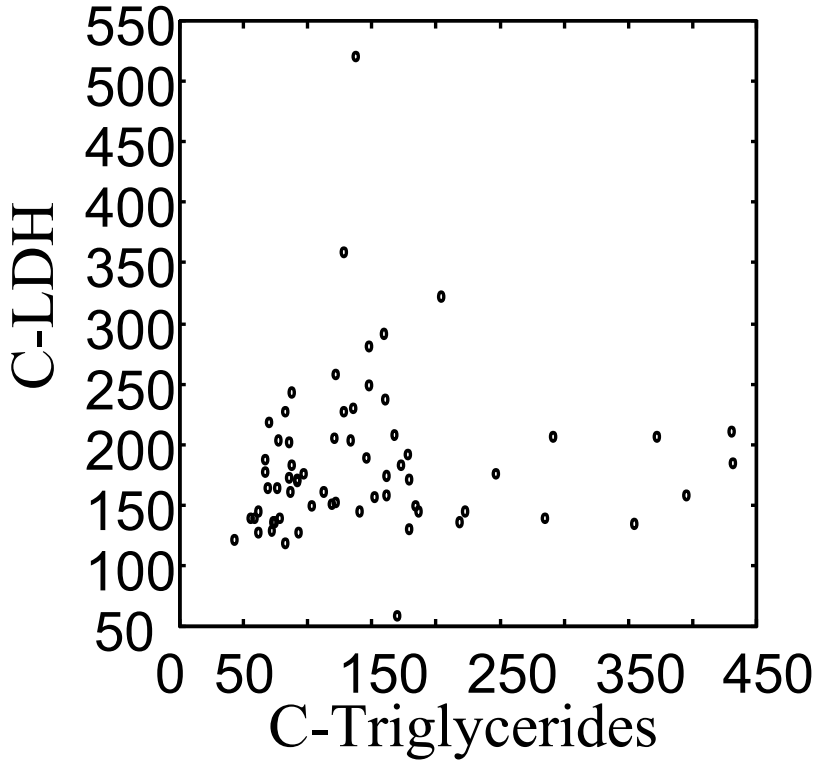
- Spectral format (53 pictures, one for each feature)



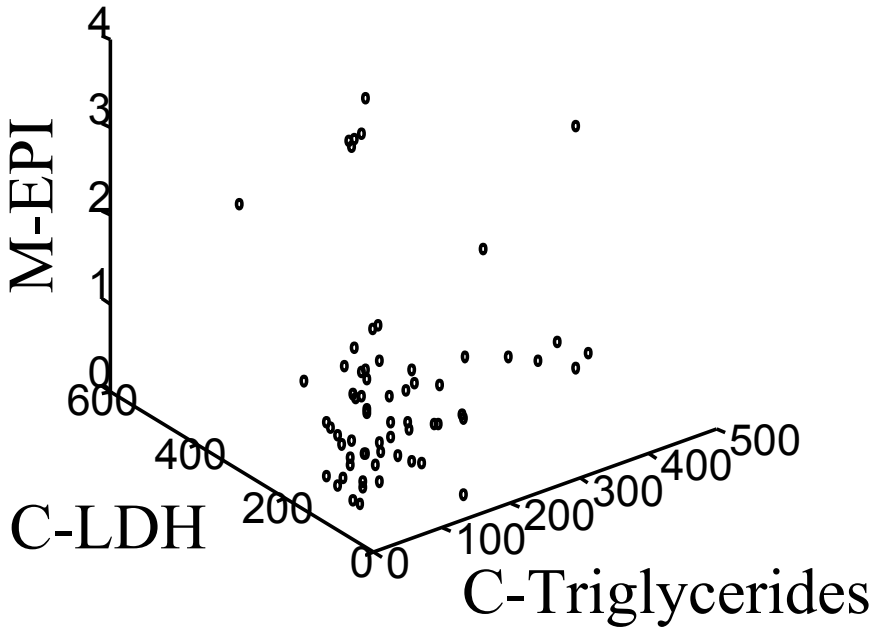
Difficult to see the correlations between the features...

Data Visualization

Bi-variate



Tri-variate



How can we visualize the other variables???

