Caltech

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 12:

Clustering & Dimensionality Reduction

Miniproject 2

- Train HMM on dataset of sonnets
  - No need to implement your own
- Generate new sonnets
  - I.e., sample from trained HMM
- Work in teams

Topic Overview

Supervised Learning

Linear Models

Overfitting

Loss Functions

Non-Linear Models

Learning Algorithms & Optimization

Probabilistic Modeling

Unsupervised Learning

Today
(Unsupervised Learning)

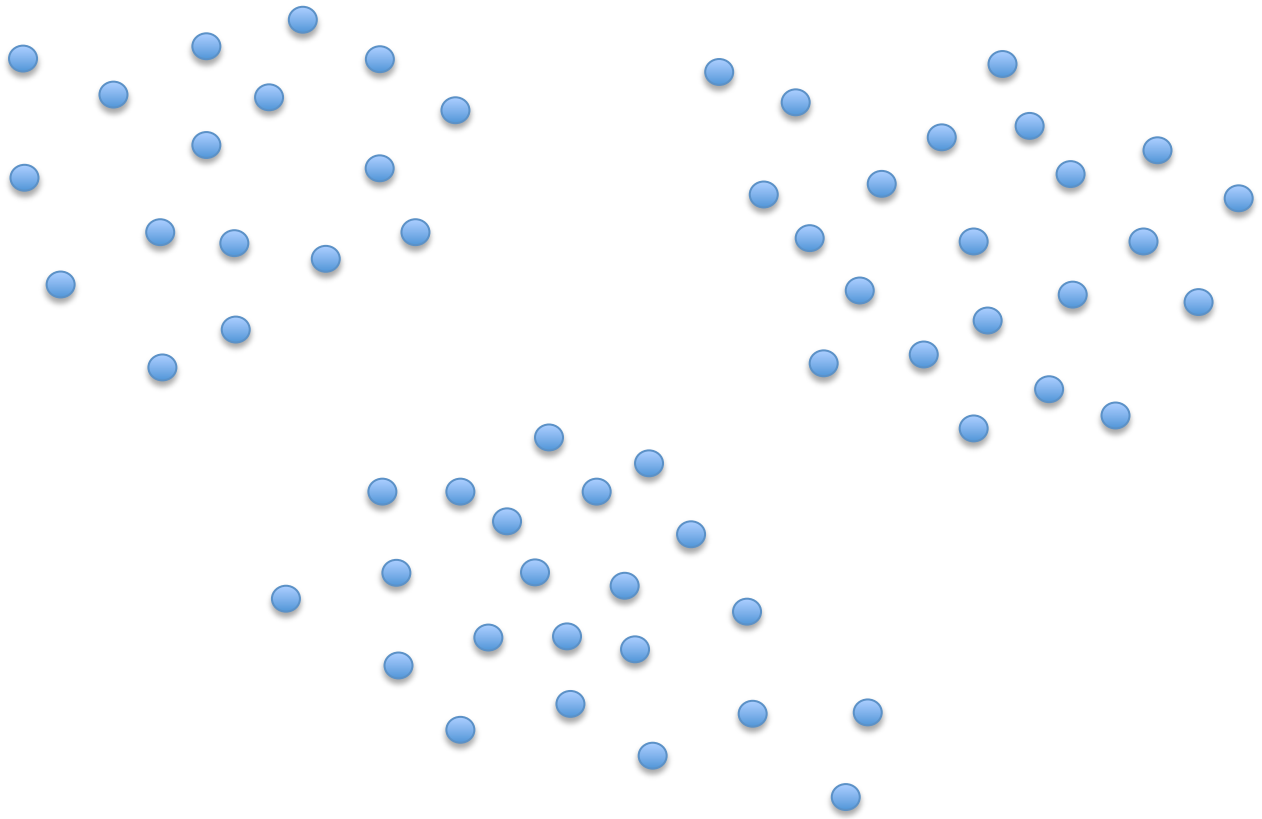
- Clustering

- Dimensionality Reduction
  - Matrix Factorization

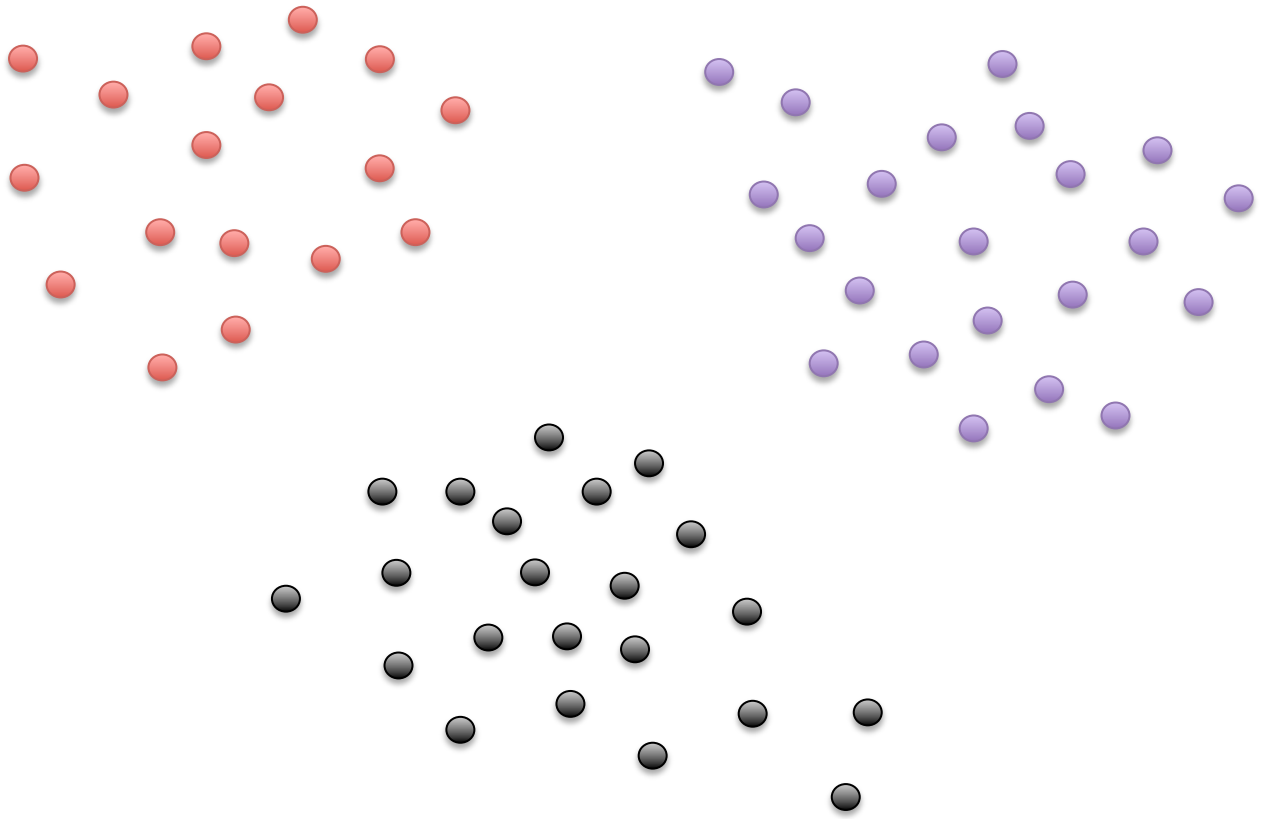
What is Clustering?

- Clustering is the process of grouping data points into “clusters”.
- High intra-cluster similarity
- Low inter-cluster similarity

Example



Example



Unsupervised Learning

- **Given:** unlabeled data:
  - Only input features
  - No labels

$$S = \{x_i\}_{i=1}^{N}$$

- **Goal:** find hidden structure/patterns
  - E.g., hidden structure is a clustering of data
  - Previously: generative model of P(x)
  - I.e., a low dimensional summary of the data

Why is Clustering Useful?

- Clustering is a "summary" of data
  - Can just inspect cluster centers
  - Or inspect a few data points per cluster

Images Related to “Pluto”

Each Row is a Cluster

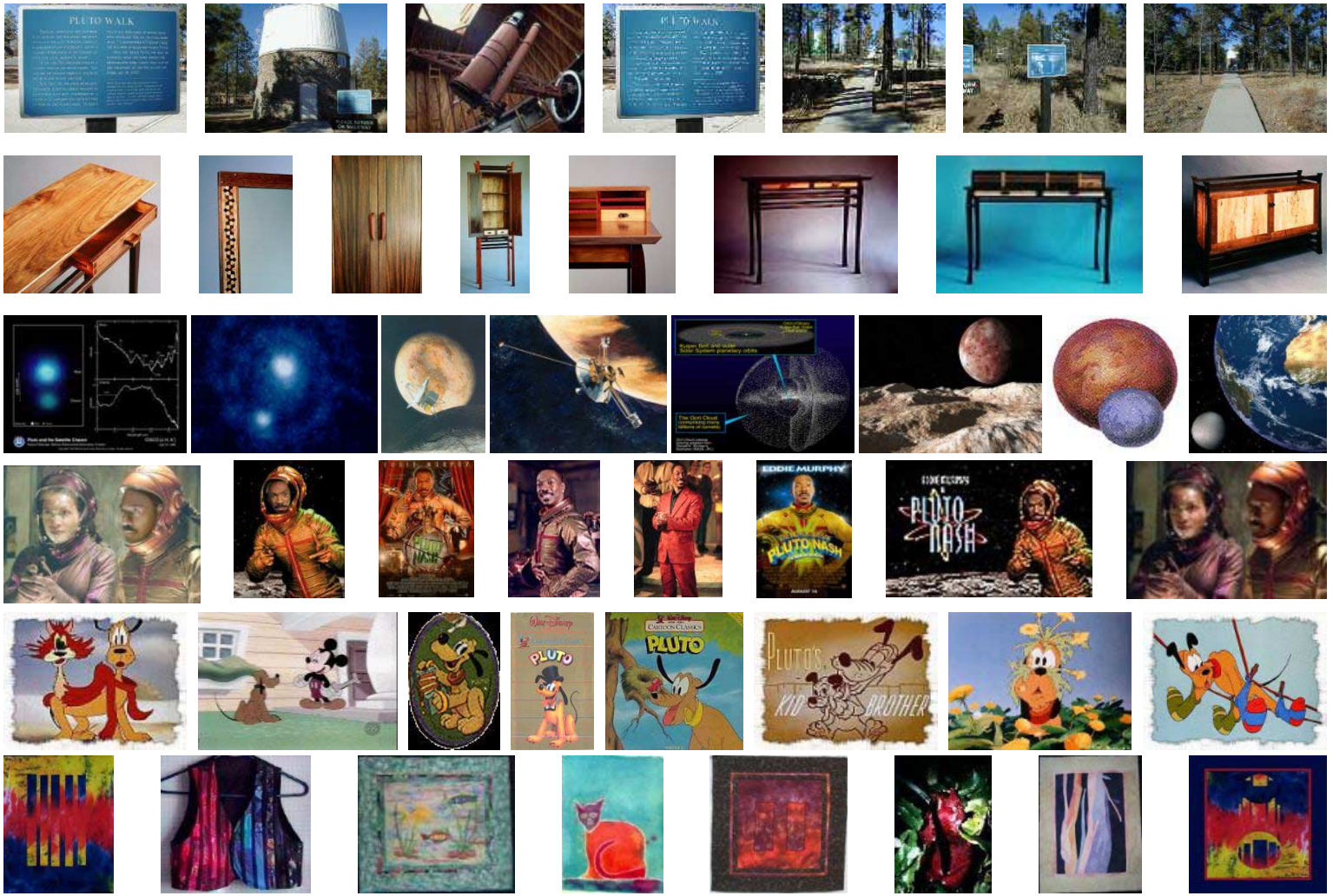


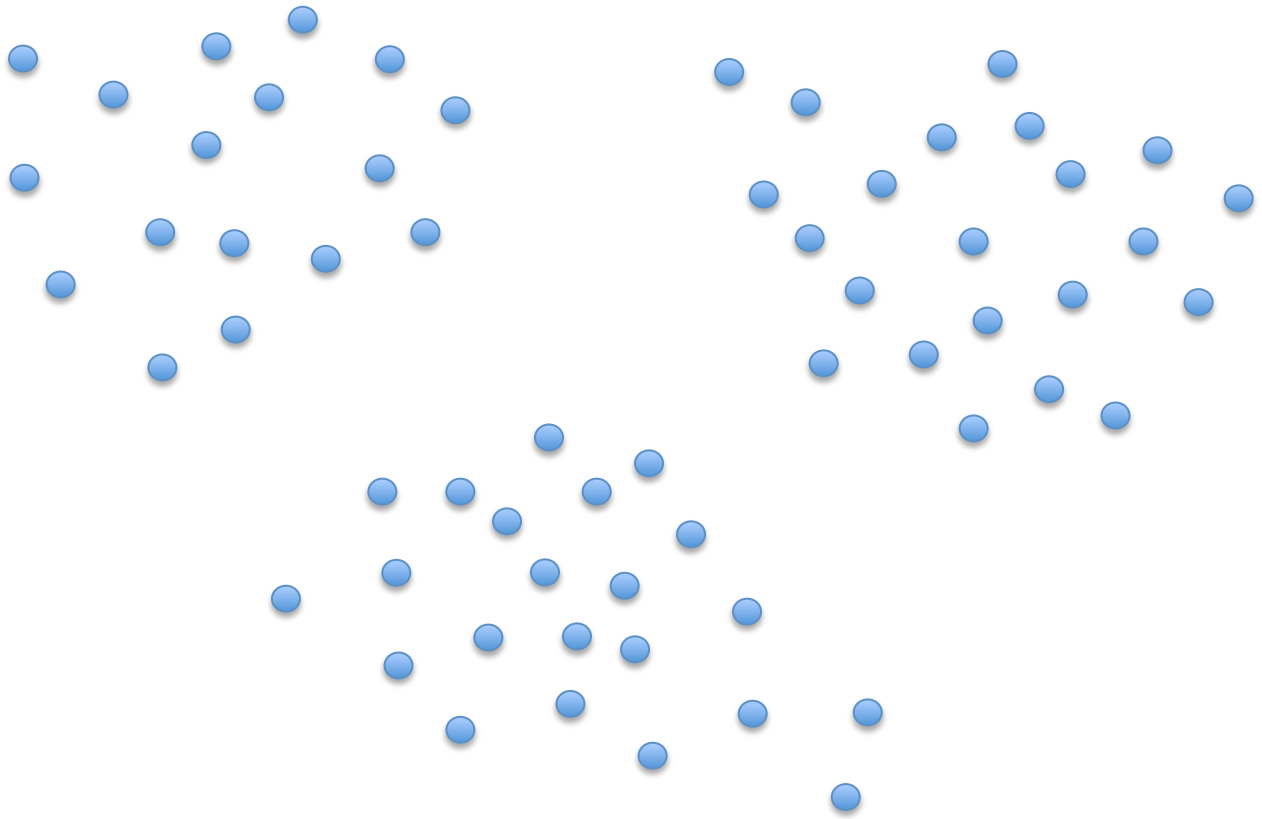
Image Source: http://research.microsoft.com/en-us/people/jrwen/mm04.pdf

Why is Clustering Useful?

