CS 760: Machine Learning
Dimensionality Reduction

University of Wisconsin-Madison

Outline

- **Clustering Review**
  - k-means, hierarchical, spectral clustering
- **GMMs Review**
  - Gaussian Mixtures, EM algorithm
- **Principal Components Analysis**
  - Definition, Algorithm, Interpretations, Analysis, Applications

Outline

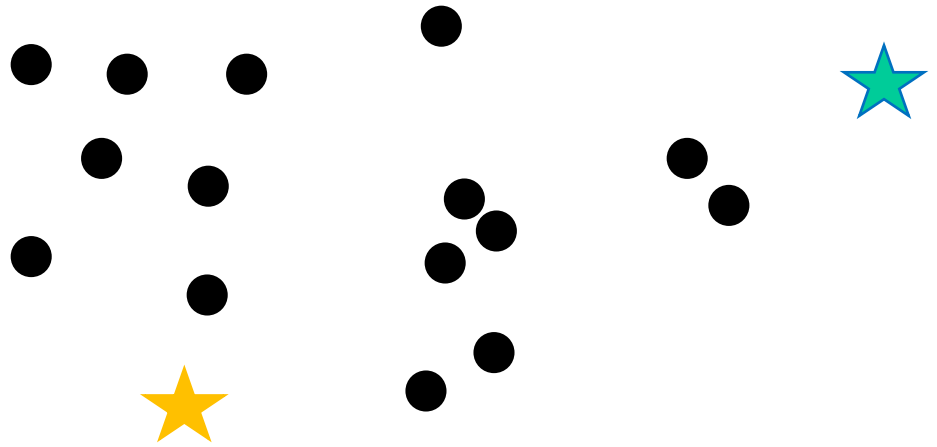
- **Clustering Review**
  - k-means, hierarchical, spectral clustering
- **GMMs Review**
  - Gaussian Mixtures, EM algorithm
- **Principal Components Analysis**
  - Definition, Algorithm, Interpretations, Analysis, Applications

K-Means Clustering

k-means is a type of partitional **centroid-based clustering**

Algorithm:

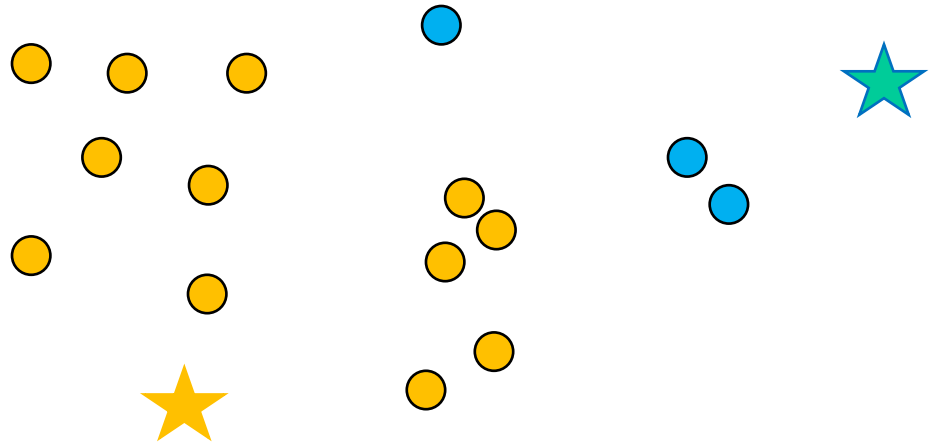
1. Randomly pick k cluster centers



K-Means Clustering: Algorithm

k-means clustering

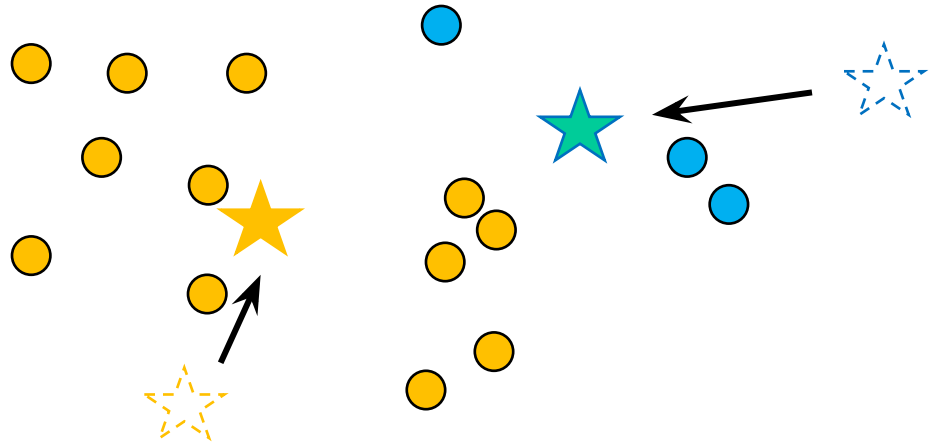
2. Find closest center for each point



K-Means Clustering: Algorithm

k-means clustering

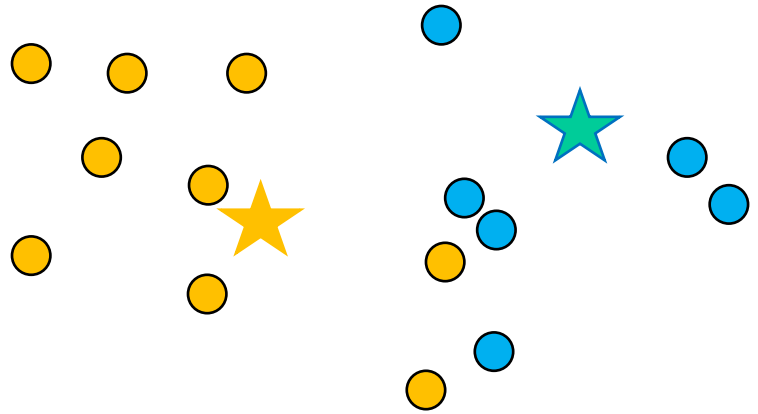
3. Update cluster centers by computing **centroids**