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

Data Visualization

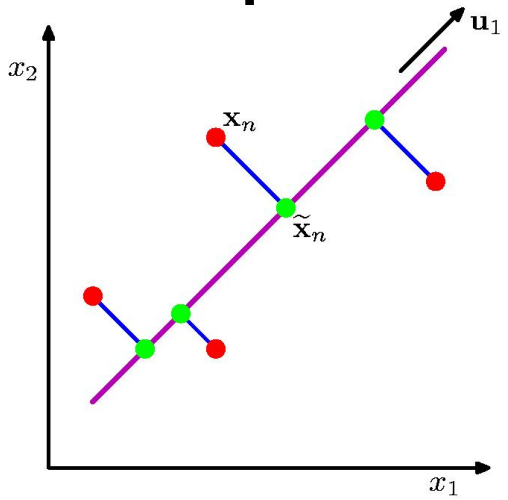
- Is there a representation better than the coordinate axes?

- Is it really necessary to show all the 53 dimensions?
  - … what if there are strong correlations between the features?

- How could we find
  the *smallest* subspace of the 53-D space that
  keeps the *most information* about the original data?

- A solution: **Principal Component Analysis**

Principle Component Analysis



PCA:

Orthogonal projection of data onto lower-dimension linear space that...

- maximizes variance of projected data (purple line)

- minimizes mean squared distance between
  - data point and
  - projections (sum of blue lines)

Principle Components Analysis

Idea:

- Given data points in a d-dimensional space, project into lower dimensional space while preserving as much information as possible
  - Eg, find best planar approximation to 3D data
  - Eg, find best 12-D approximation to 10^4-D data

- In particular, choose projection that minimizes *squared error* in reconstructing original data