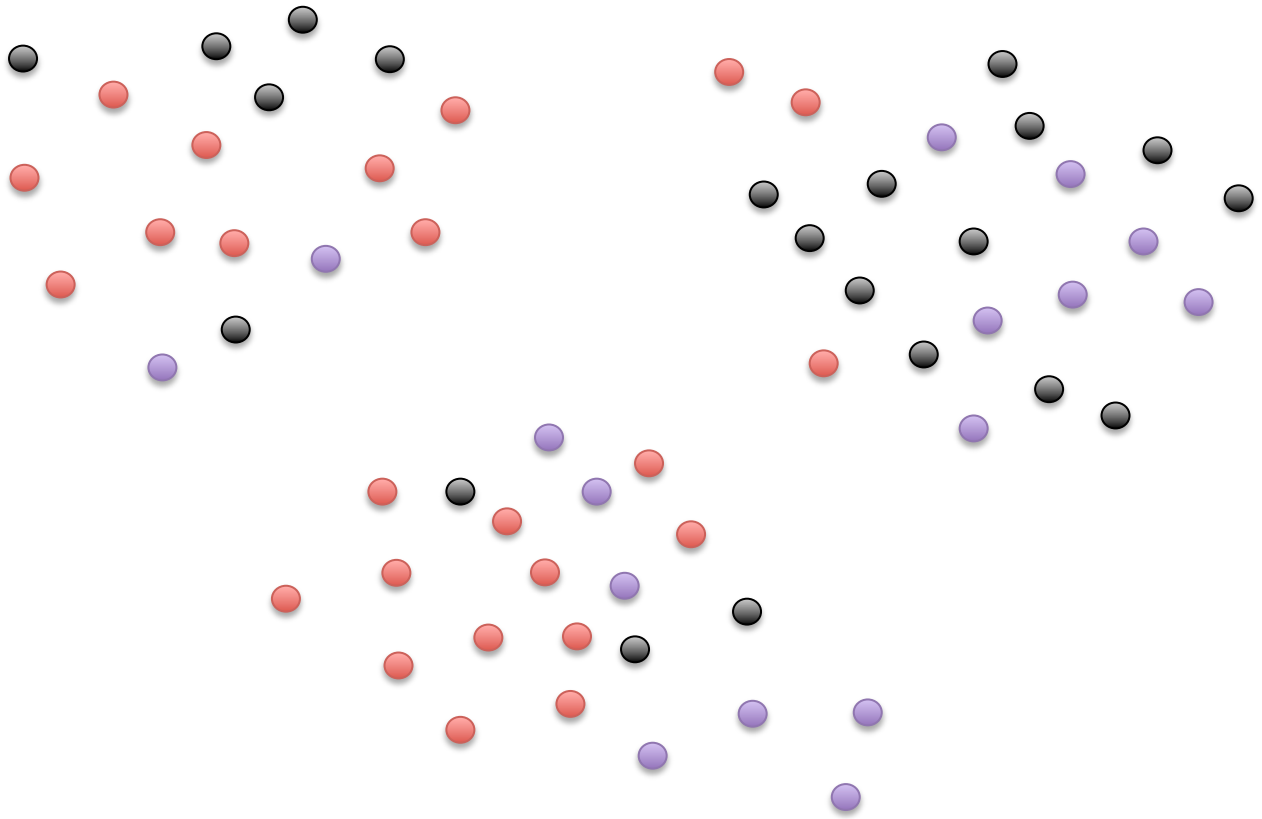
- Clustering is a "summary" of data
  - Can just inspect cluster centers
  - Or inspect a few data points per cluster

- Compact pre-processing of data before supervised training

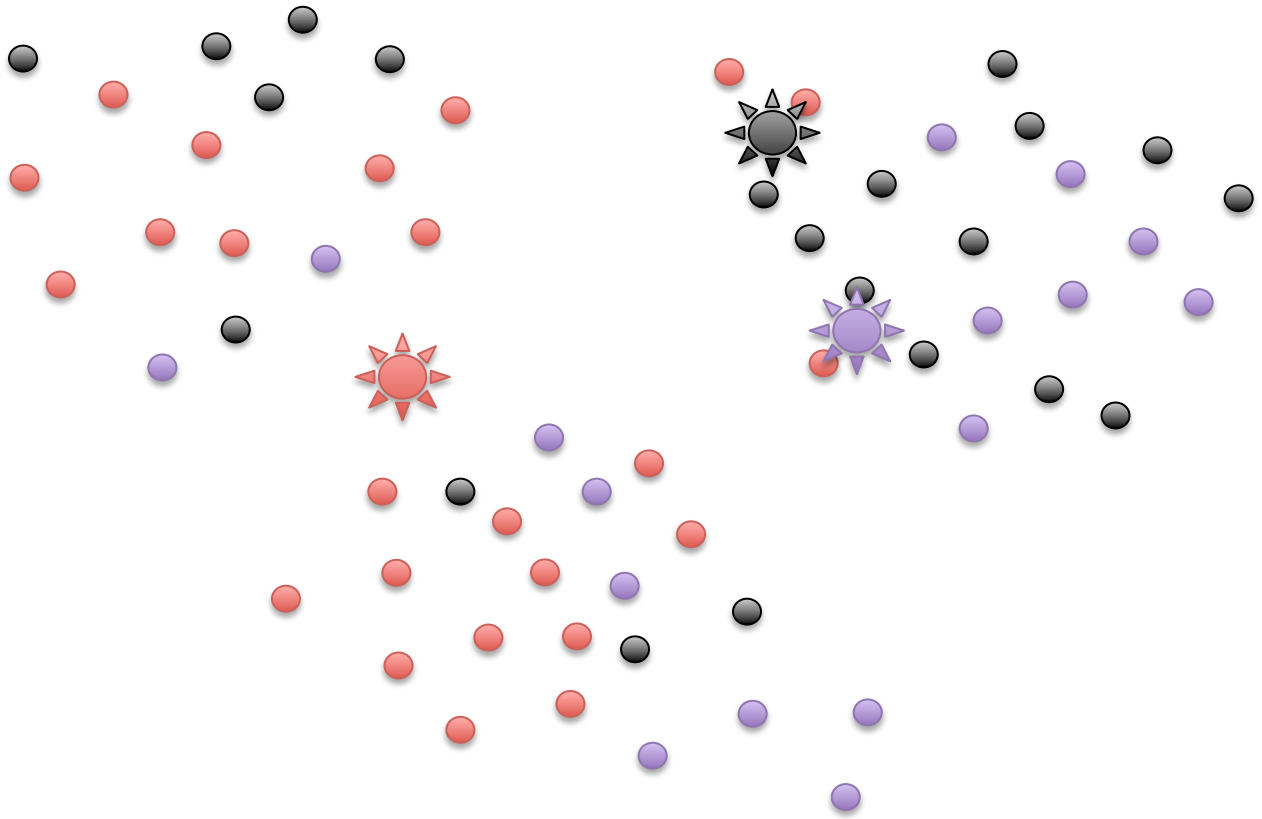
Centroid Based Clustering
(K-Means)



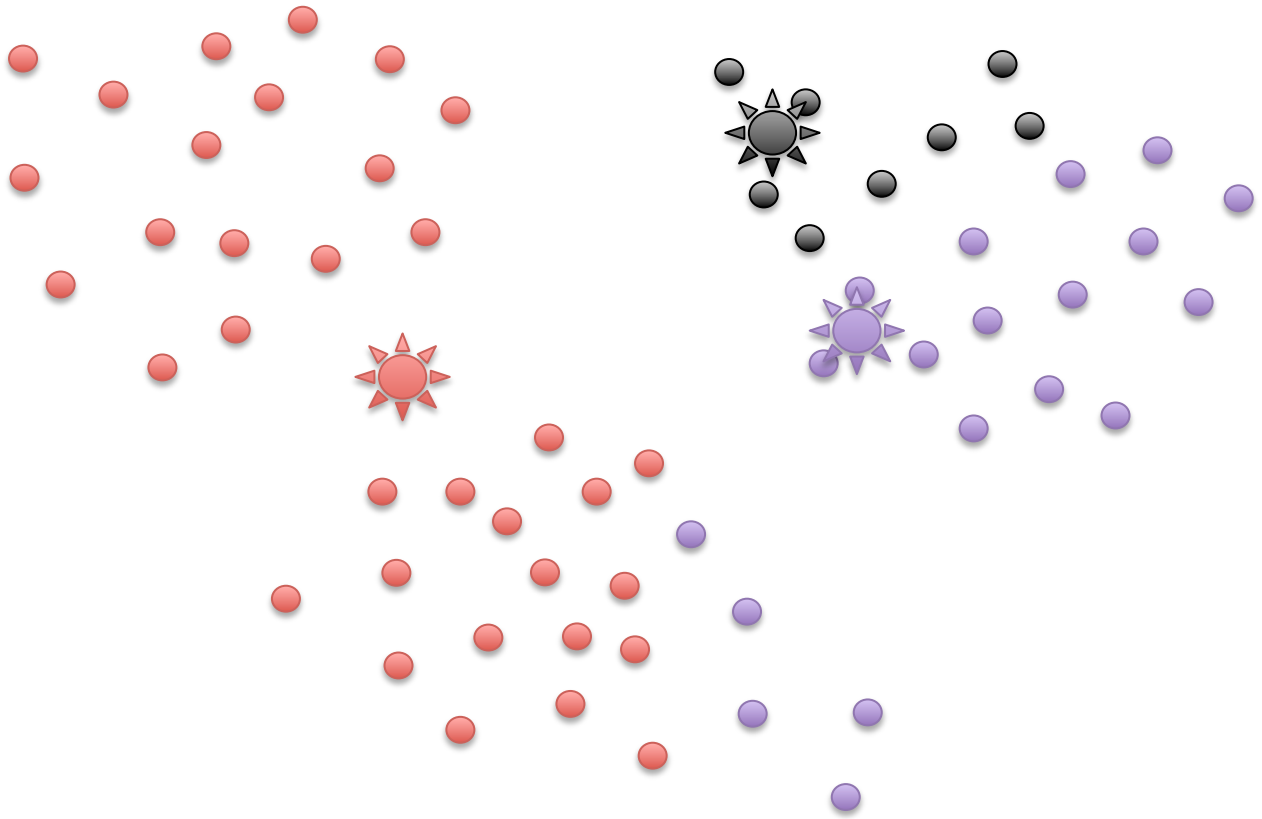
Centroid Based Clustering
(K-Means)



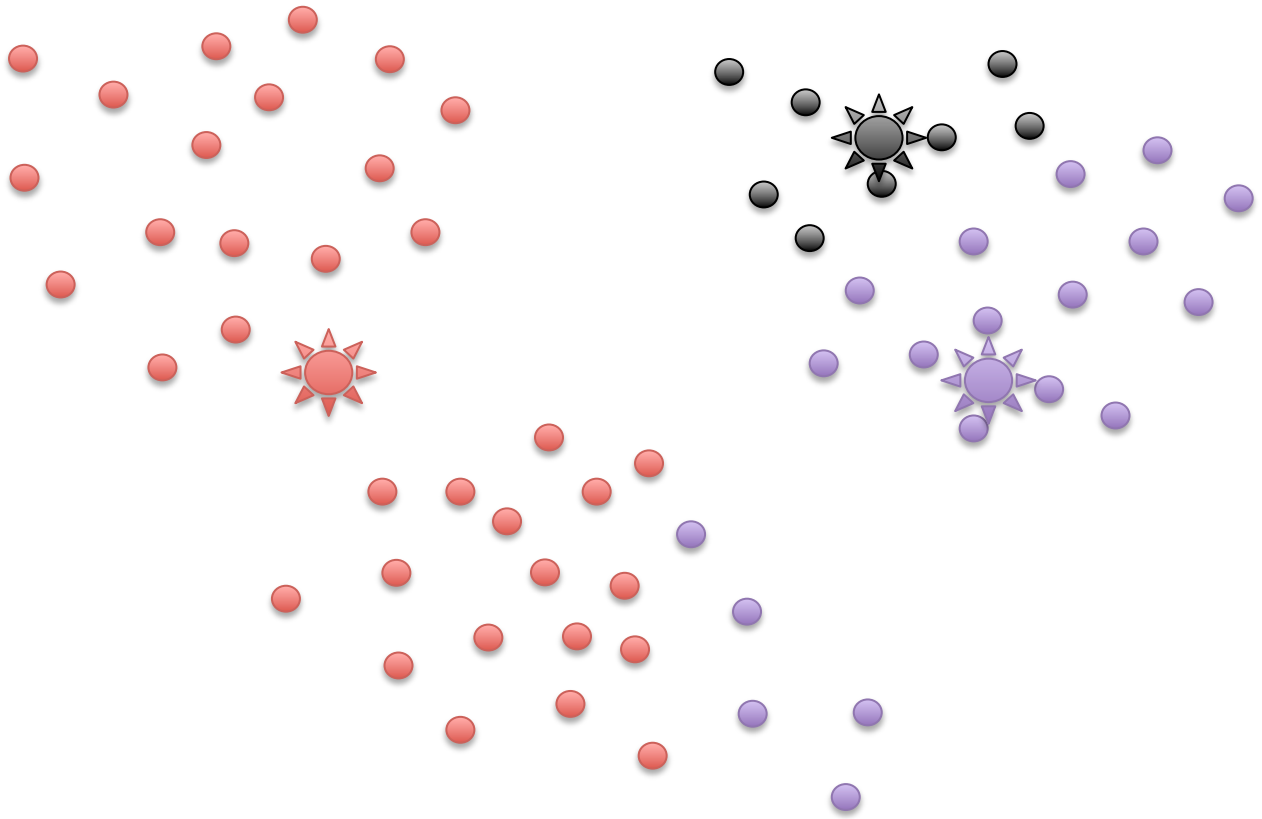
Centroid Based Clustering
(K-Means)



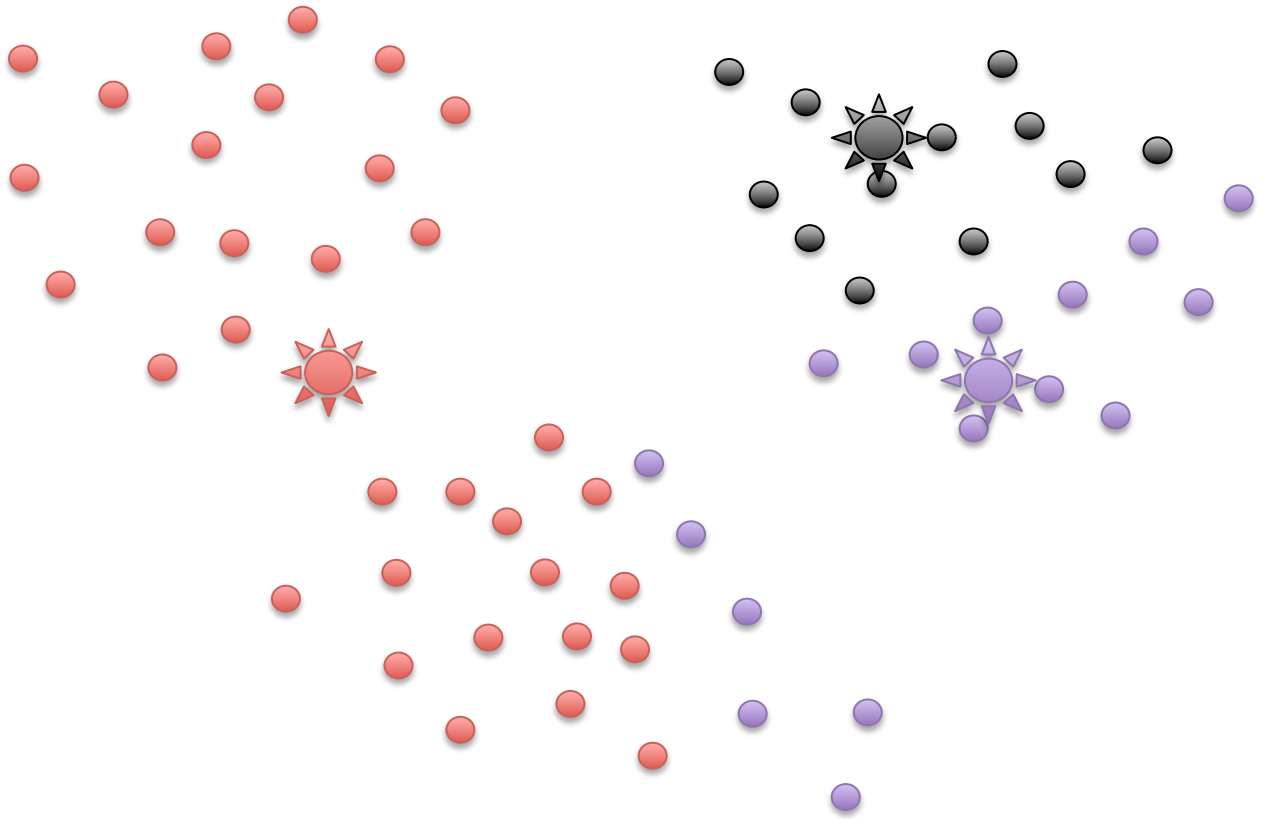
Centroid Based Clustering
(K-Means)