K-Means Clustering: Algorithm

k-means clustering

Repeat Steps 2 & 3 until convergence

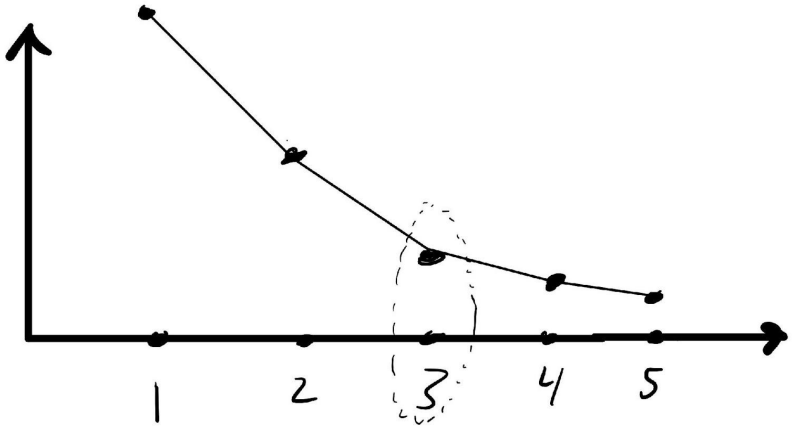


Questions on k-means

- What is k-means trying to optimize?

$$\sum_{x \in \{x_1, \ldots, x_n\}} \left\| x - c_{i_x} \right\|^2$$

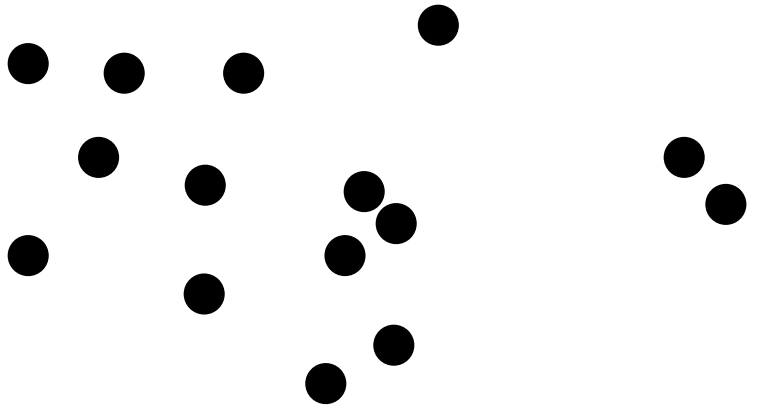
- Will k-means stop (converge)?
- Will it find a global or local optimum?
- How many clusters should we use?
- How to pick starting cluster centers?



Chris De Sa

HC: Agglomerative Clustering Example

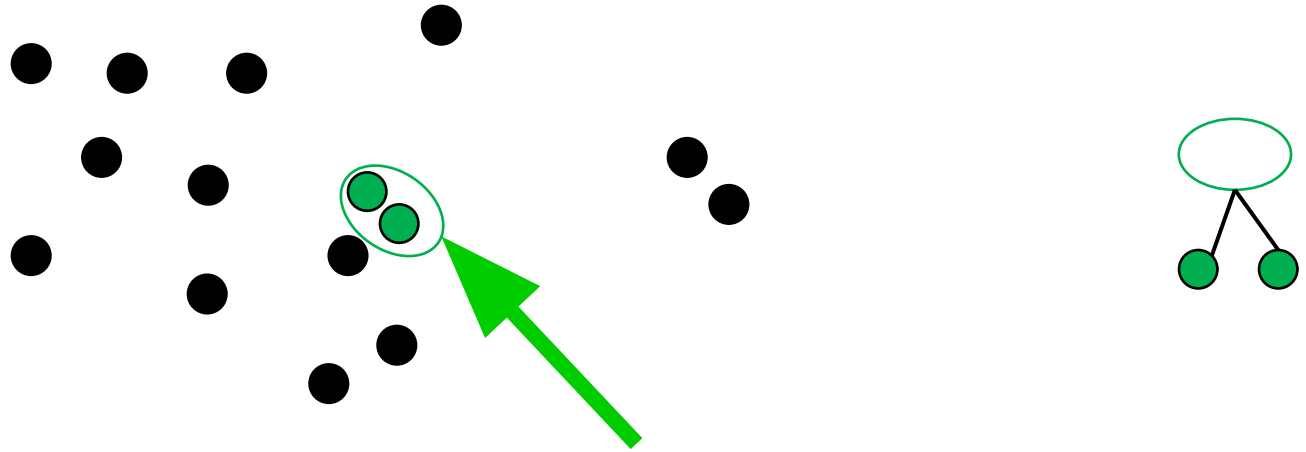
Agglomerative: Start: every point is its own cluster



HC: Agglomerative Clustering Example

Basic idea: build a “hierarchy”