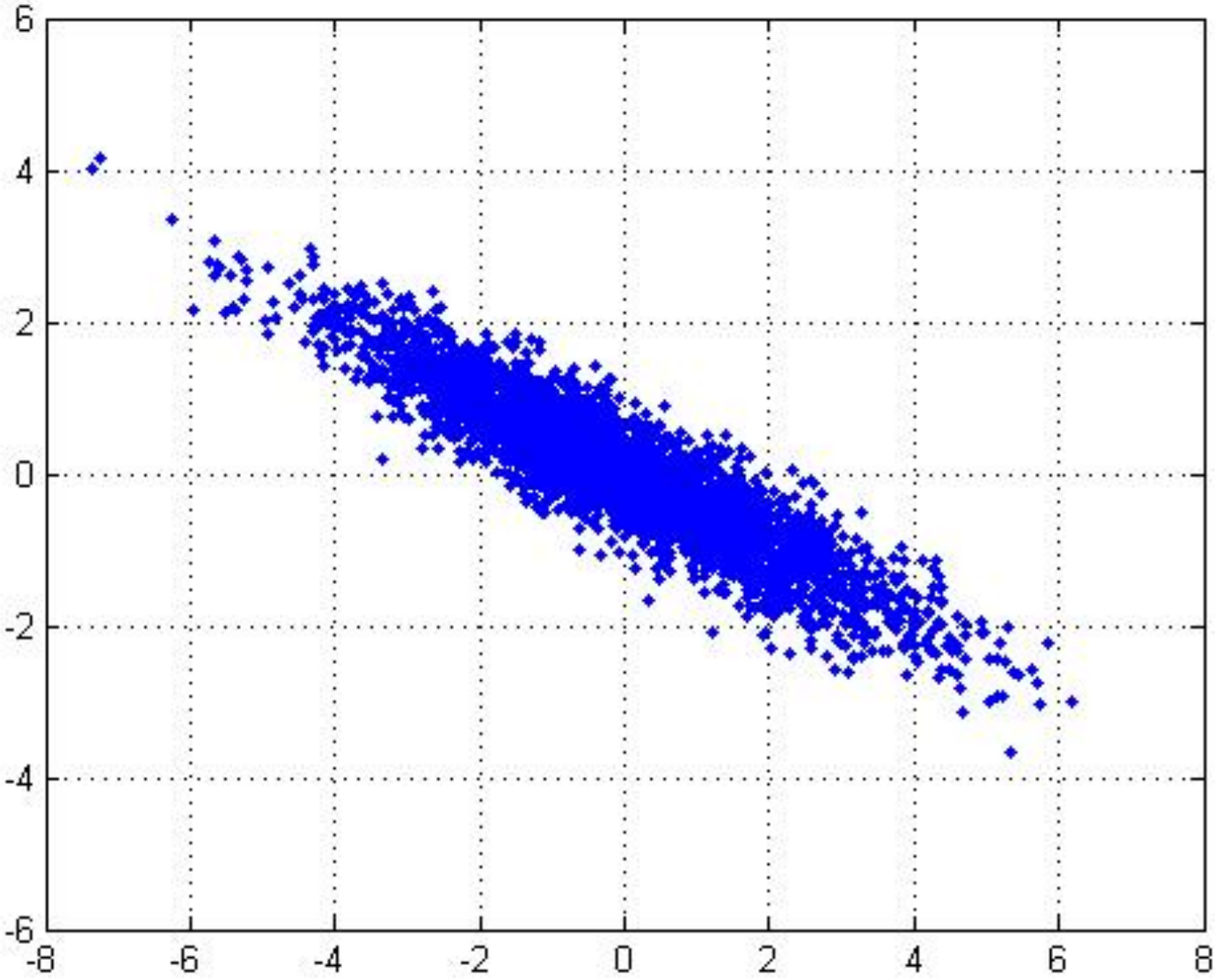
The Principal Components

- **Vectors** originating from the center of mass