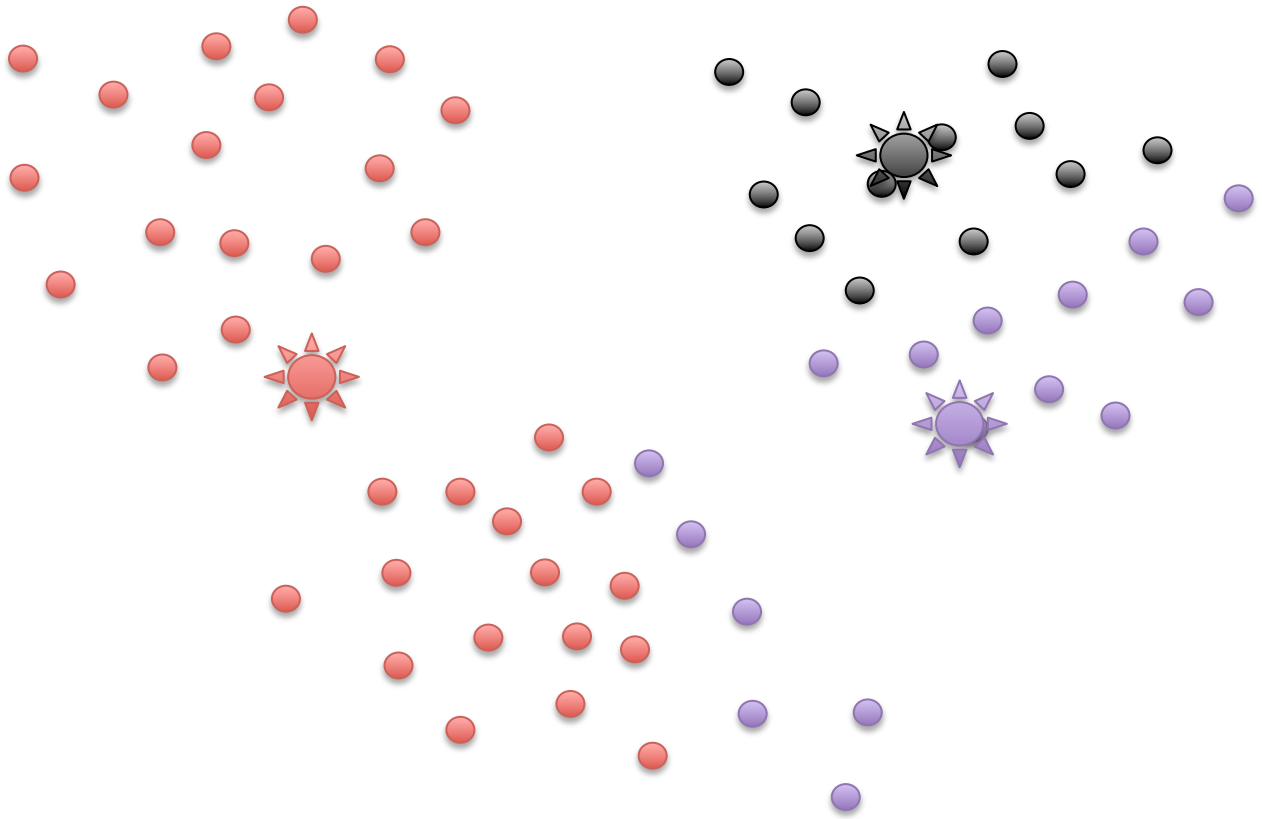
Centroid Based Clustering
(K-Means)



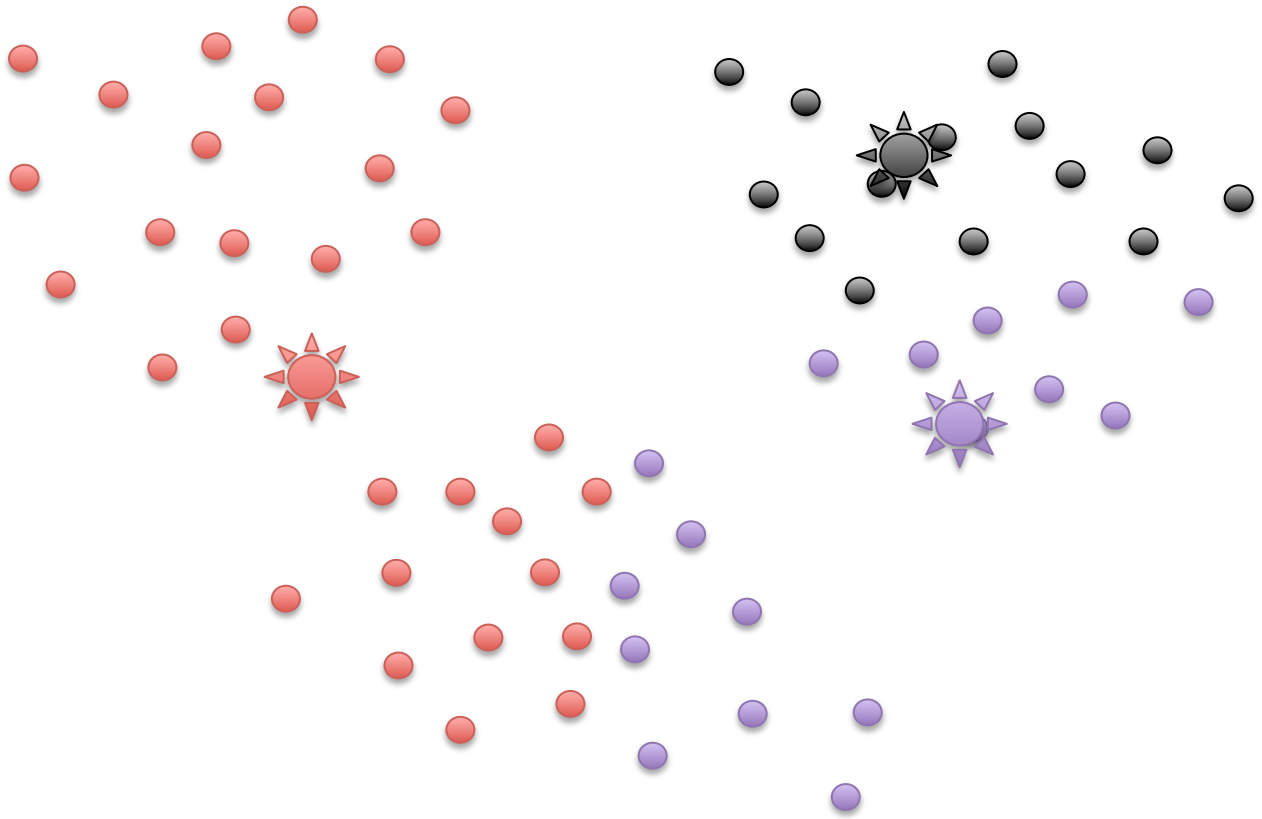
Centroid Based Clustering
(K-Means)



Centroid Based Clustering
(K-Means)

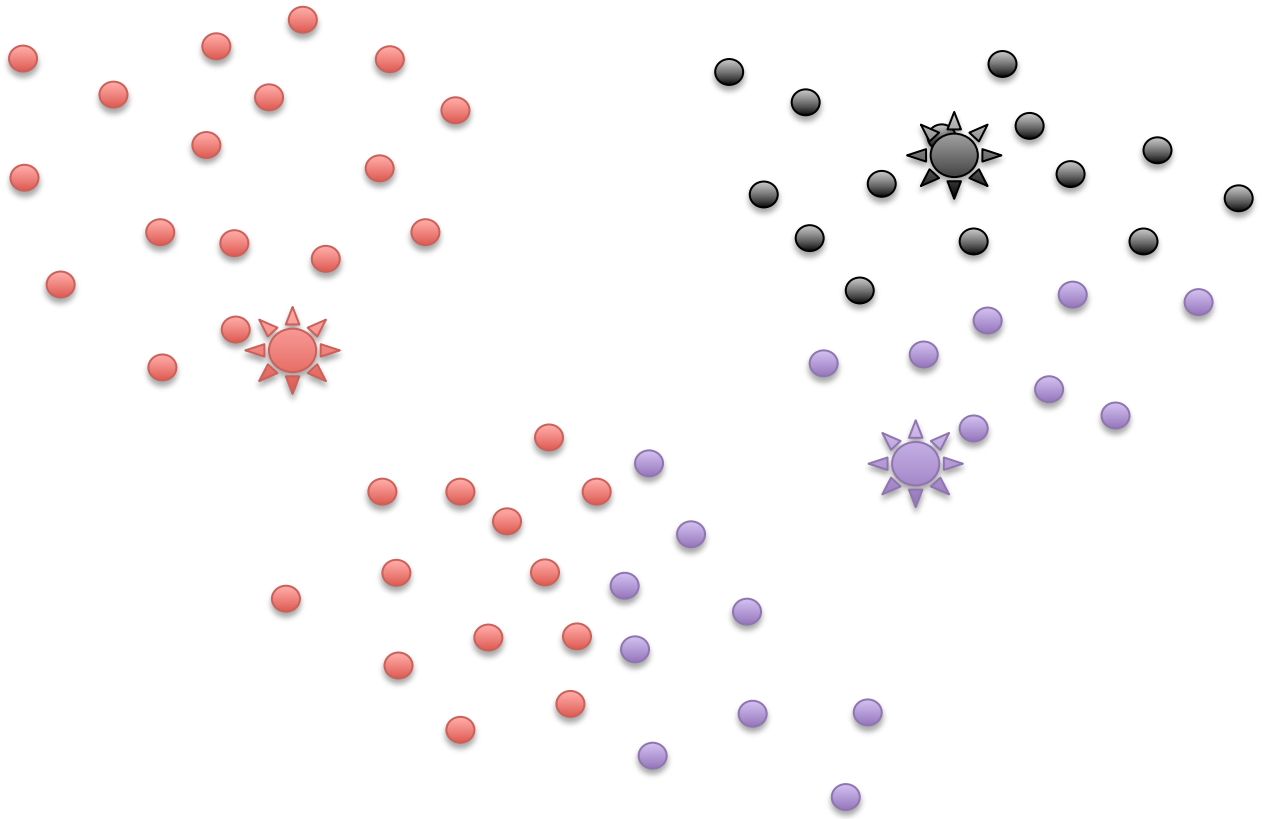


Centroid Based Clustering
(K-Means)

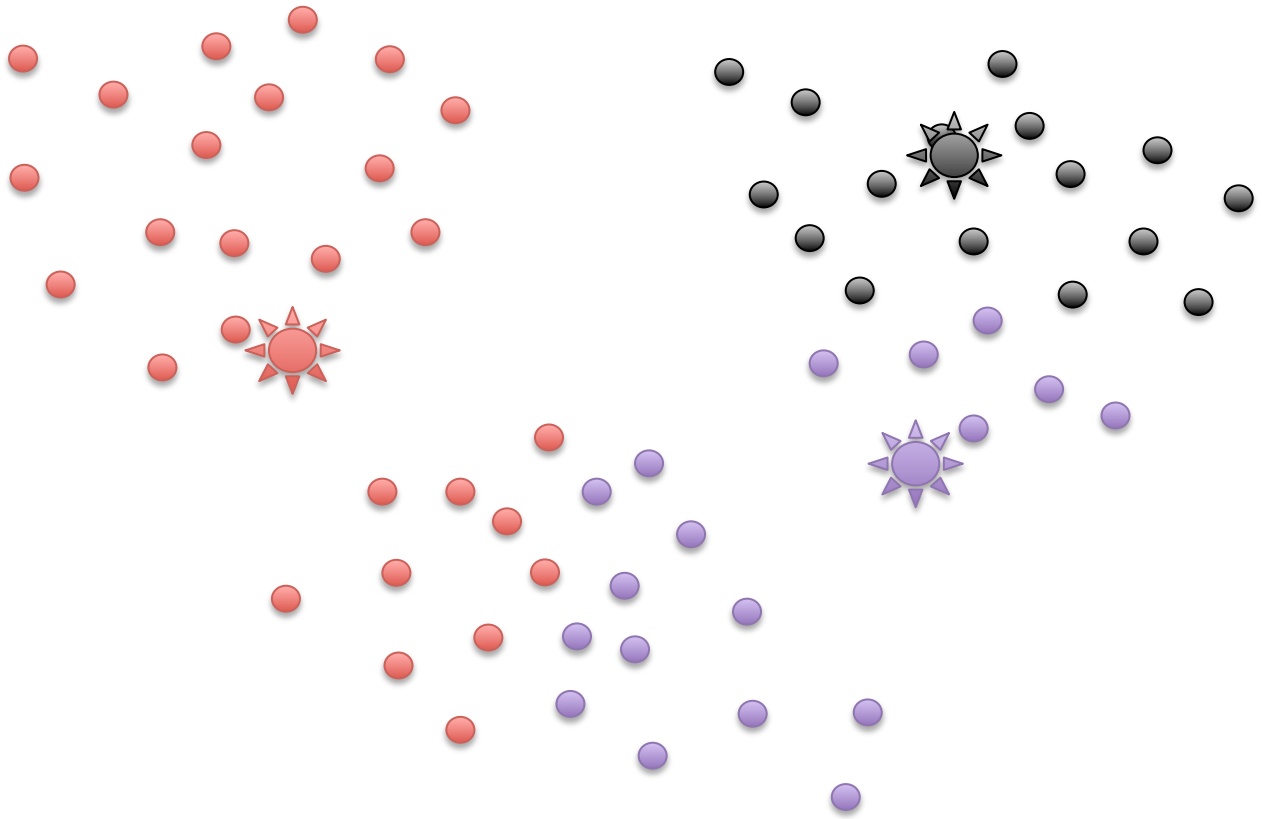


Centroid Based Clustering

(K-Means)

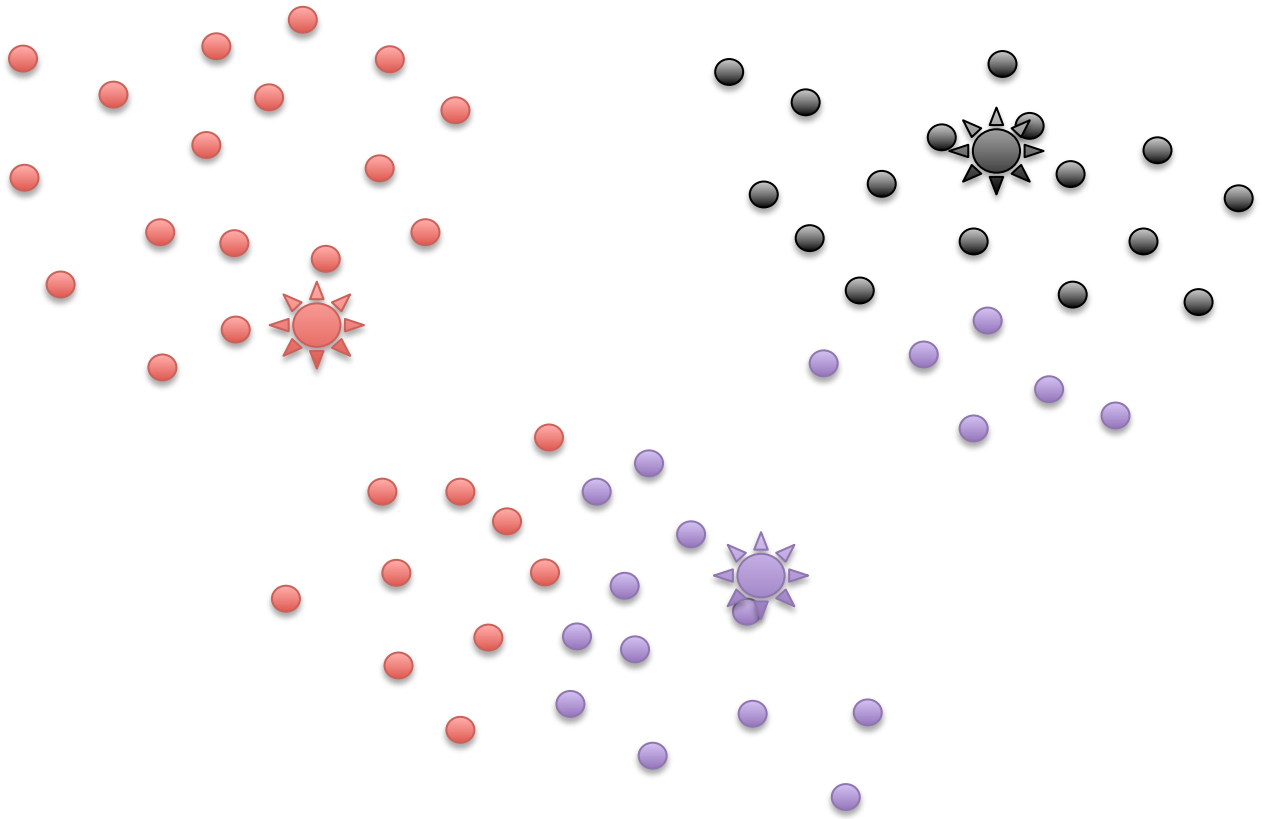


Centroid Based Clustering
(K-Means)