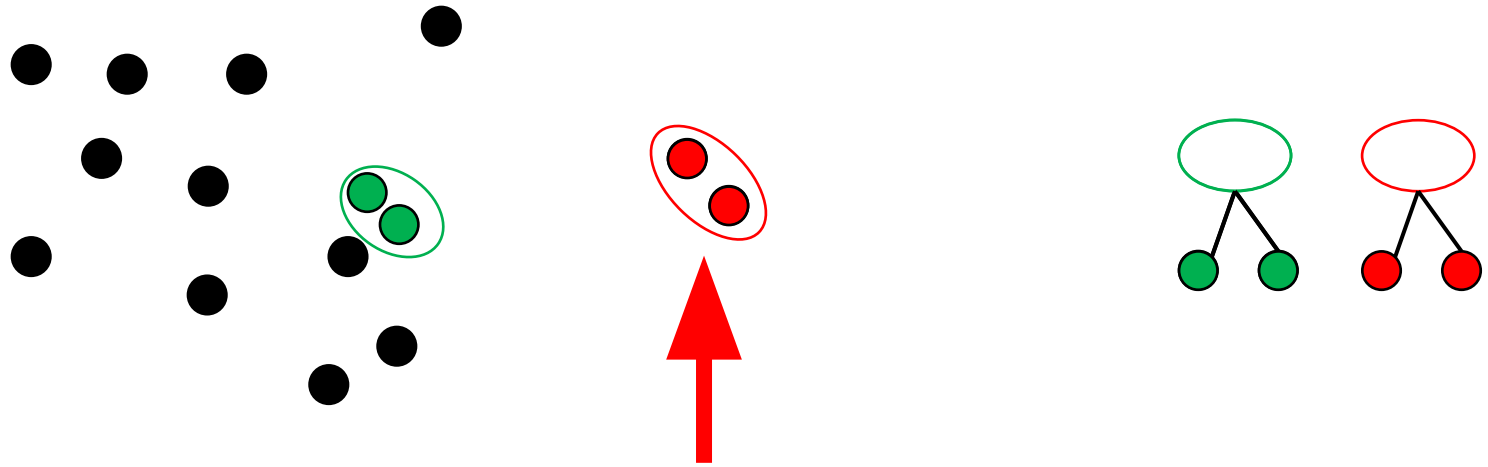
- Get pair of clusters that are closest and merge



HC: Agglomerative Clustering Example

Basic idea: build a “hierarchy”

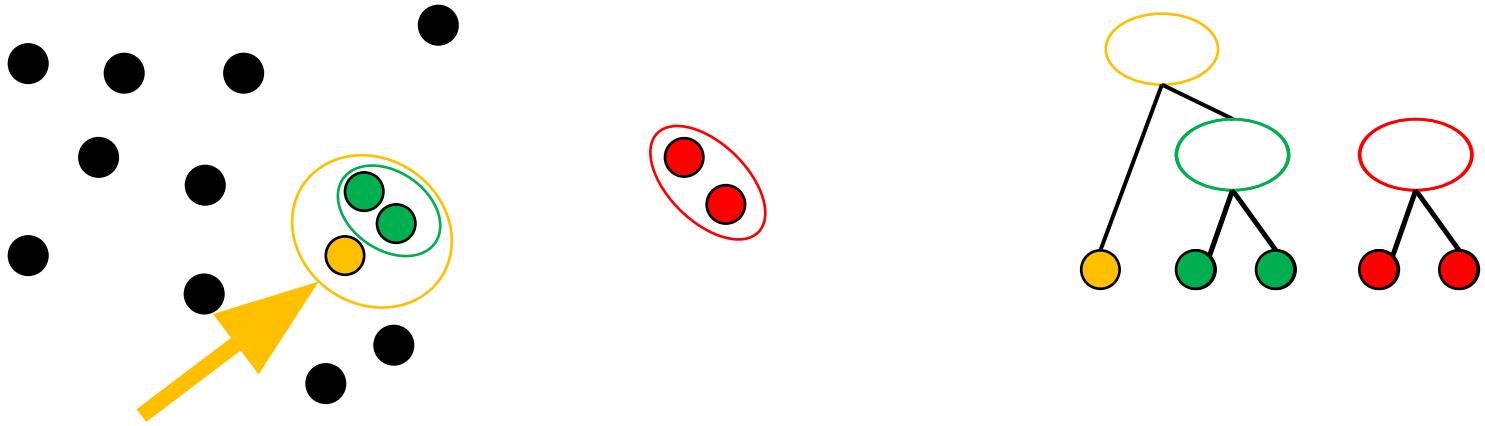
- **Repeat:** Get pair of clusters that are closest and merge



HC: Agglomerative Clustering Example

Basic idea: build a “hierarchy”

- **Repeat:** Get pair of clusters that are closest and merge



HC: Merging Criteria

Merge: use closest clusters. Define closest?

First define a distance between points $d(x_1, x_2)$. Then, define distance between clusters.

- Single-linkage $$d(A, B) = \min_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$
- Complete-linkage $$d(A, B) = \max_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$
- Average-linkage $$d(A, B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Break & Quiz

Break & Quiz

Q: You have seven 2-dimensional points. You run 3-means on it, with initial clusters

$$C_1 = \{(2,2),(4,4),(6,6)\}, C_2 = \{(0,4),(4,0)\}, C_3 = \{(5,5),(9,9)\}$$

Cluster centroids at the next iteration are?

- A. C_1: (4,4), C_2: (2,2), C_3: (7,7)
- B. C_1: (6,6), C_2: (4,4), C_3: (9,9)
- C. C_1: (2,2), C_2: (0,0), C_3: (5,5)
- D. C_1: (2,6), C_2: (0,4), C_3: (5,9)

Break & Quiz

Q 2.1: You have seven 2-dimensional points. You run 3-means on it, with initial clusters

$$C_1 = \{(2,2),(4,4),(6,6)\}, C_2 = \{(0,4),(4,0)\}, C_3 = \{(5,5),(9,9)\}$$

Cluster centroids at the next iteration are?