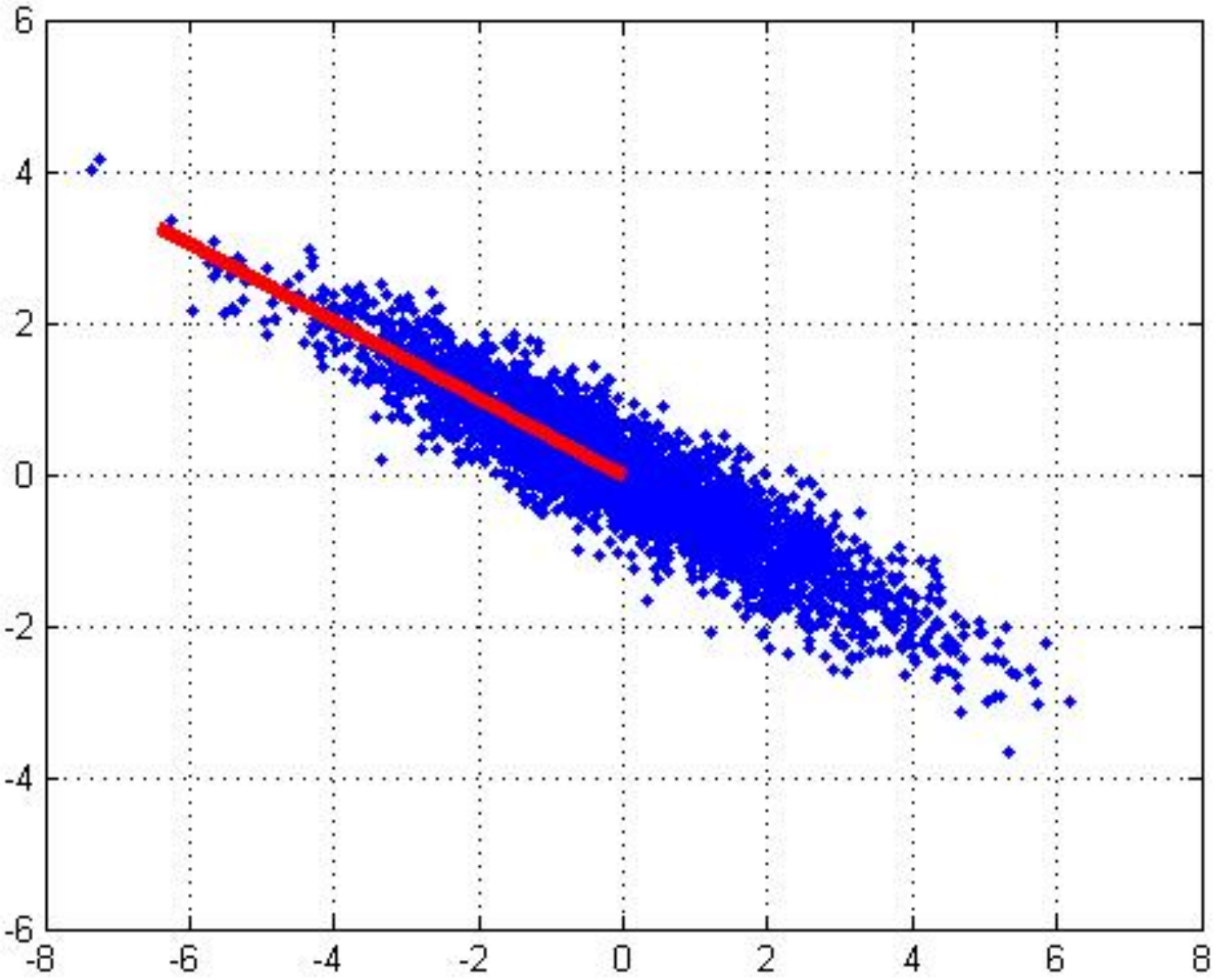
- Principal component #1 points in the direction of the **largest variance**.