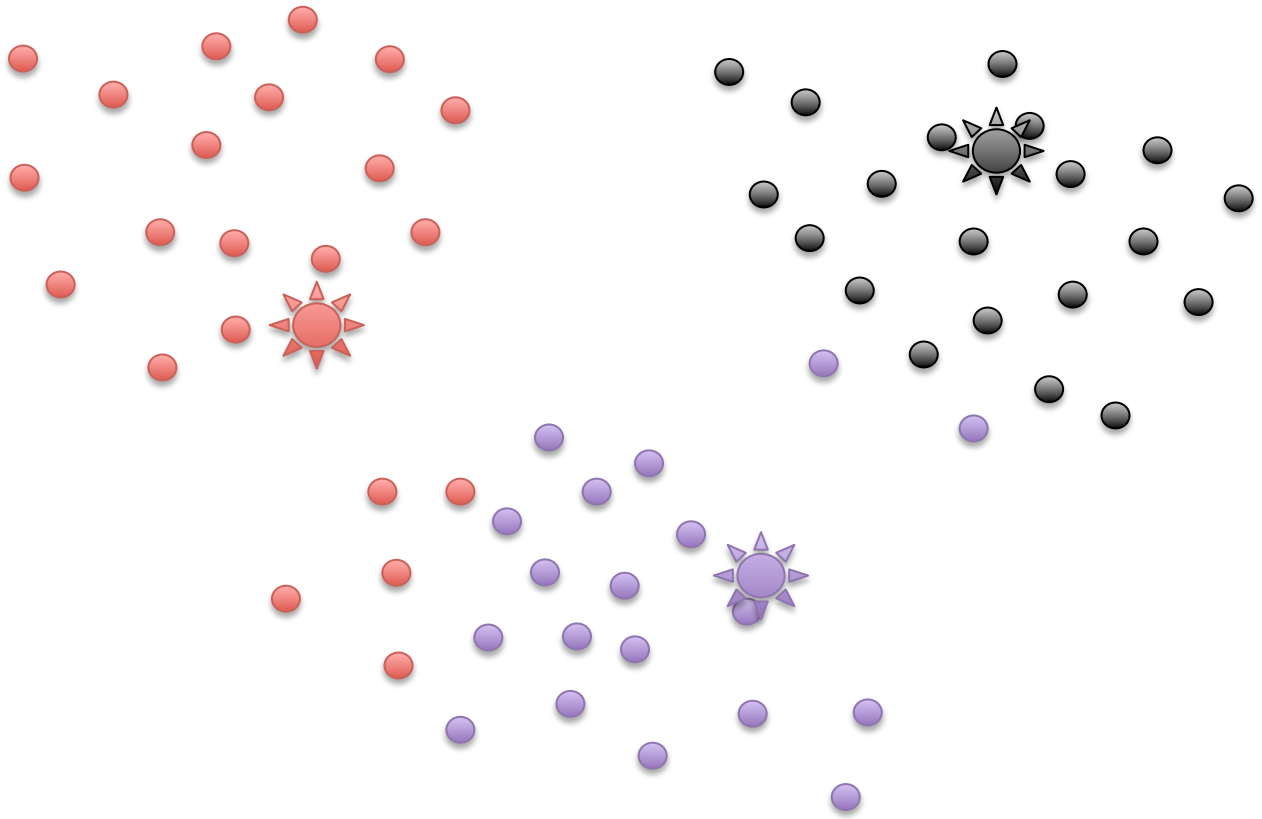


Centroid Based Clustering

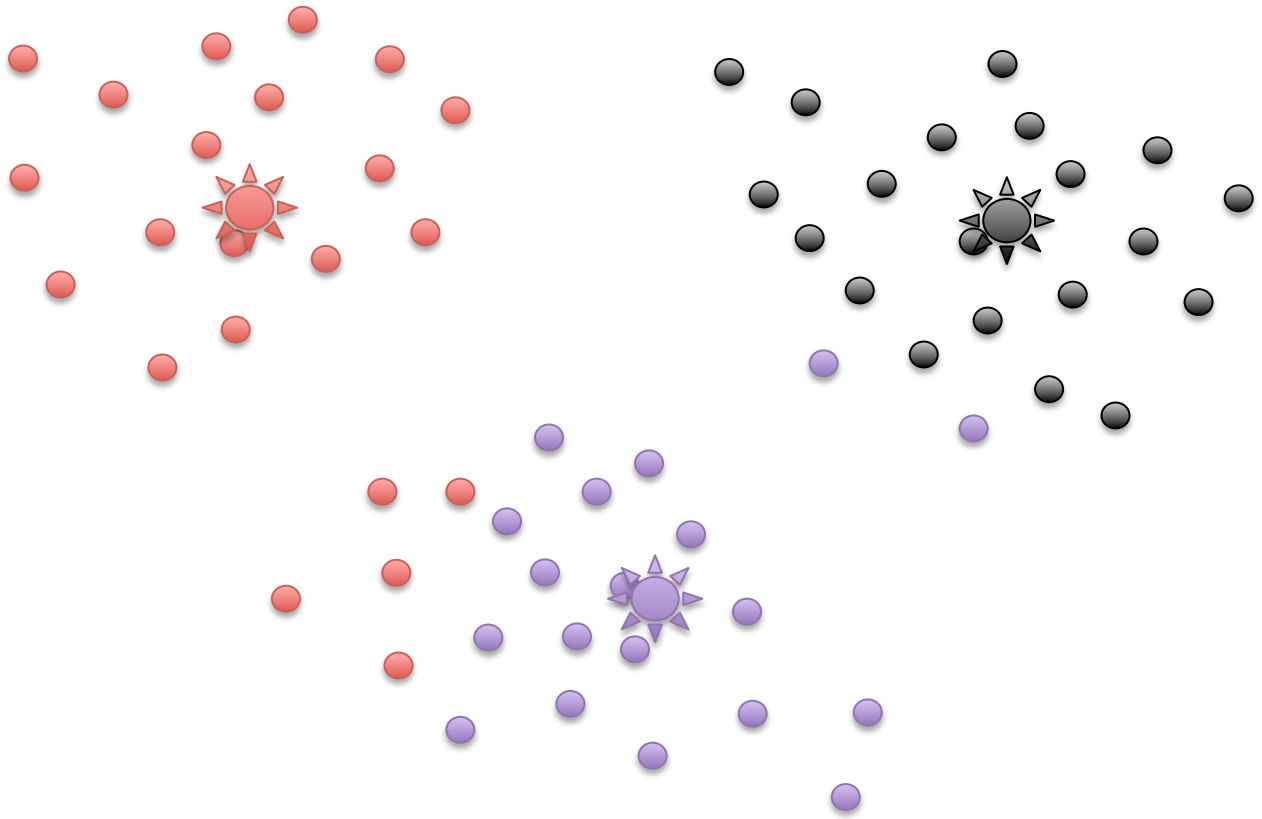
(K-Means)



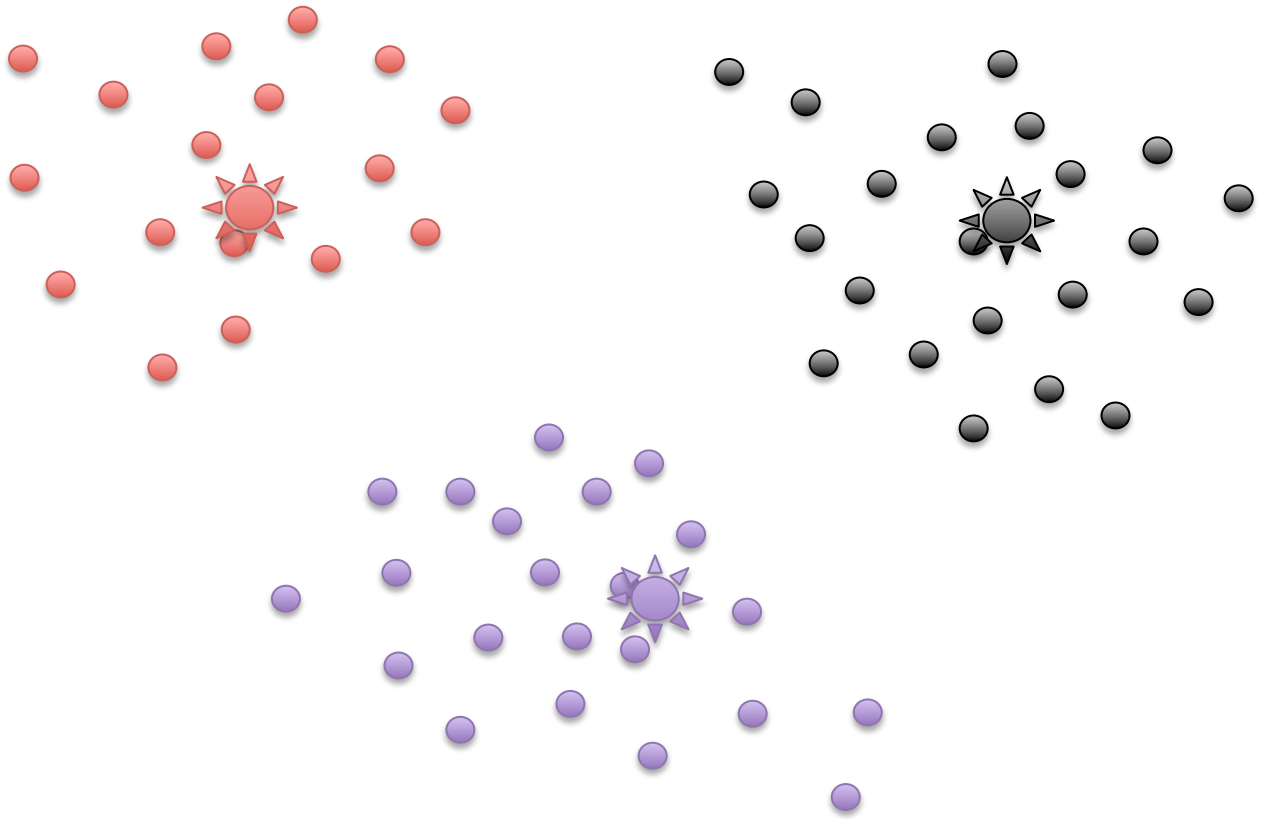
Centroid Based Clustering
(K-Means)



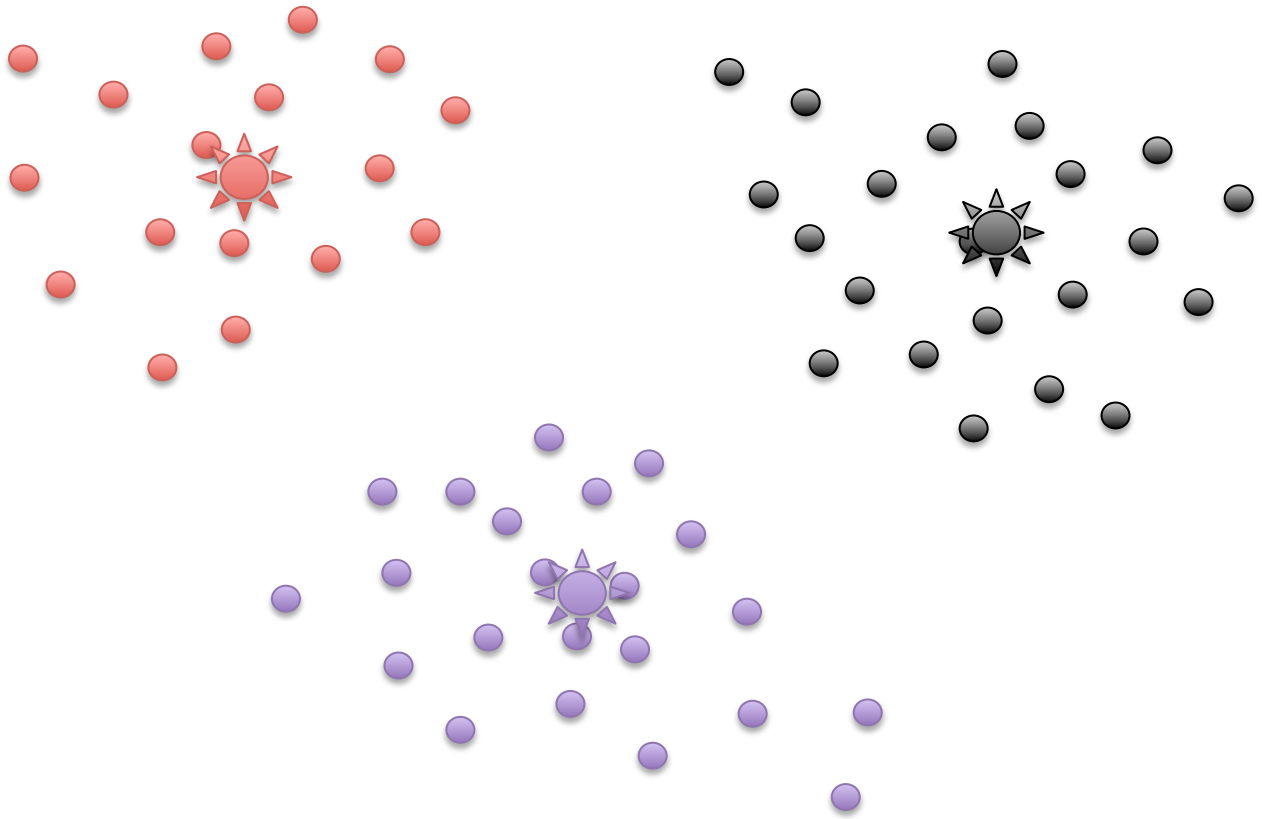
Centroid Based Clustering
(K-Means)



Centroid Based Clustering
(K-Means)



Centroid Based Clustering
(K-Means)



K-Means Objective

$$S = \{x_i\}_{i=1}^{N}$$

$$\underset{S=C_1 \cup \ldots \cup C_K,\ \{c_1,\ldots,c_K\}}{\operatorname{argmin}} \sum_k \sum_{x \in C_k} \|x - c_k\|^2$$

Clustering Cluster Centers

Equivalent!

$$\underset{S=C_1 \cup \ldots \cup C_K}{\operatorname{argmin}} \sum_k |C_k| \operatorname{var}(C_k)$$

EM Algorithm for K-Means

$$S = \{x_i\}_{i=1}^{N}$$

$$\underset{S=C_1 \cup \ldots \cup C_K,\ \{c_1,\ldots,c_K\}}{\operatorname{argmin}} \sum_k \sum_{x \in C_k} \|x - c_k\|^2$$