- **A. C_1: (4,4), C_2: (2,2), C_3: (7,7)**
- B. C_1: (6,6), C_2: (4,4), C_3: (9,9)
- C. C_1: (2,2), C_2: (0,0), C_3: (5,5)
- D. C_1: (2,6), C_2: (0,4), C_3: (5,9)

Outline

- **Clustering Review**
  - k-means, hierarchical, spectral clustering
- **GMMs Review**
  - Gaussian Mixtures, EM algorithm
- **Principal Components Analysis**
  - Definition, Algorithm, Interpretations, Analysis, Applications

Mixture Models

- Have dataset:

$$\{(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$$

- One type of model: **mixtures**
  - A function of a **latent variable** z
  - Model:

$$p(x^{(i)}|z^{(i)})p(z^{(i)})$$

Mixture Models: Gaussians

- Many different types of mixtures, but let us focus on Gaussians.
- What does this mean?
- Latent variable z has some multinomial distribution, $\sum_{i=1}^{k} \phi_i = 1$

$$z^{(i)} \sim \text{Multinomial}(\phi)$$

- Then, let us make x be Gaussian conditioned on z

$$x^{(i)}|(z^{(i)} = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$$

Mean Covariance Matrix

Gaussian Mixture Models: Likelihood

- How should we learn the parameters? ϕ, μ_j, Σ_j
- Could try our usual way: maximum likelihood
  - Log likelihood:

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{n} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi)$$

  - Turns out to be **hard** to solve… inner sum leads to problems!

GMMs: Supervised Setting

- What if we knew the z's?
  - "Supervised" setting…
- First, empirically estimate the multinomial parameters:

$$\phi_j = \frac{1}{n}\sum_{i=1}^{n} 1\{z^{(i)} = j\}$$

- Next the Gaussian components:

$$\mu_j = \frac{\sum_{i=1}^{n} 1\{z^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{z^{(i)} = j\}}$$

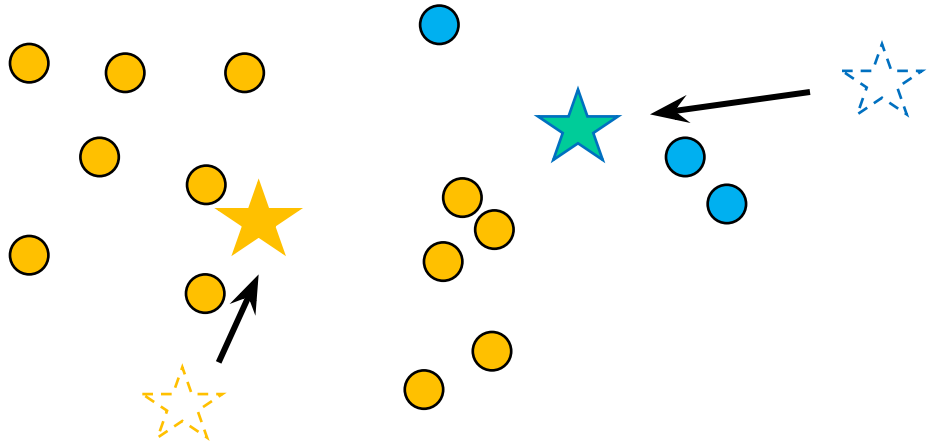
Average of x's where z = j

$$\Sigma_j = \frac{\sum_{i=1}^{n} 1\{z_j^{(i)} = j\}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} 1\{z_j^{(i)} = j\}}$$

GMMs: Back to Latent Setting

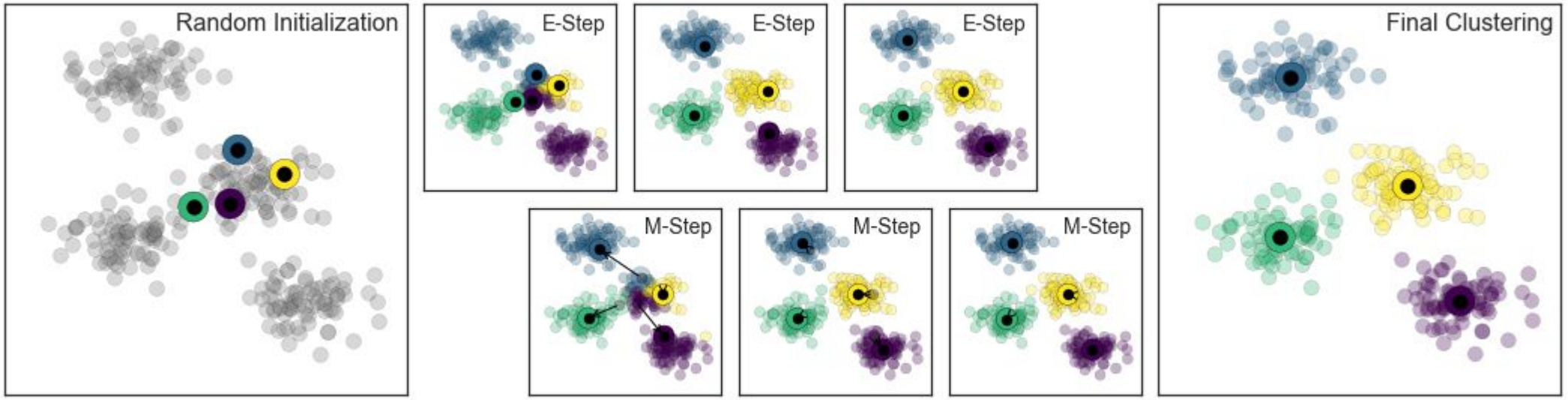
- But, we don't get to see the z's!

- What could we do instead?
- Recall our **k-means** approach: we don't know the centers, but we pretend we do, perform a clustering, re-center, iterate



GMMs: Expectation Maximization

- EM :an algorithm for dealing with latent variable problems
- Iterative, alternating between two steps:
  - **E**-step: estimate latent variable (probabilities) based on current model
  - **M**-step: update the parameters of $p(x|z)$
  - Note similarity to k-means clustering.