- Each subsequent principal component…
  - is **orthogonal** to the previous ones, and
  - points in the directions of the **largest variance of the residual subspace**

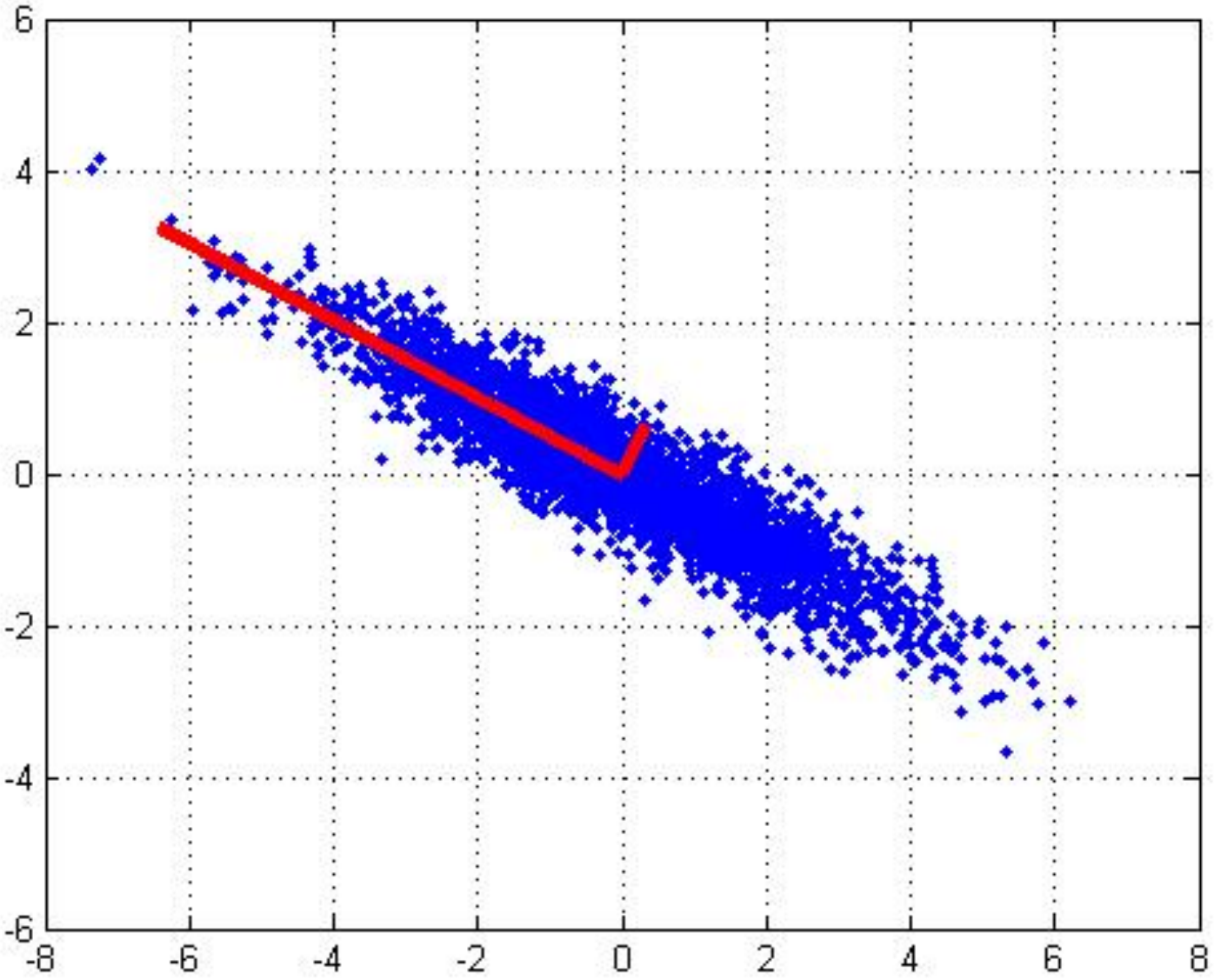
2D Gaussian dataset



1st PCA axis



2nd PCA axis



Data standardization

- In the Euclidean space, standardization of attributes is recommended so that all attributes can have equal impact on the computation of distances.
- Consider the following pair of data points
  - $\mathbf{x}_i$: (0.1, 20) and $\mathbf{x}_j$: (0.9, 720).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9 - 0.1)^2 + (720 - 20)^2} = 700.000457,$$

- The distance is almost completely dominated by (720-20) = 700.
- Standardize attributes: to force the attributes to have a common value range

Interval-scaled attributes

- Their values are real numbers following a linear scale.
  - The difference in Age between 10 and 20 is the same as that between 40 and 50.
  - The key idea is that intervals keep the same importance through out the scale
- Two main approaches to standardize interval scaled attributes, **range** and **z-score**. *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

Interval-scaled attributes (cont …)

- Z-score: transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f, is computed as follows

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

Z-score: $$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

PCA algorithm II (sample covariance matrix)

- Given data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, compute covariance matrix Σ

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \quad \text{where} \quad \bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$$

- **PCA** basis vectors = the eigenvectors of Σ

- Larger eigenvalue $\Rightarrow$ more important eigenvectors

PCA algorithm II

PCA algorithm($\mathbf{X}$, k): top k eigenvalues/eigenvectors

% $\underline{\mathbf{X}}$ = N × m data matrix,

% … each data point $\mathbf{x}_i$ = column vector, i=1..m

-
- $\mathbf{X} \leftarrow$ subtract mean $\underline{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$ from each column vector $\mathbf{x}_i$ in $\underline{\mathbf{X}}$
- $\Sigma \leftarrow \mathbf{X}\mathbf{X}^T$ … covariance matrix of $\mathbf{X}$
- $\{ \lambda_i, \mathbf{u}_i \}_{i=1..N}$ = eigenvectors/eigenvalues of Σ

  ... $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$
- Return $\{ \lambda_i, \mathbf{u}_i \}_{i=1..k}$