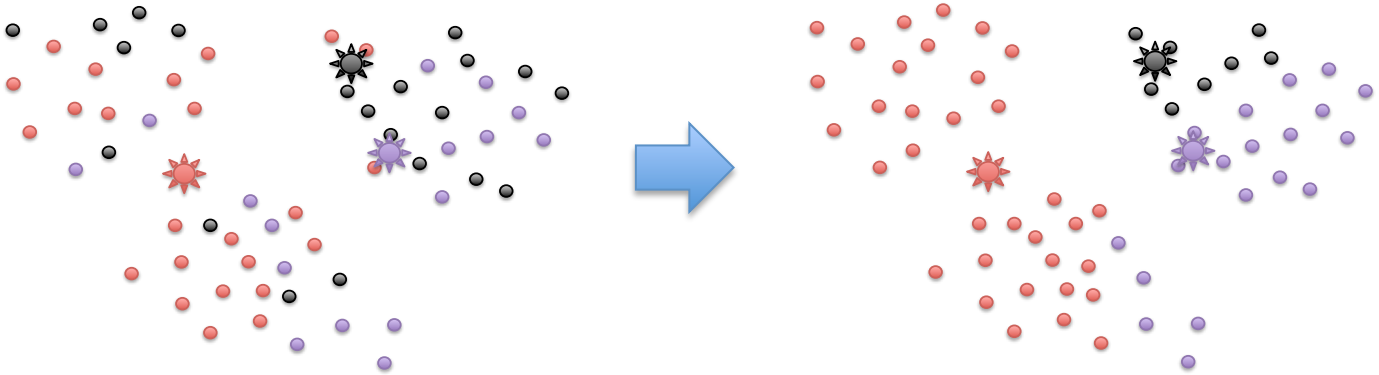
Clustering Cluster Centers

- E-Step
  - Estimate C_k
  - Estimate cluster membership
- M-Step
  - Estimate c_k
  - Estimate model parameters

E-Step

$$\underset{S=C_1\cup\ldots\cup C_K,\ \{c_1,\ldots,c_K\}}{\operatorname{argmin}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2 \qquad S = \{x_i\}_{i=1}^N$$

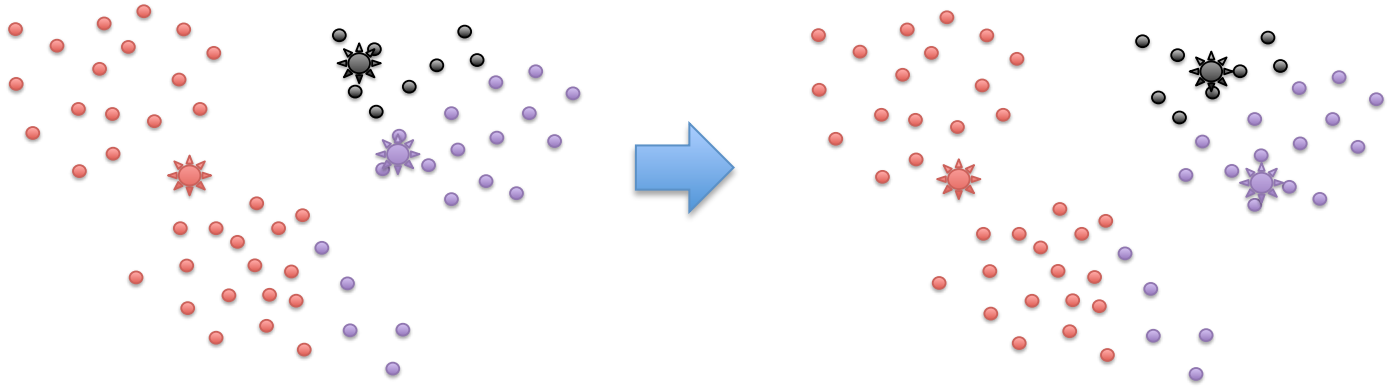
- For each x:
  - Assign to cluster C_k with smallest distance to c_k



M-Step

$$\underset{S=C_1\cup\ldots\cup C_K,\ \{c_1,\ldots,c_K\}}{\operatorname{argmin}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2 \qquad S = \{x_i\}_{i=1}^N$$

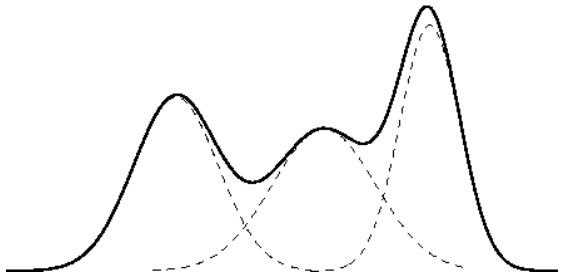
- For each c_k:
  - Compute c_k = mean(C_k)



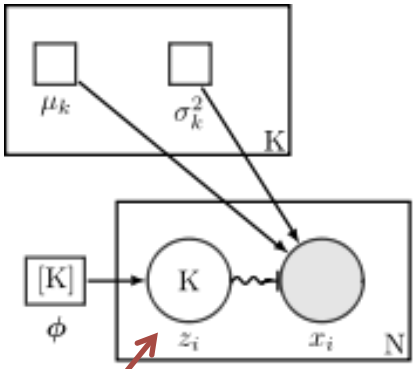
Aside: Gaussian Mixture Models

- Each data point is associated with a membership to a Gaussian distribution
  - Denoted by z variable

- 1D Example with 3 Gaussians



K Gaussian Distributions



N Data Points

Membership variable per data point

"Nonbayesian-gaussian-mixture" by Benwing – Created using LaTeX, TikZ. Licensed under CC BY 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Nonbayesian-gaussian-mixture.svg#/media/File:Nonbayesian-gaussian-mixture.svg

Aside: Gaussian Mixture Models

$$P\left(x \in C_k \mid c_1,\ldots,c_K\right) = \frac{\exp\left\{-\left\|x - c_k\right\|^2 / 2\sigma^2\right\}}{\sum_{k'} \exp\left\{-\left\|x - c_{k'}\right\|^2 / 2\sigma^2\right\}}$$

$$\propto \exp\left\{-\left\|x - c_k\right\|^2 / 2\sigma^2\right\}$$

- Prob of cluster membership proportional to $\exp(-\text{dist}^2/2\sigma^2)$
- "Sharpness" of distribution increases as σ decreases
- Converges to K-Means as σ goes to 0

Aside: Gaussian Mixture Models

Gaussian likelihood

$$\underset{\{c_1,\ldots,c_K\}}{\operatorname{argmax}} \prod_{x \in S} P(x) = \prod_{x \in S} \sum_{k} P\left(x \in C_k\right) P(k) \qquad S = \left\{x_i\right\}_{i=1}^{N}$$

Prior on each Gaussian mixture
(can assume = 1/K for simplicity)

- E-Step: Estimate probabilities
- M-Step: Maximize model parameters $c_1,\ldots,c_k$

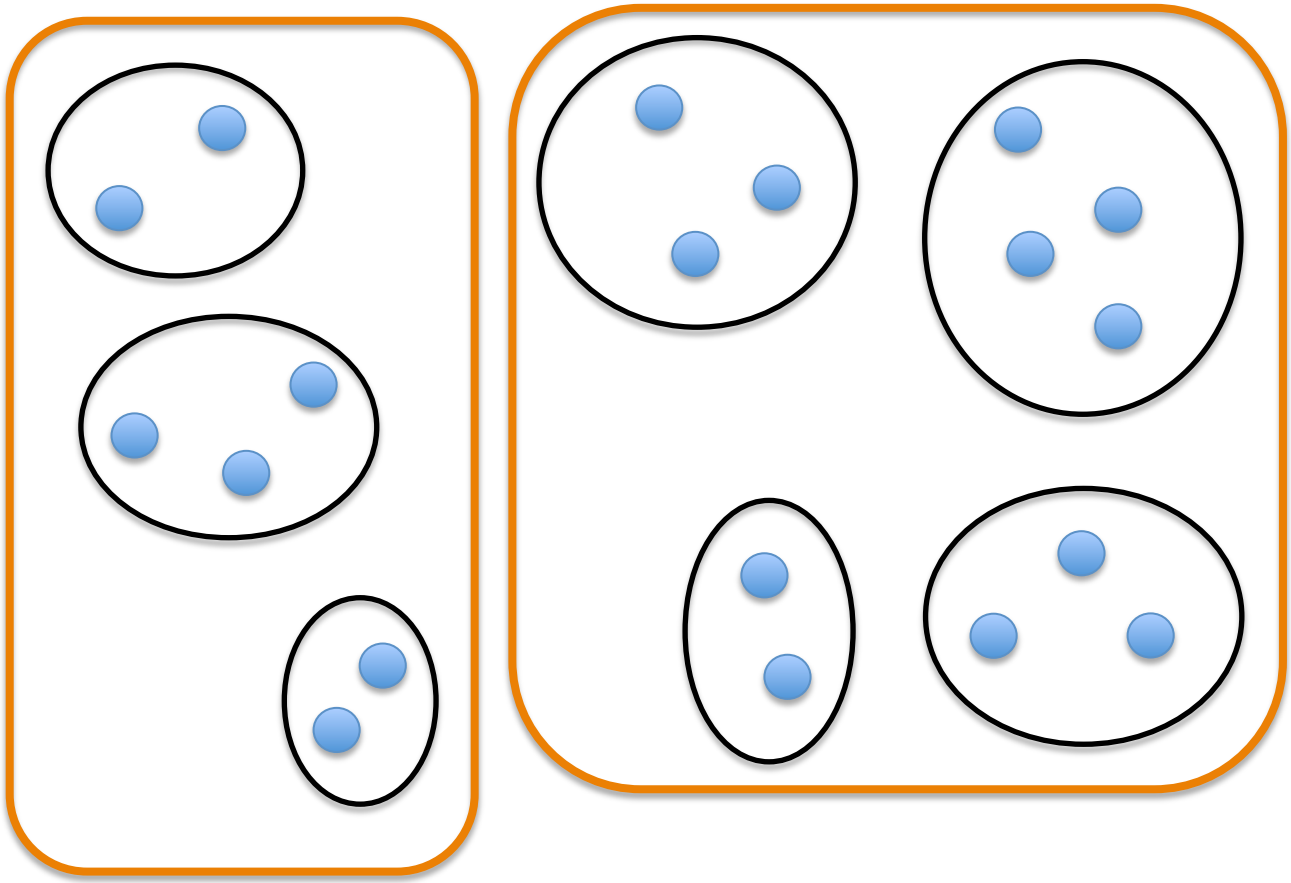
$$P\left(x \in C_k \mid c_1,\ldots,c_K\right) = \frac{\exp\left\{-\left\|x - c_k\right\|^2 / 2\sigma^2\right\}}{\sum_{k'} \exp\left\{-\left\|x - c_{k'}\right\|^2 / 2\sigma^2\right\}}$$

Recap: K-Means

- Centroid-based Clustering
  - Defines clusters using a notional of centrality
  - E.g., all items in the cluster must be close to each other
- Solve using EM algorithm
  - Also probabilistic variant (Gaussian Mixture Models)
- Useful when centrality assumption is good
  - But bad when centrality assumption is bad…

Thought Experiment

What is good clustering?

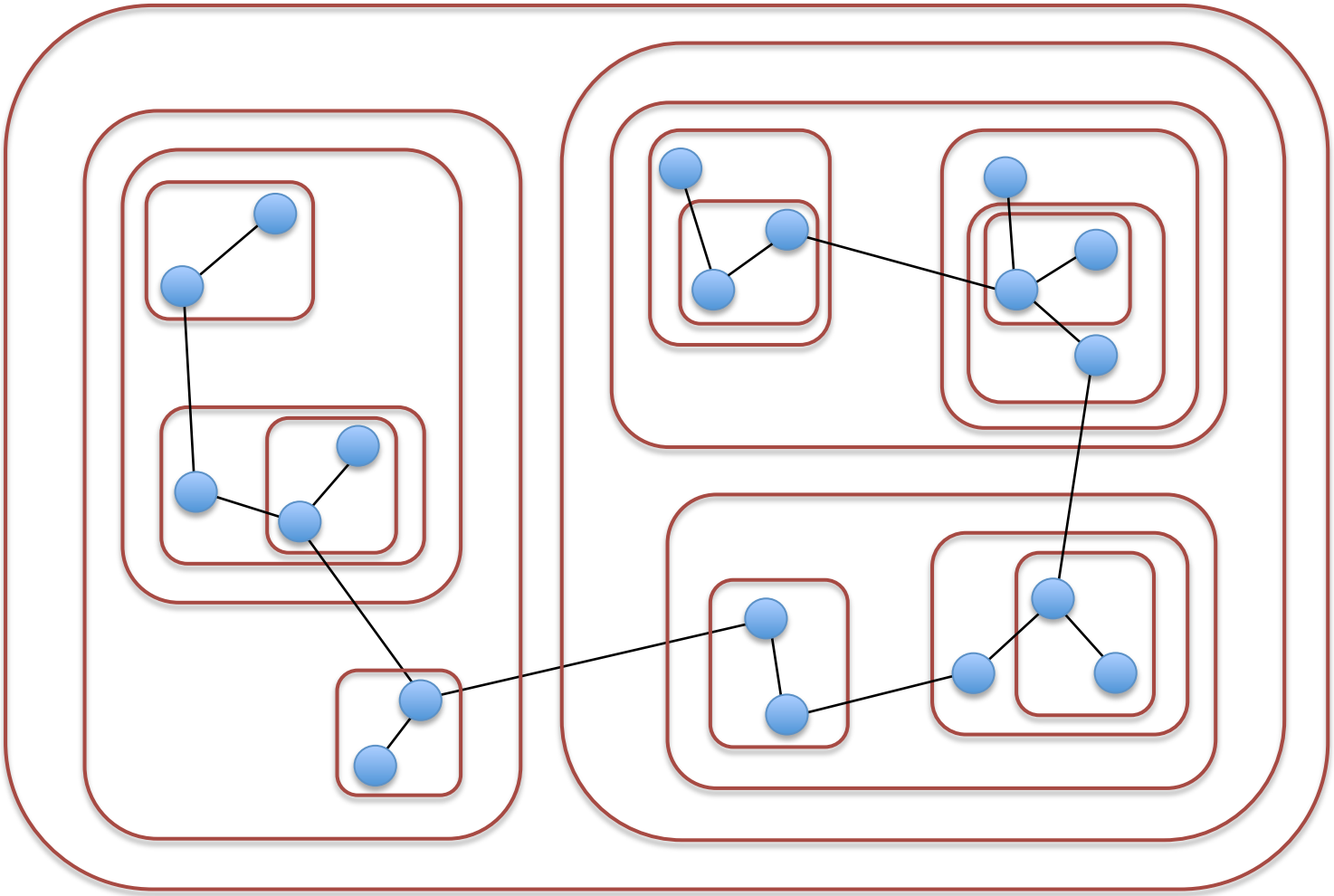


Linkage Based Clustering
(Hierarchical Clustering)

- K-Means used centroid clustering structure
  - Clustered data points are “close” to cluster center

- Sometimes a linkage structure is better…
  - Employ hierarchical clustering
  - E.g., agglomerative clustering