Jake VanderPlas

GMM EM: E-Step

- Let us write down the formulas.
- **E-step**: fix parameters, compute posterior:

$$w_j^{(i)} = p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

- These w's are "soft" assignments of the z terms, i.e. probabilities over the values z could take. Concretely:

$$w_j^{(i)} = p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{\ell=1}^{k} p(x^{(i)} | z^{(i)} = \ell; \mu, \Sigma) p(z^{(i)} = \ell; \phi)}$$

GMM EM: M-Step

- Let's write down the formulas.
- **M-step**: fix w, update parameters:

$$\phi_j = \frac{1}{n}\sum_{i=1}^{n} w_j^{(i)}$$

Soft version of our counting estimator for the supervised case.

$$\mu_j = \frac{\sum_{i=1}^{n} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n} w_j^{(i)}}$$

Soft version of our empirical mean and covariances.

$$\Sigma_j = \frac{\sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} w_j^{(i)}}$$

Break & Quiz

Q: True or False: the M step of EM directly optimizes the log-likelihood

Answer: sort of. Directly we're maximizing a lower bound, but it can be shown that improving it improves the likelihood.

$$\mathcal{L}(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} Q_j^{(i)} \log \left(\frac{p_\theta(x^{(i)}, z^{(i)} = j)}{Q_j^{(i)}} \right)$$

Outline

- **Clustering Review**
  - k-means, hierarchical, spectral clustering
- **GMMs Review**
  - Gaussian Mixtures, EM algorithm
- **Principal Components Analysis**
  - Definition, Algorithm, Interpretations, Analysis, Applications

High-Dimensional Data

High-dimensions = lots of features

We've seen this repeatedly, but some examples:

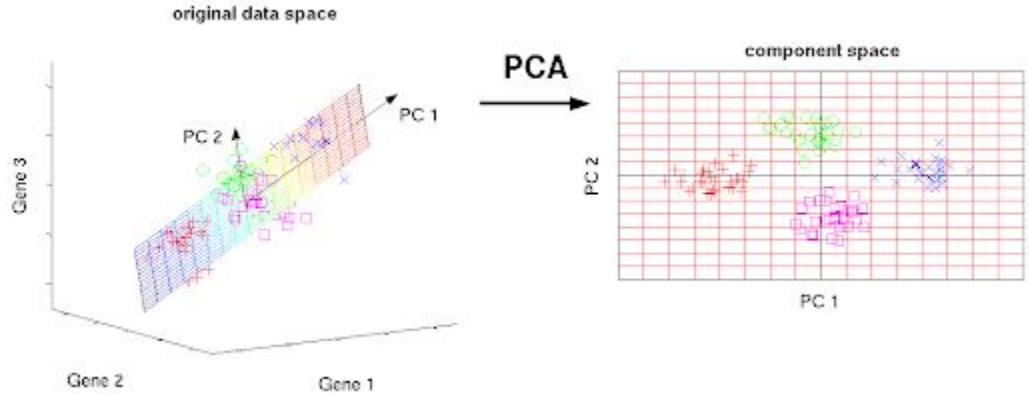
- Document classification
  - Features per document = thousands of words/unigrams, millions of bigrams, contextual information
- Surveys - Netflix

  480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

Dealing with Dimensionality

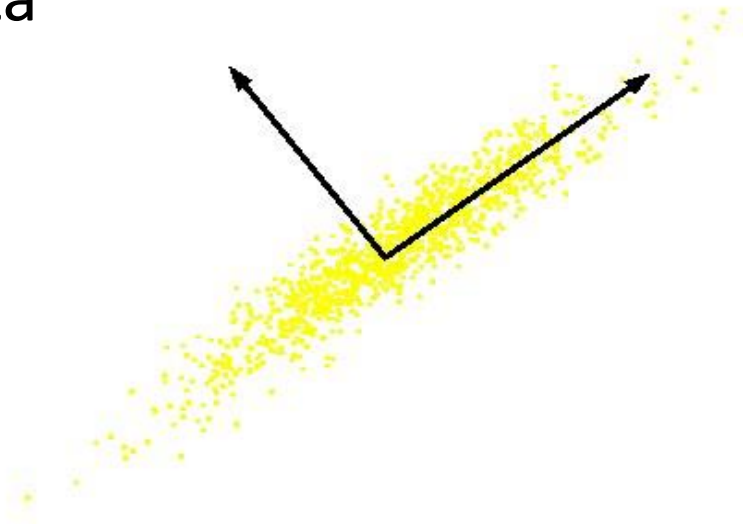
- **PCA, Kernel PCA, ICA**: Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.
- Some uses:
  - Visualization
  - More efficient use of resources (e.g., time, memory, communication)
  - Noise removal (improving data quality)
  - Further processing by machine learning algorithms (representation transfer)