  % top k principle components

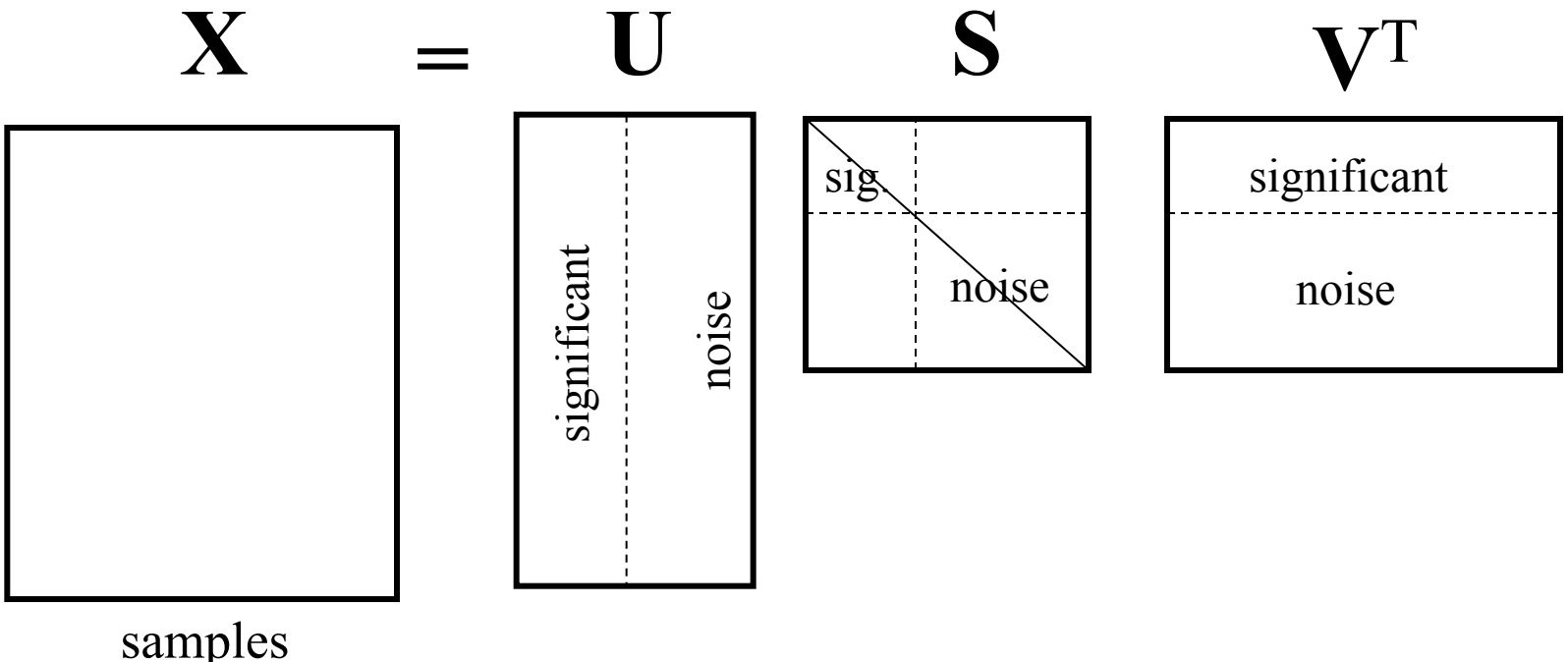
PCA algorithm III
(SVD of the data matrix)

Singular Value Decomposition of the **centered** data matrix **X**.

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{N \times m},$$

m: number of instances,
N: dimension

$$\mathbf{X}_{\text{features} \times \text{samples}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$



PCA algorithm III

- **Columns of U**
  - the principal vectors, $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$
  - orthogonal and has unit norm – so $U^T U = I$
  - Can reconstruct the data using linear combinations of $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$

- **Matrix S**
  - Diagonal
  - Shows importance of each eigenvector

- **Columns of V^T**
  - The coefficients for reconstructing the samples

PCA Example:
Face recognition

Challenge: Facial Recognition

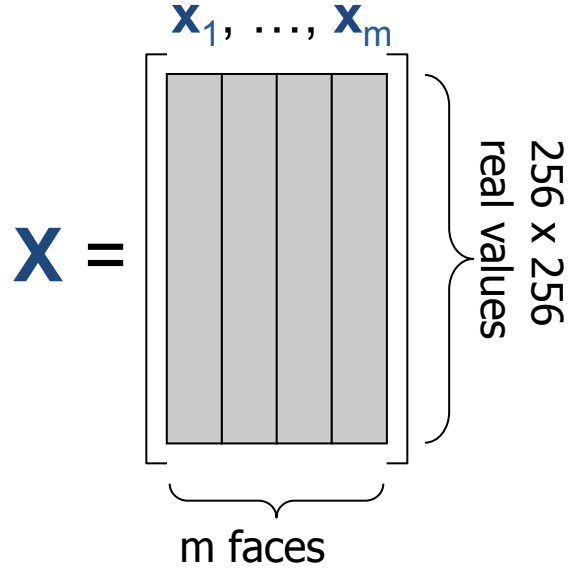
- Want to identify specific person, based on facial image
- Robust to glasses, lighting,…

  ⇒ Can't just use the given 256 x 256 pixels

Applying PCA: Eigenfaces

Method A: Build a PCA subspace for each person and check which subspace can reconstruct the test image the best

Method B: Build one PCA database for the whole dataset and then classify based on the weights.