Agglomerative Clustering



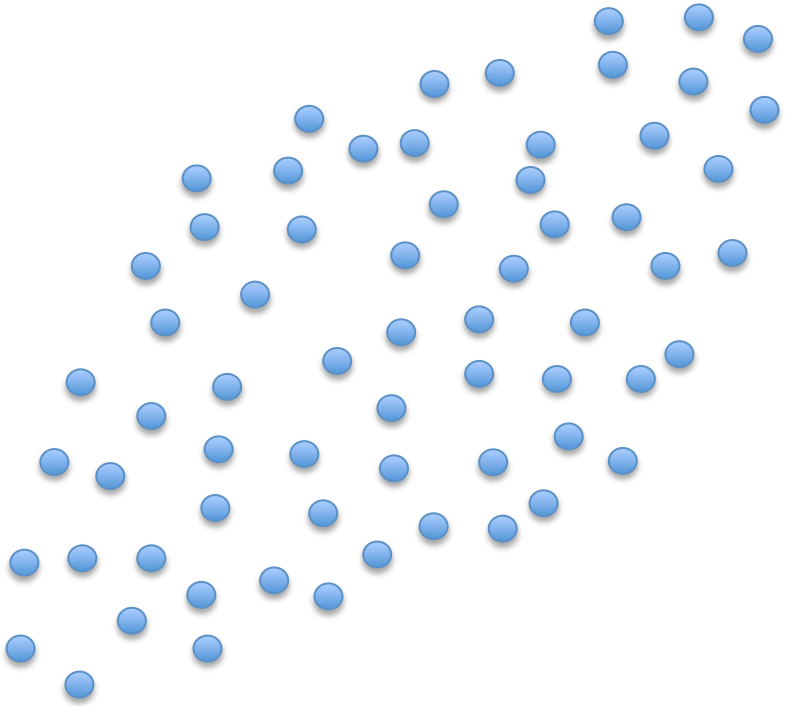
Agglomerative Clustering

- Equivalent to finding minimum spanning tree
  - Kruskal's Algorithm
  - http://en.wikipedia.org/wiki/Kruskal%27s_algorithm
- Order that edges are added defines the cluster hierarchy
- Equivalent to finding a binary tree partitioning with progressively smaller partition distances

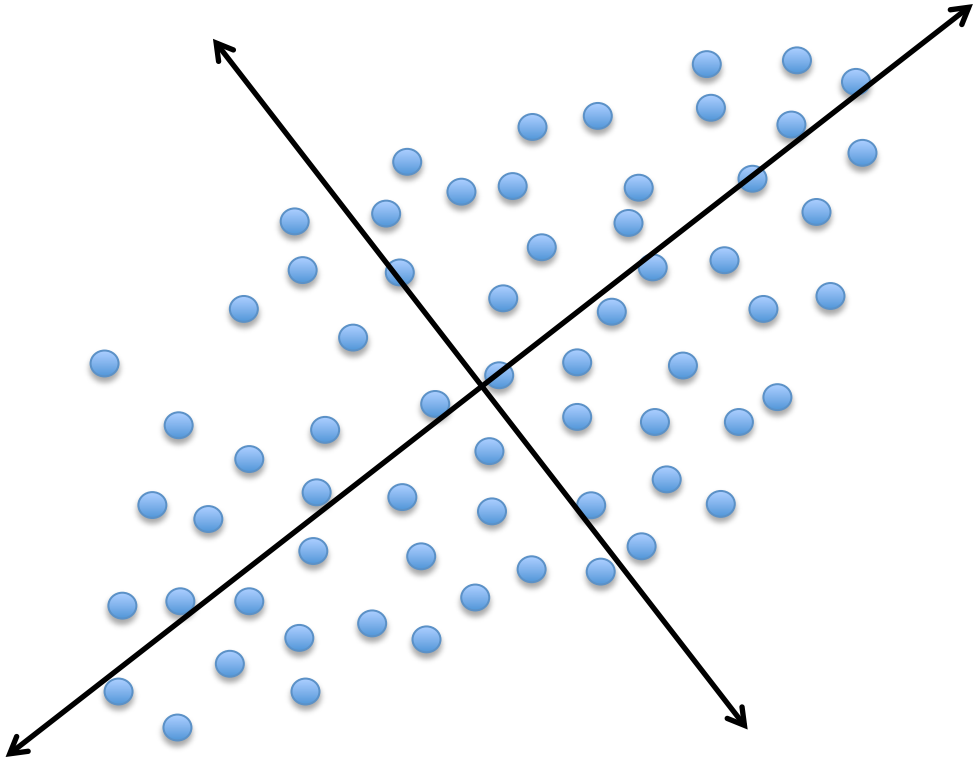
Recap: Clustering

- Unsupervised learning
  - Finds the clustering structure of input features
- Centroid based
  - Clusters should be clumped together
  - K-Means
- Linkage Based
  - Clusters can be organized hierarchically
  - Agglomerative Clustering
- Works great when clustering assumption is good!

Limitations of Clustering



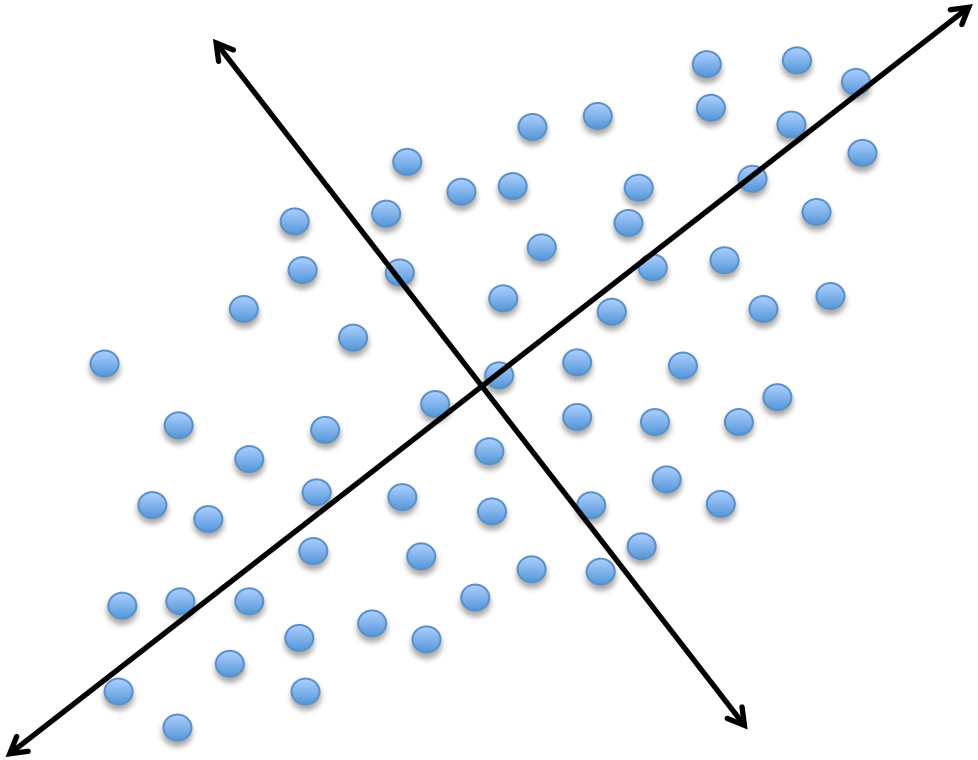
Principal Component Analysis



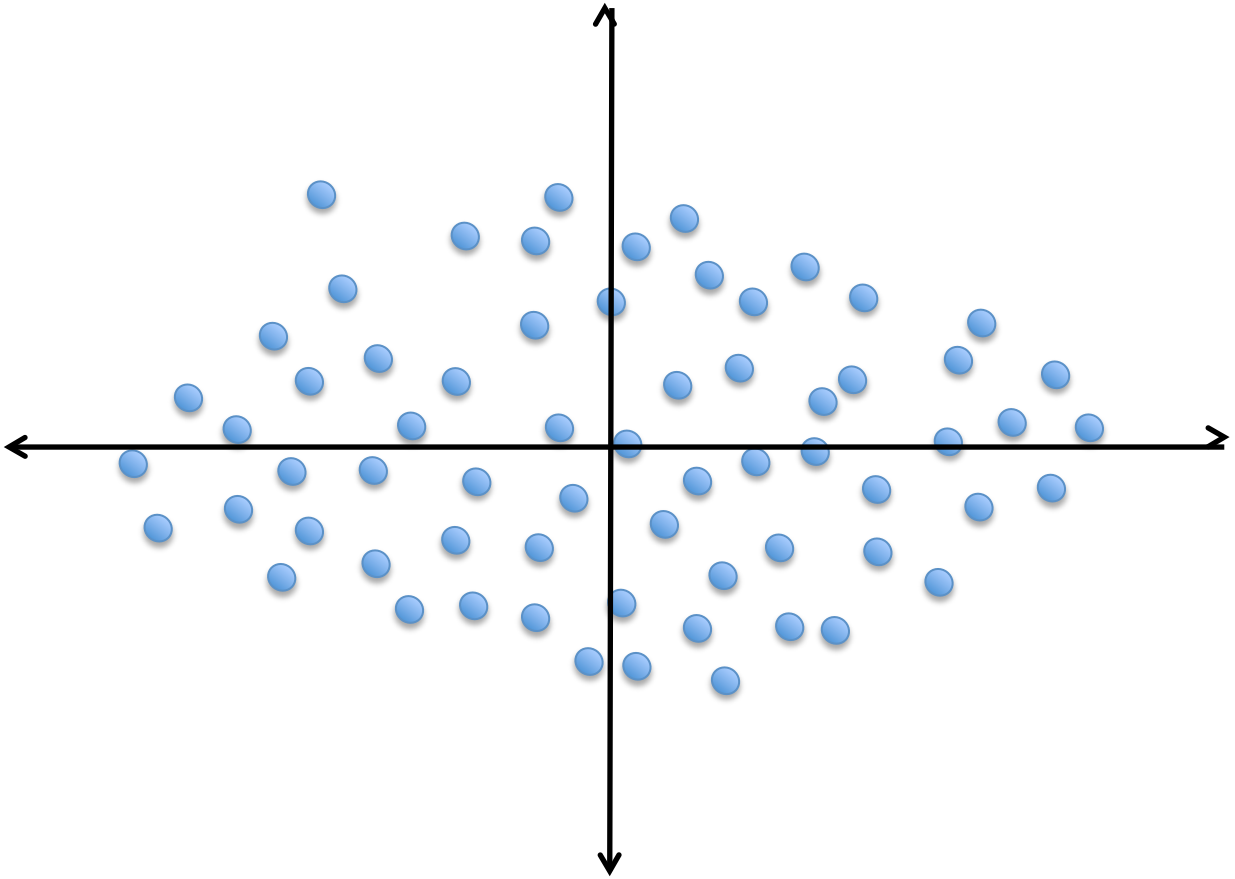
Summarizing Data

- Summarize data using smaller #attributes $S = \{x_i\}_{i=1}^{N}$
- Clustering: summarize data via clusters
  - K-Means: summarize via cluster membership
  - Gaussian Mixture Model: Summarize via distribution over K clusters
- PCA: summarize via orthogonal projections
  - Define new feature representation
  - Rotation + Projection

Principal Component Analysis



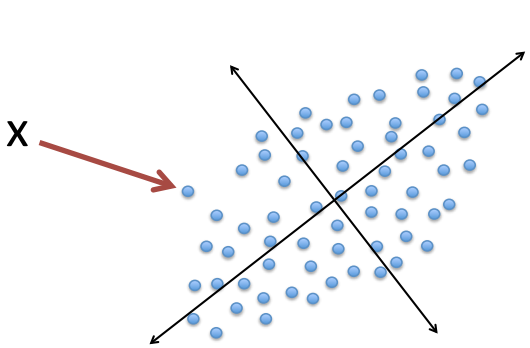
Principal Component Analysis

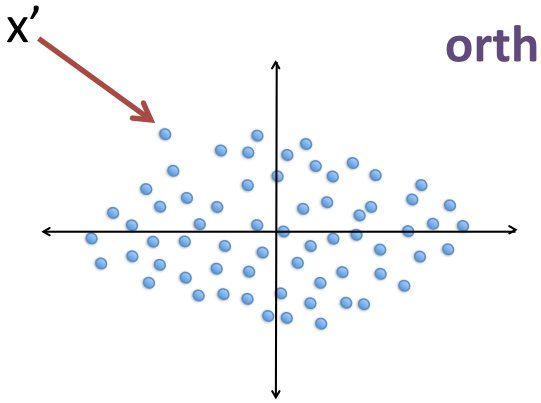


New Feature Representation!

Orthogonal Matrix

- A matrix U is orthogonal if $UU^T = U^TU = I$
  - For any column u: $u^Tu = 1$
  - For any two columns u, u': $u^Tu' = 0$
  - U is a rotation matrix, and U^T is the inverse rotation
  - If $x' = U^Tx$, then $x = Ux'$





PCA finds a specific orthogonal U

Properties of Orthogonal Matrices

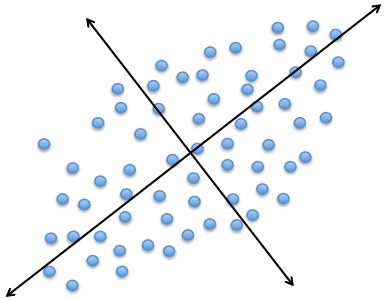
- $x' = U^T x, \; x = Ux'$
- Norm preserving:

$$x'^T x' = \left(U^T x\right)^T \left(U^T x\right) = x^T U U^T x = x^T x$$

- Preserves Total Variance:

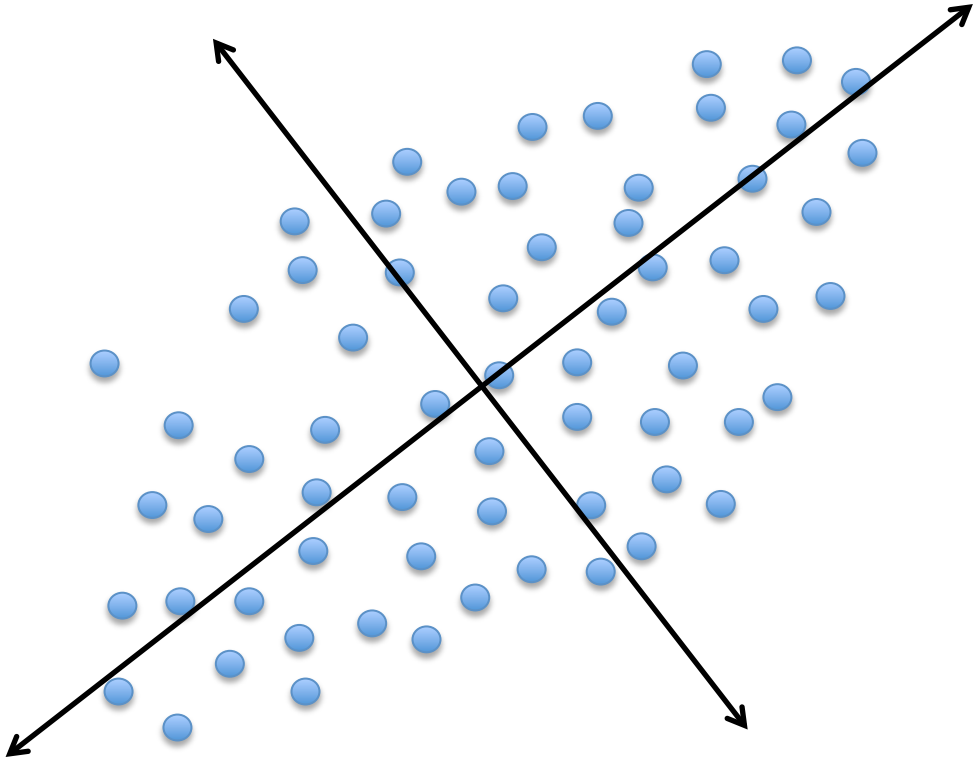
$$\sum_{d=1}^{D}\sum_{i=1}^{N}\left(x_i^{(d)}\right)^2 = \sum_{d=1}^{D}\sum_{i=1}^{N}\left(x'^{(d)}_i\right)^2$$

Assuming zero mean



Principal Component Analysis

Summarize Using 1 Feature?

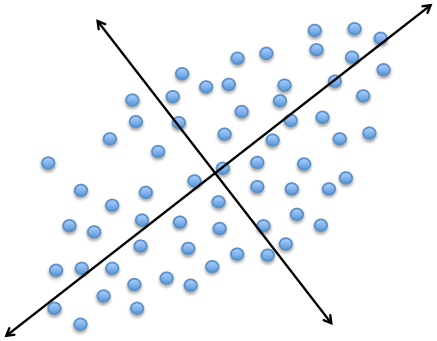


Principal Component Analysis

Summarize Using 1 Feature?

Principal Component Analysis

Summarize Using 1 Feature?