Principal Components Analysis

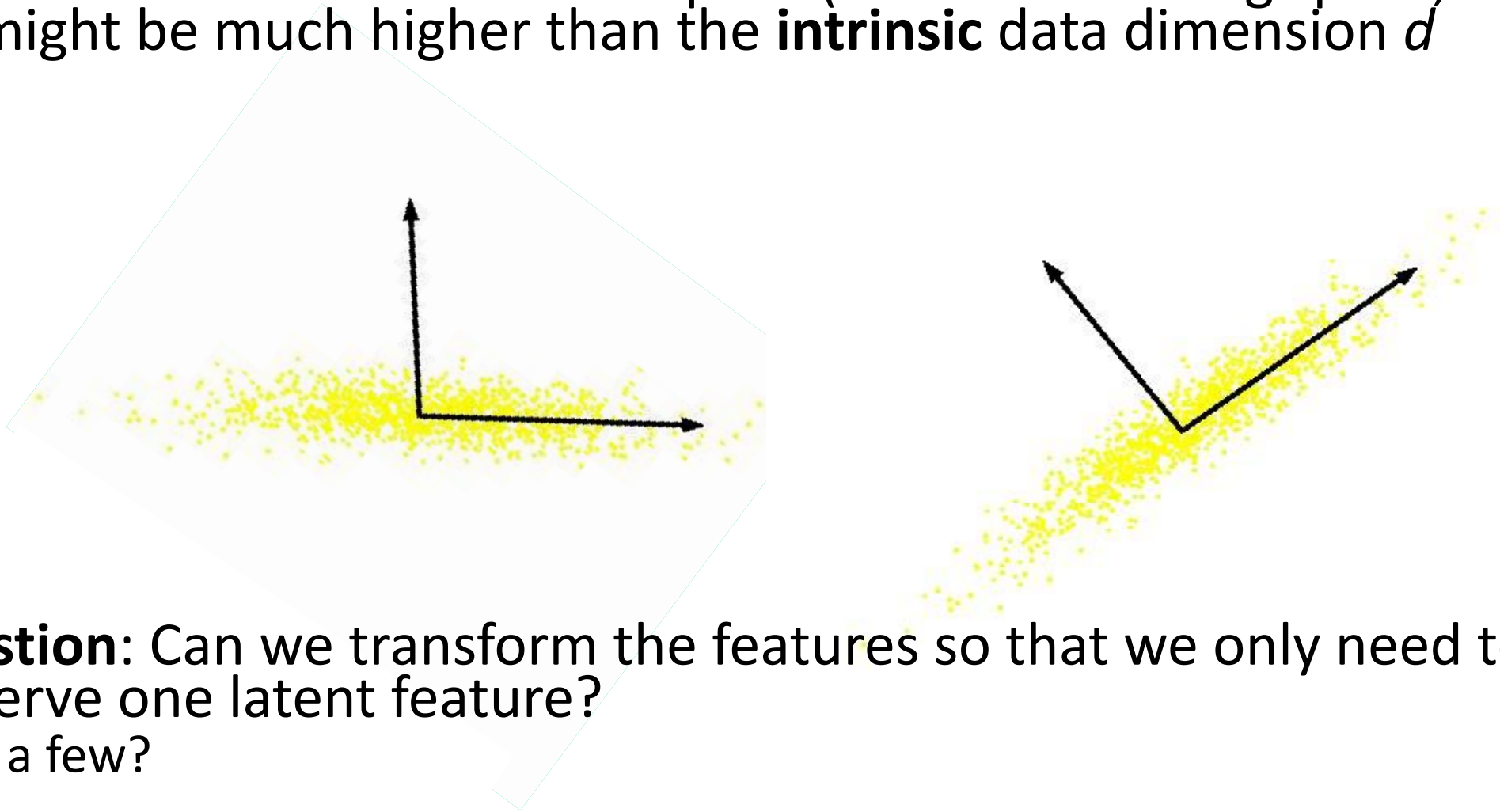
- Unsupervised technique for extracting variance structure from high dimensional datasets
  - also reduces dimensionality

- PCA: orthogonal projection / transformation of the data
  - Into a (possibly lower dimensional) subspace
  - Goal: maximize variance of the projected data



PCA Intuition

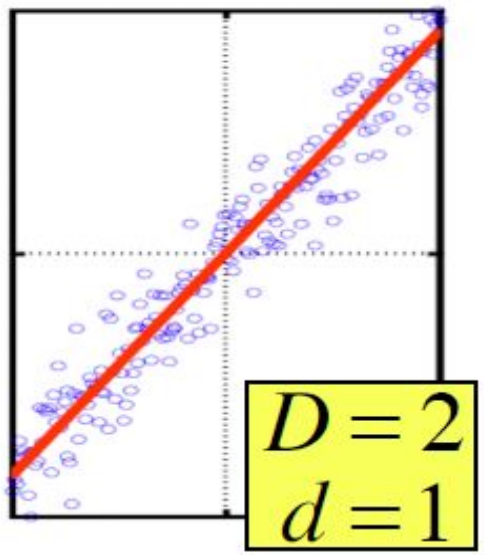
- The dimension of the ambient space (a.k.a. embedding space, ie, R^D) might be much higher than the **intrinsic** data dimension *d*

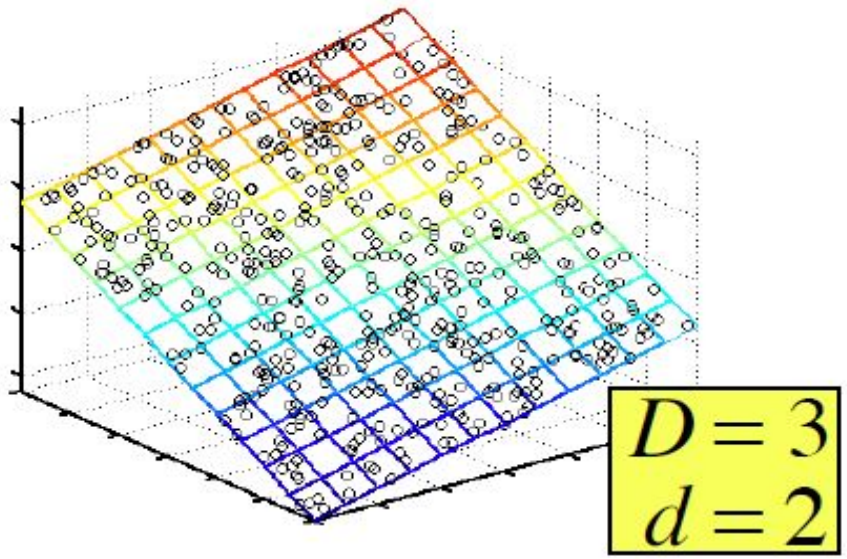


- **Question**: Can we transform the features so that we only need to preserve one latent feature?
  - Or a few?