- Example data set: Images of faces
  - Famous Eigenface approach
    [Turk & Pentland], [Sirovich & Kirby]
- Each face $\mathbf{x}$ is …
- 256 × 256 values (luminance at location)
  - $\mathbf{x}$ in $\Re^{256\times256}$ (view as 64K dim vector)
- Form $\mathbf{X} = [\, \mathbf{x}_1, \ldots, \mathbf{x}_m \,]$ **centered** data mtx
- Compute $S = \mathbf{X}\mathbf{X}^T$
- Problem: S is 64K × 64K … HUGE!!!

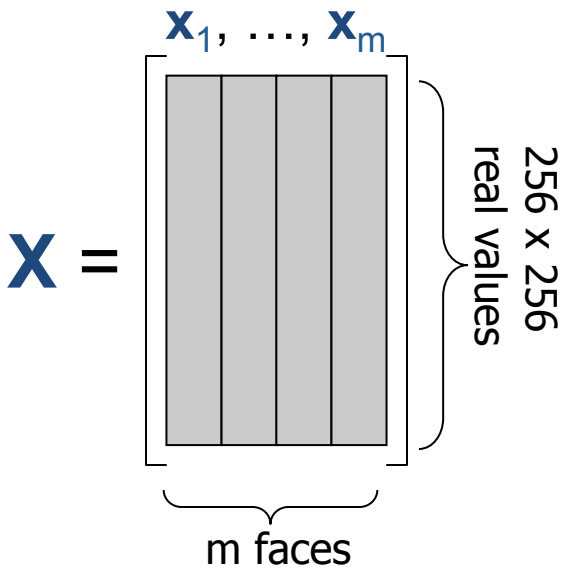
Computational Complexity

- Suppose m instances, each of size N
  - Eigenfaces: m=500 faces, each of size N=64K
- Given N×N covariance matrix Σ, can compute
  - all N eigenvectors/eigenvalues in $O(N^3)$
  - first k eigenvectors/eigenvalues in $O(k\ N^2)$