$u_1^T x$

$u_1 u_1^T x$

Works with arbitrary subsets of columns of orthogonal U

E.g., U' = [u_1,u_5,u_{20}]

PCA Formal Definition

- Define M=matrix of all data:

$$X = \left[x_1, \ldots, x_N \right] \in \mathrm{Re}^{D \times N}$$

- Mean center:

$$\bar{X} = X - \left[\bar{x}, .., \bar{x} \right]$$

- PCA:

$$\bar{X}\bar{X}^T = U \Lambda U^T$$

Symmetric

Orthogonal

Diagonal

Properties of PCA

$$XX^T = U\Lambda U^T$$

Assuming zero mean

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix}$$

- Each column of U is an Eigenvector
- Each λ is an Eigenvalue
  - $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$

$$\left(XX^T\right)u_d = \lambda_d u_d$$

Interpretation

Feature Covariance Matrix:

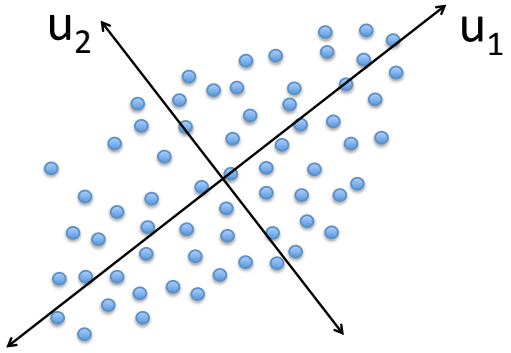
$$\Sigma = XX^T = U\Lambda U^T$$

Assuming zero mean

PCA Solution

- $\Sigma_{dd'}$ is the covariance of features d & d' in training data.
- The first column u_1 is the single direction of greatest variation
  - λ_1 is the total variation along u_1:

$$\lambda_1 = \sum_{i=1}^{N} \left(u_1^T x_i\right)^2 = \sum_{i=1}^{N} \left(x'^{(1)}_i\right)^2$$



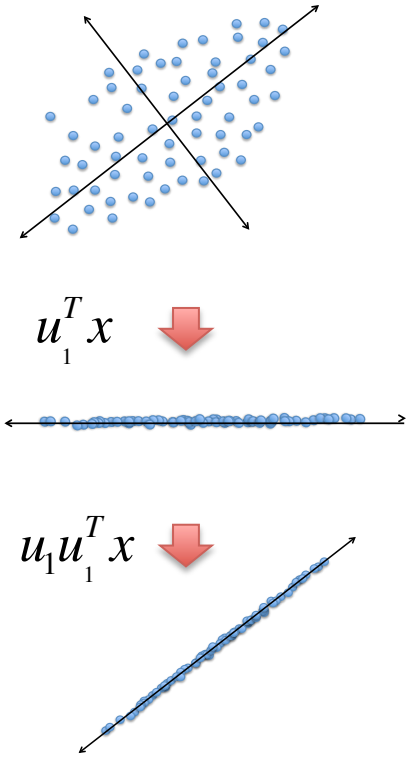
Interpretation Continued

- The first column u_1 is the single direction that minimizes the squared loss of reconstructing the original x's
  - I.e., minimizes the amount of residual variation
- **One can prove that:**

$$u_1 = \underset{u:\, u^T u=1}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| x_i - u u^T x_i \right\|^2$$

"Residual"

- (From definition in previous slide)



Definition: u_1 is the direction that captures the most variation

$$u_1 = \arg\max_{u:\, u^T u=1} \sum_{i=1}^{N} \left\| u^T x_i \right\|^2$$

Step 1: for any x, its residual direction is orthogonal to u_1

Residual: $x - u_1 u_1^T x$

$$\left(x - u_1 u_1^T x\right)^T u_1 = x^T u_1 - x^T u_1 u_1^T u_1 = x^T u_1 - x^T u_1 = 0$$

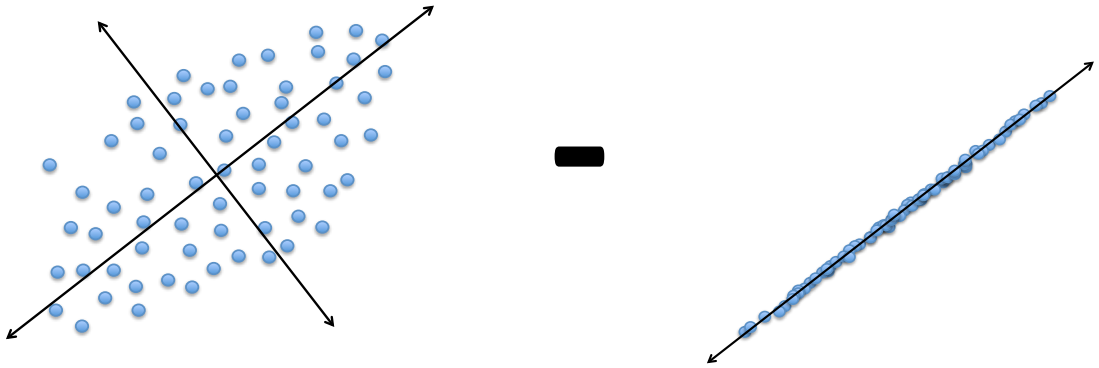
Step 2: establish relationship and complete proof

$$\sum_{i=1}^{N} \left\| x_i - uu^T x_i \right\|^2 = \sum_{i=1}^{N} \left(x_i - uu^T x_i\right)^T \left(x_i - uu^T x_i\right) = \sum_{i=1}^{N} \left(x_i^T x_i - 2 x_i^T uu^T x_i + x_i^T uu^T uu^T x_i\right)$$

$$= \sum_{i=1}^{N} \left(x_i^T x_i - x_i^T uu^T x_i\right) \qquad = \sum_{i=1}^{N} \left(x_i^T x_i\right) - \sum_{i=1}^{N} \left(x_i^T uu^T x_i\right)$$

Interpretation Continued

Find the u_1 that minimizes the residual squared norm:



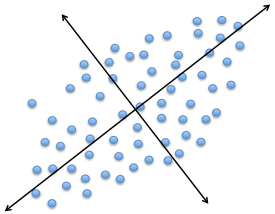
$$u_1 u_1^T x$$

Solving PCA
(Iterative Algorithm)

- Given: $X = [x_1, \ldots, x_N] \in \mathrm{Re}^{D \times N}$ Assuming zero mean
- Init: $X_1 = X$
- For d=1,…,D
  - Solve: $u_d = \arg\min_{u:\ u^T u = 1} \left\| X_d - u u^T X_d \right\|_{Fro}^2$
  - Update: $X_{d+1} = X_d - u_d u_d^T X_d$

Property of PCA

$$XX^T = U\Lambda U^T$$



- The first K columns of U are guaranteed to be the K-dimensional subspace that captures the most variability of X

- We just proved K=1 a few slides ago

Dimensionality Reduction

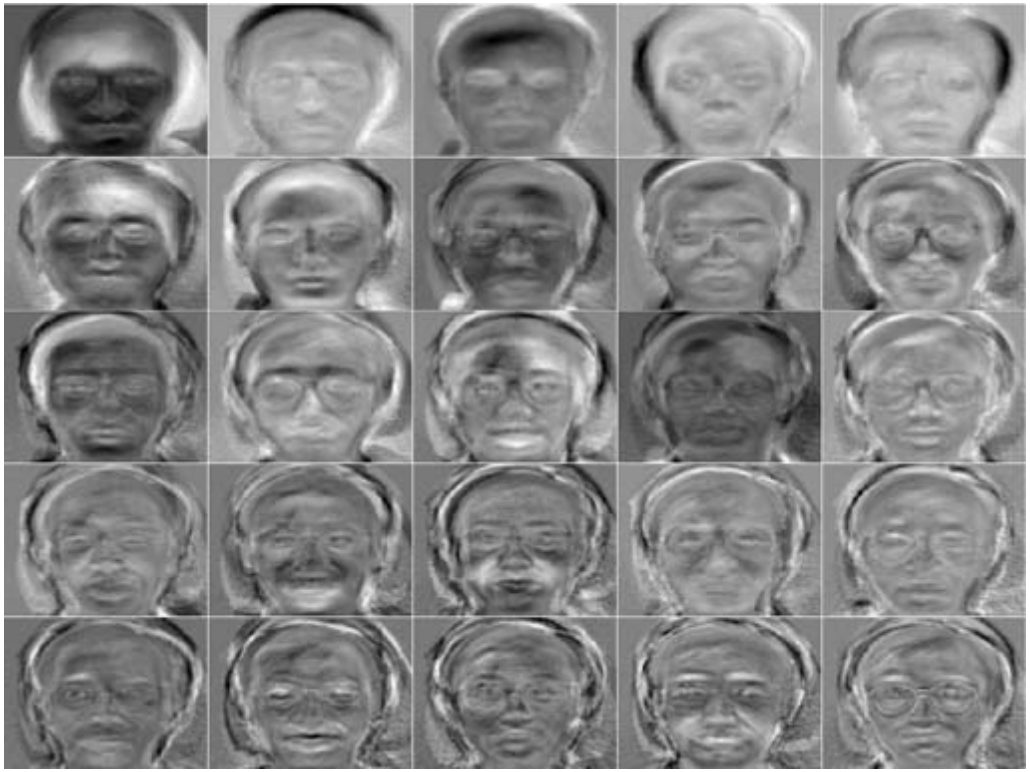
- Solve PCA: $XX^T = U\Lambda U^T$

- Use first K columns of U to create K-dim representation:

$$x' = U_{1:K}^T x$$

- This creates a compact summary of original dataset
  - E.g., K = 50, D = 1,000,000

Example: Eigenfaces



PCA on a corpus of faces.
Every pixel is a “feature”
Visualizing the top Eigenvectors of U

http://www.cs.princeton.edu/~cdecoro/eigenfaces/

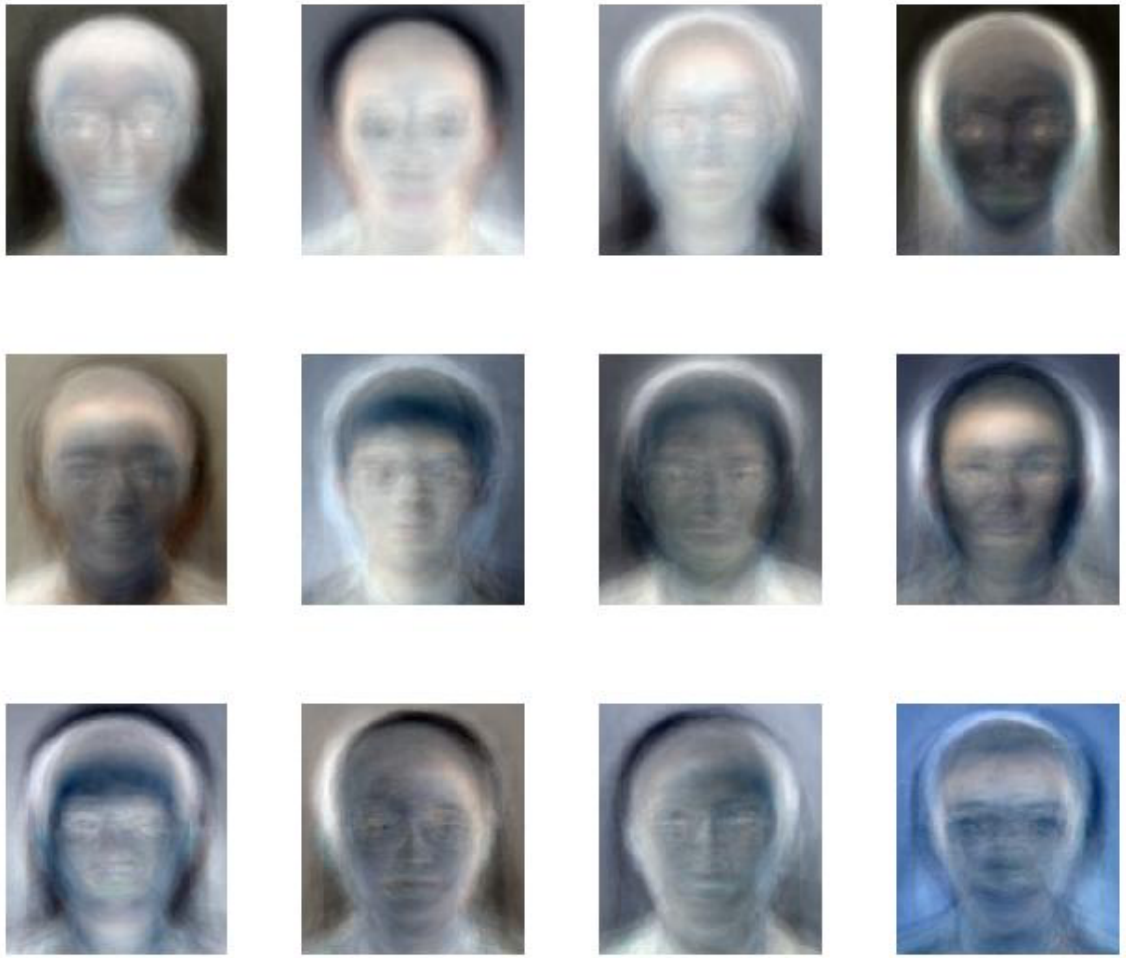
Example: Eigenfaces

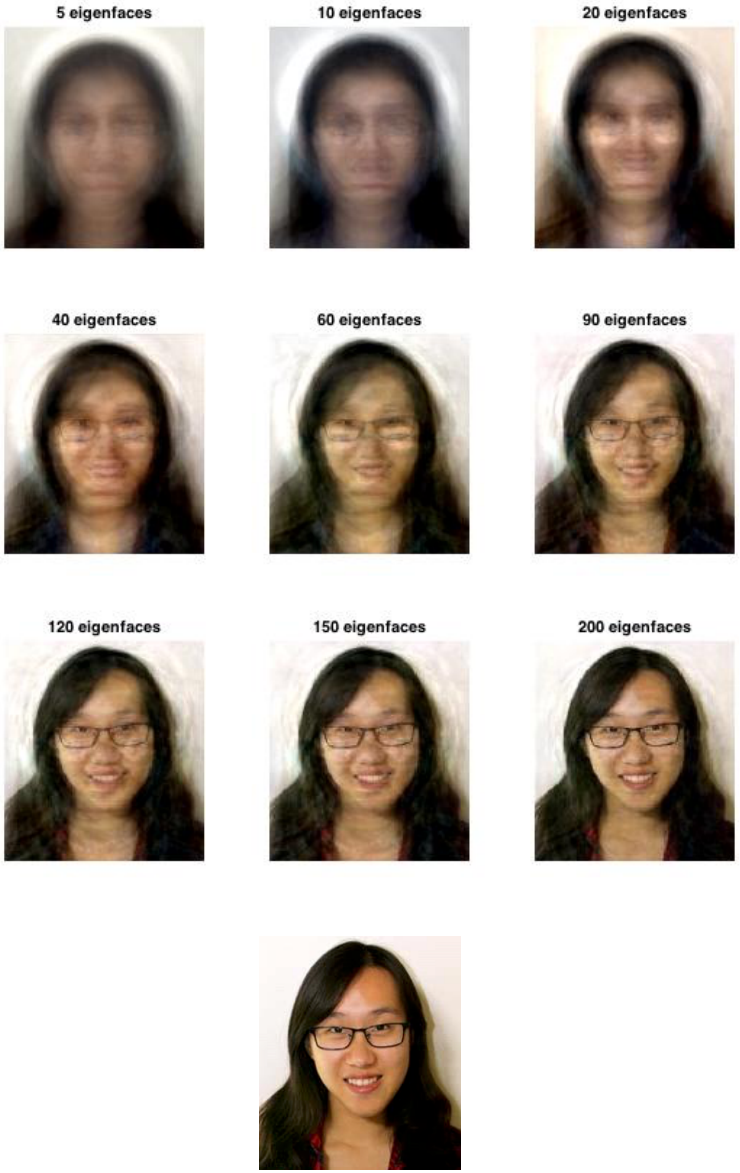
Visualizing Projection using top K Eigenvectors: $U_{1:K} U_{1:K}^T x$

http://www.cs.princeton.edu/~cdecoro/eigenfaces/

CS 155 Eigenfaces

Avg Face




5 eigenfaces
10 eigenfaces
20 eigenfaces
40 eigenfaces
60 eigenfaces
90 eigenfaces
120 eigenfaces
150 eigenfaces
200 eigenfaces
5 eigenfaces
10 eigenfaces
20 eigenfaces
40 eigenfaces
60 eigenfaces
90 eigenfaces
120 eigenfaces
150 eigenfaces
200 eigenfaces