PCA Intuition

- Some more visualizations

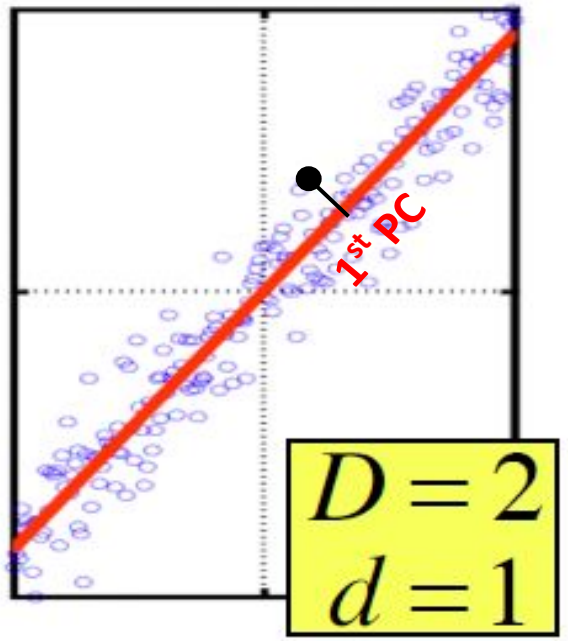




- In case where data lies on or near a low d-dimensional linear subspace, axes of this subspace are an effective representation of the data.

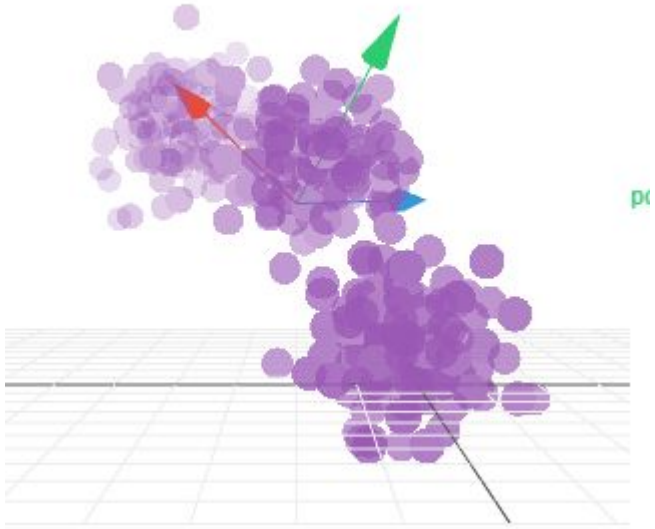
PCA: Principal Components

- **Principal Components (PCs)** are orthogonal directions that capture most of the variance in the data.
  - First PC – direction of greatest variability in data.
  - Projection of data points along first PC discriminates data most along any one direction



PCA: Principal Components and Projection

- How does dimensionality reduction work? From *d* dimensions to *r* dimensions:
  - Get orthogonal $v_1, v_2, \ldots, v_r \in \mathbb{R}^d$
- Maximizing variability
  - Equivalent to **minimizing reconstruction error**
- Then project data onto PCs $\longrightarrow$ d-dimensional



Victor Powell

PCA Approach Overview

- Want directions/components (unit vectors) so that
  - Projecting data maximizes variance
  - Specifically, for centered data (*xi=xi-mu*, *X* size is n*d)

$$\sum_{i=1}^{n} \langle x_i, v \rangle = \|Xv\|^2$$

- Do this **recursively**
  - Get orthogonal directions $v_1, v_2, \ldots, v_r \in \mathbb{R}^d$

PCA First Step

- First component,

$$v_1 = \arg\max_{\|v\|=1} \sum_{i=1}^{n} \langle v, x_i \rangle^2$$

- Same as getting

$$v_1 = \arg\max_{\|v\|=1} \|Xv\|^2$$

PCA Recursion

- Once we have *k-1* components, next?

$$\hat{X}_k = X - \sum_{i=1}^{k-1} X v_i v_i^T$$

deflation

- Then do the same thing

$$v_k = \arg\max_{\|v\|=1} \|\hat{X}_k w\|^2$$

PCA Interpretations

- The v's are eigenvectors of X^TX
  - We'll see why in a second

- X^TX (proportional to) sample covariance matrix
  - When data is 0 mean!
  - i.e. PCA is the eigendecomposition of sample covariance

- Nested subspaces *span(v1), span(v1,v2)*,…,

PCA Interpretations: First Component

- Two specific ways to think about the first component
- **Maximum variance direction**
  - What we saw so far

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

- **Minimum reconstruction error**
  - A direction so that projection yields minimum MSE in reconstruction

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

PCA Interpretations: Equivalence

- Interpretation 1.

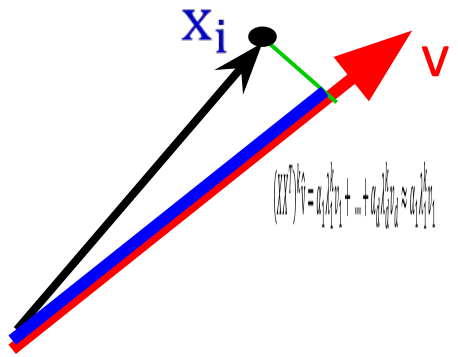
Maximum variance direction

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

- Interpretation 2.