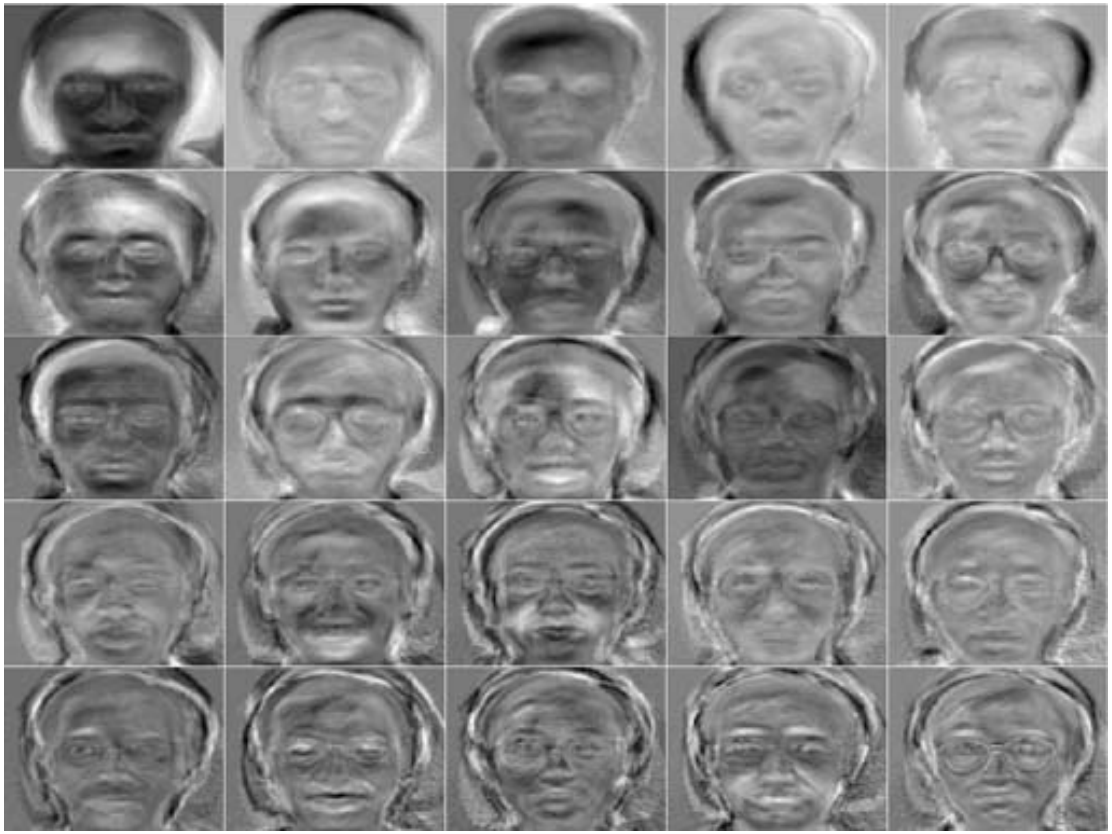
- But if N=64K, EXPENSIVE!

A Clever Workaround

- Note that $m << 64K$
- Use $\mathbf{L}=\mathbf{X}^T\mathbf{X}$ instead of $S=\mathbf{X}\mathbf{X}^T$
- If $\mathbf{v}$ is eigenvector of $\mathbf{L}$ then $\mathbf{X}\mathbf{v}$ is eigenvector of S



Principle Components (Method B)



Reconstructing… (Method B)

- … faster if train with…
  - only people w/out glasses
  - same lighting conditions

Shortcomings

- Requires carefully controlled data:
  - All faces centered in frame
  - Same size
  - Some sensitivity to angle
- Alternative:
  - "Learn" one set of PCA vectors for each angle
  - Use the one with lowest error

- Method is completely knowledge free
  - (sometimes this is good!)
  - Doesn't know that faces are wrapped around 3D objects (heads)
  - Makes no effort to preserve class distinctions

How to choose a clustering algorithm

- Clustering research has a long history. A vast collection of algorithms are available.
  - We only introduced several main algorithms.
- Choosing the "best" algorithm is a challenge.
  - Every algorithm has limitations and works well with certain data distributions.
  - It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any "ideal" structure or distribution required by the algorithms.
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

Choose a clustering algorithm (cont …)

- Due to these complexities, the common practice is to
  - run several algorithms using different distance functions and parameter settings, and
  - then carefully analyze and compare the results.
- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.
- Clustering is highly application dependent and to certain extent subjective (personal preferences).

Summary: Unsupervised learning, so far

- Unsupervised vs. supervised learning: pros and challenges
- Clustering
- Some common approaches: K-means, PCA
- Real-world tips and tricks
  - Data scaling
  - When to use which algorithm
- Next Homework: supervised and unsupervised learning

In Class Assignment: Regression

- This question uses the “longley” data set which is part of the datasets package
- An annual multiple time series from 1947 to 1962 with 4 variables.

In Class Assignment: Regression

- Produce some numerical and graphical summaries of the data. Do there appear to be any patterns?
- Use the full data set to perform a linear regression with “employment” as the response and other variables as the predictors
- Compute the confusion matrix and overall fraction of correct predictions
- Experiment with different sets of predictor variables including lagged variables – are any statistically significant?

In Class Assignment: Regression vs KNN

- Repeat implmentation of the regression model, now only using data from 1947 to 1960 as the training data period
- Assess the overall fraction of correct predictions for the held out data set (1961-1962)
- Repeat the above using KNN with K = 1
- Which method provides better results? Experiment with different values of K, and combinations of predictors in the regression model