Singular Value Decomposition

$$X = U\Sigma V^T$$

Orthogonal

Orthogonal Diagonal

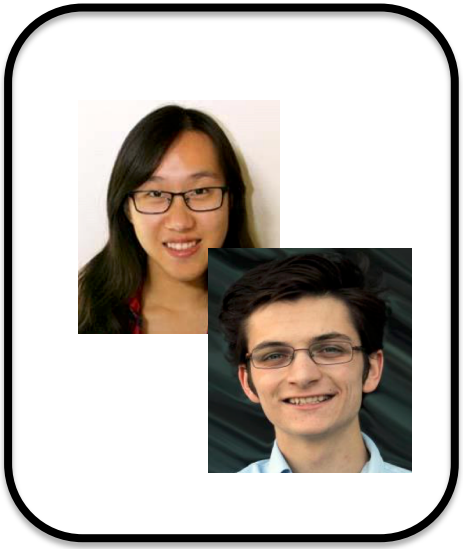
- SVD operates on X, as opposed to XX^T
- Equivalence between SVD & PCA

$$XX^T = \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T = U\Sigma V^T V\Sigma U^T = U\Sigma^2 U^T$$

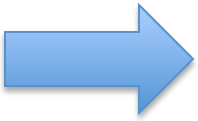
- V corresponds to new representation x'

Eigenfaces Step 1

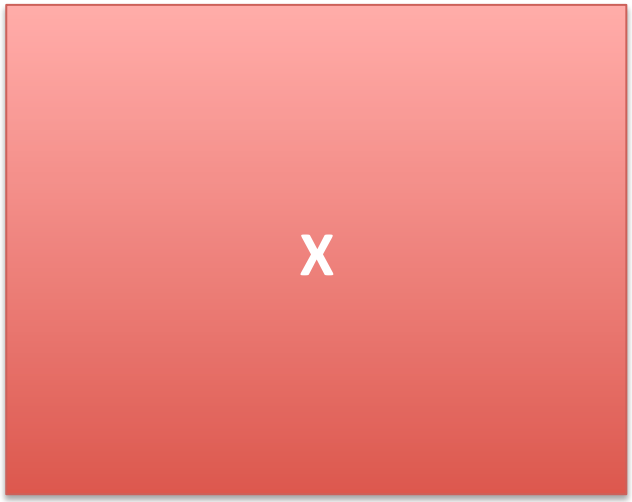
- Flatten each image into vector



HxWx3



225000-dimensional!



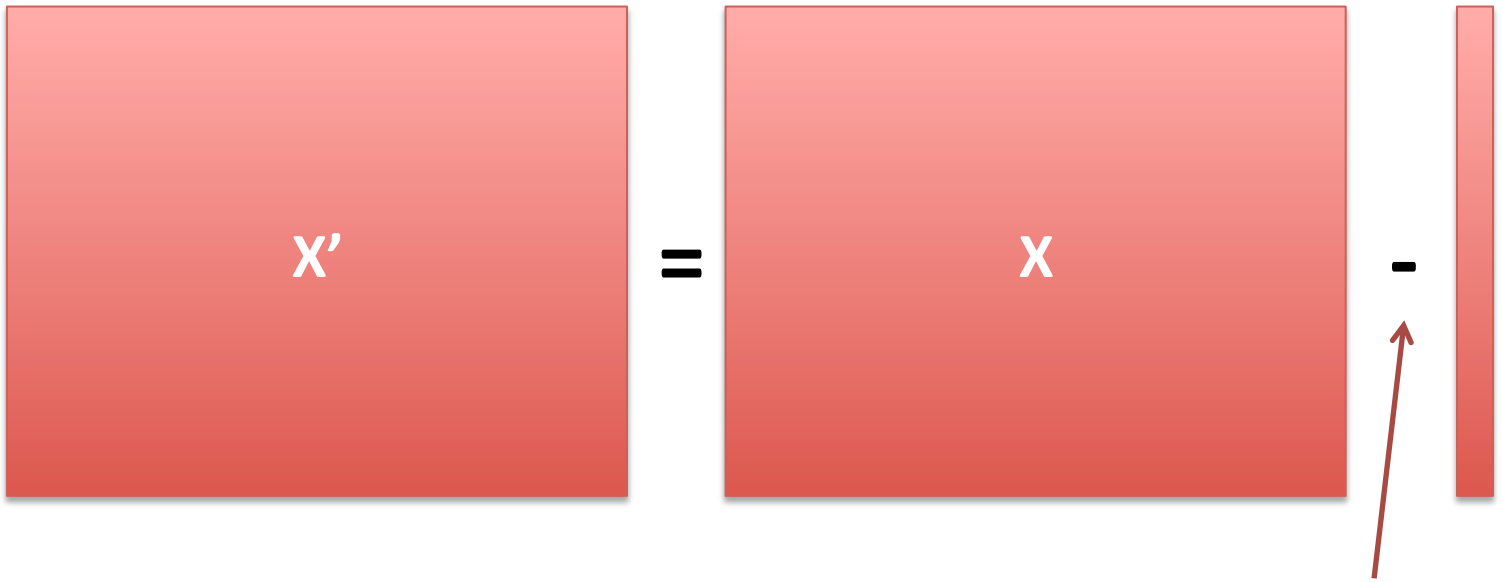
(3*H*W)xN

Each Column is Image

Eigenfaces Step 2

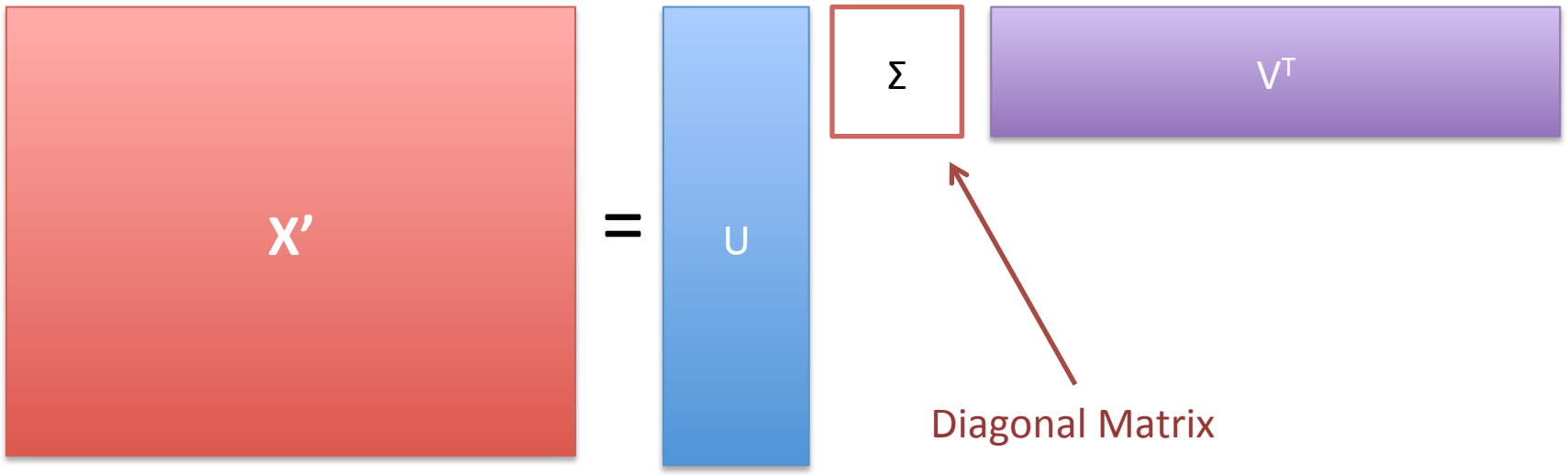
- Mean center

Mean



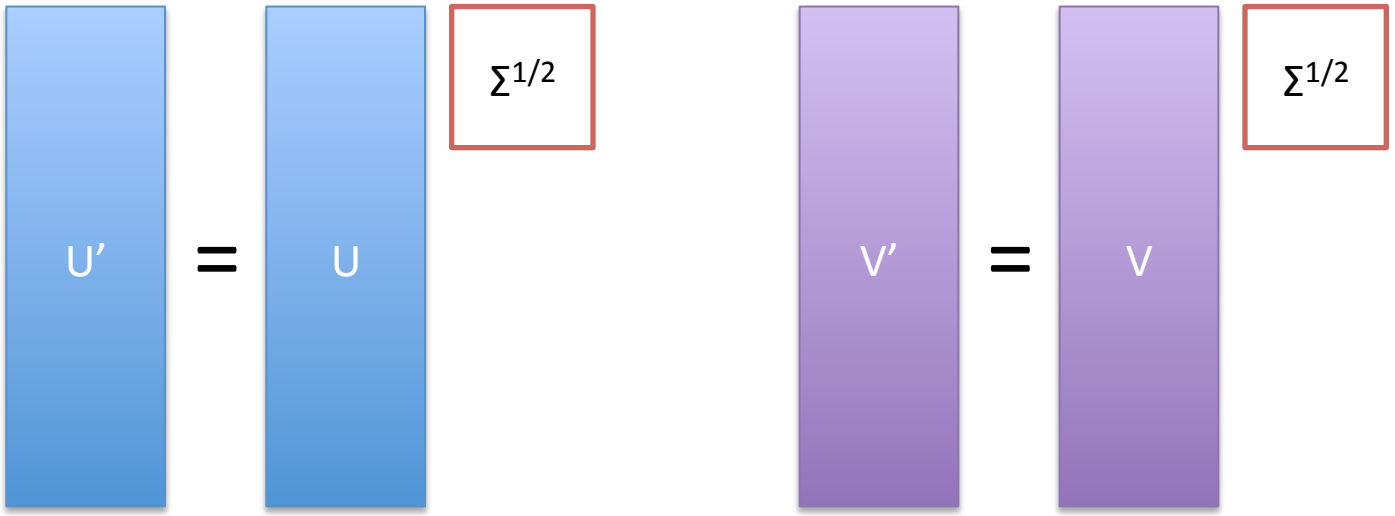
Eigenfaces Step 3

- Singular Value Decomposition: $X' = U\Sigma V^T$

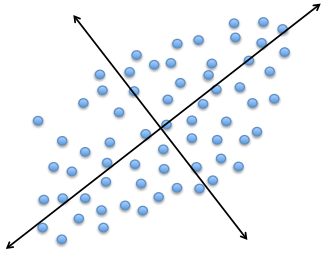


Eigenfaces Step 4

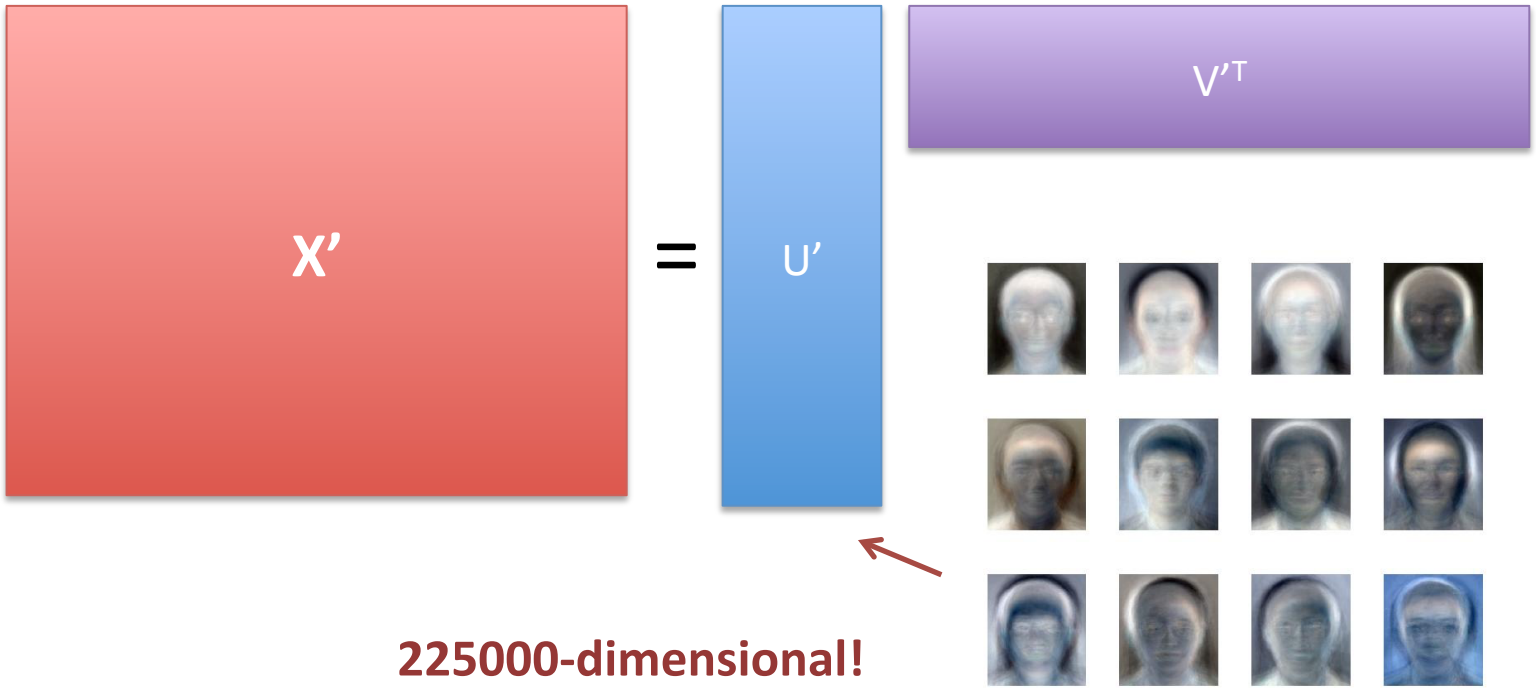
- Merging Σ into U and V: $X' = U\Sigma V^T = U'V'^T$

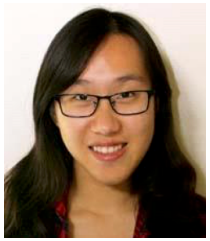


Interpreting U & V



- Each col of U' is an "Eigenface"
- Each col of V'^T = coefficients of a student






24.4685 , 67.098


-53.1528 , 44.1135


34.1016 , 8.8478


-22.2237 , 11.9005


4.4777 , -10.7395


9.2998 , 14.604


-1.8439 , 14.0771


27.2681 , 22.2581


15.5636 , -14.8784

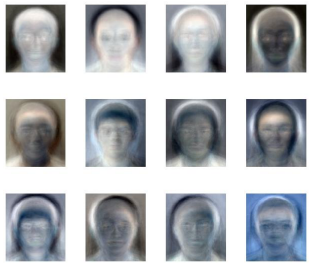

28.1657 , 0.35045


-0.09123 , 6.3281


-26.2752 , -3.1067

Limitations of Eigenfaces

- Each dimension is a pixel (& color channel)
  - Not semantically meaningful
  - Squared reconstruction error in pixel space



- Suppose each dimension had more meaning
  - E.g., dim 1 = location of left eye
  - Then U components would have cleaner visualization

Summary

- Clustering & PCA (and SVD) reduce the dimensionality of data representation.
- For each data point
  - Store K numbers
  - Cluster membership probabilities
  - Coefficients in K-dimensional projection
- Nice visualization & interpretation?
  - Depends on semantics of raw dimensions…

Next Lecture

- Latent Factor Models
- Matrix Factorization with Missing Values
  - E.g., the "Netflix Problem"
- No Recitation Today