Minimum reconstruction error

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



- Why are these equivalent?
  - Use Pythagorean theorem.
  - Maximizing **blue** segment is the same as minimizing the **green**

PCA Gram Matrix Interpretation

- Recall our first PC, maximized variance:

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

- Constrained optimization (Lagrangian + KKT conditions)

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

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

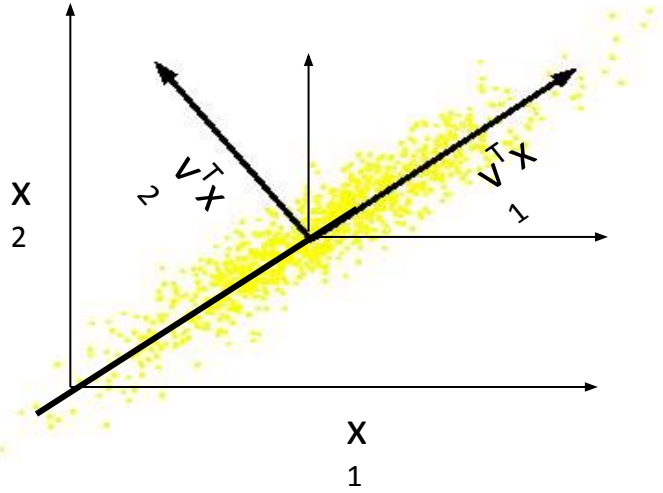
PCA Covariance Matrix Interpretation

- So $\Rightarrow (X^T X \quad \mathbf{v} = \lambda \mathbf{v}$

  - means that v (the first PC) is an eigenvector of $X^T X$
  - eigenvalue λ denotes the amount of variability captured along that dimension

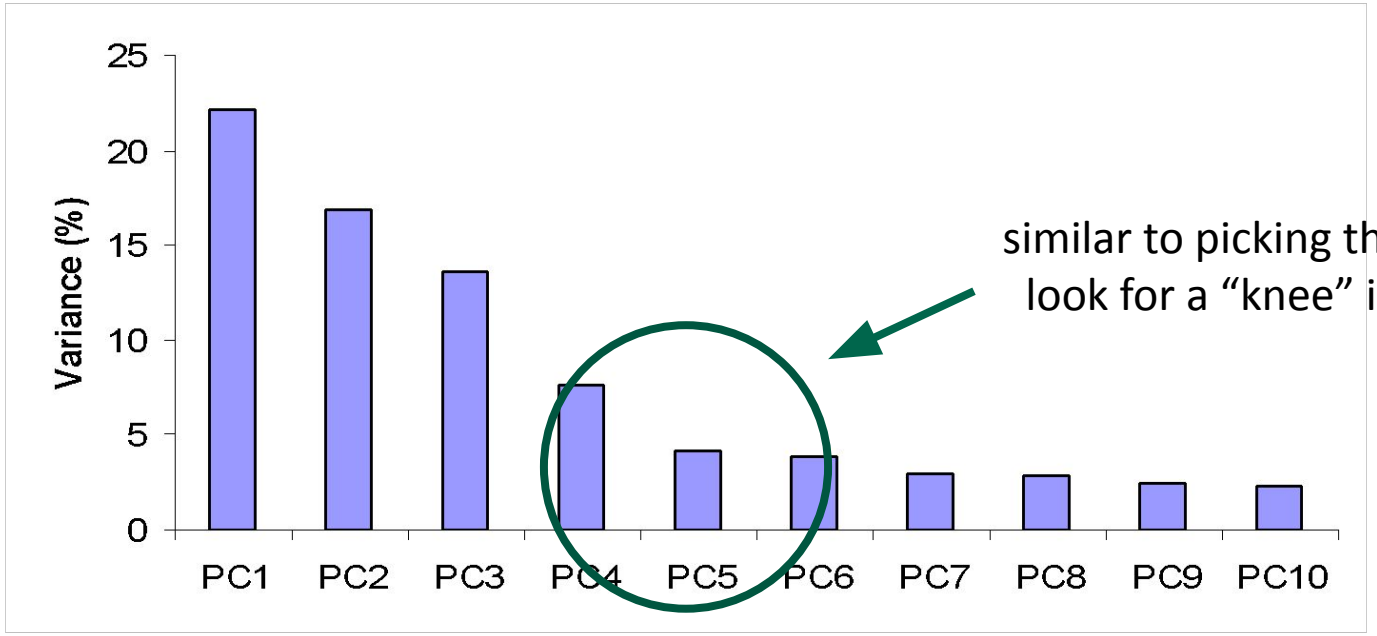
- PCs are just the eigenvectors…
  - How to find them? Eigendecomposition

- Don't need to keep all eigenvectors
  - Just the ones for largest eigenvalues



PCA Dimensionality Reduction

- In high-dimensional problems, data sometimes lies near a linear subspace, as noise introduces small variability
- Only keep data projections onto principal components with **large** eigenvalues
- Can *ignore* the components of smaller significance.



Applications of PCA

- Visualization
- **More efficient use of resources (e.g. time, memory, communication)**
- Noise removal (improving data quality)
- Further processing by machine learning algorithms (representation transfer)

Application: Image Compression

- Start with image; divide into 12x12 patches
  - i.e. 144-D vector
  - **Original image:**

Application: Image Compression

- Project to 6D:

Compressed

Original

Applications of PCA

- Visualization
- More efficient use of resources (e.g. time, memory, communication)
- Noise removal (improving data quality)
- **Further processing by machine learning algorithms (representation transfer)**

PCA representations for supervised learning

- PCA can be used to obtain low-dimensional representations of words
  - Also known as word embeddings / word vectors
  - Can be applied on downstream supervised tasks like document classification

- Many more powerful text embedding methods have been introduced
  - GloVe / word2vec for individual words
  - BERT / later LLMs embed the entire document

Break & Quiz

Q: Are these statements true or false?
(A) The principal component with the largest eigenvalue maximizes the reconstruction error.
(B) The dimension of original data representation is always higher than the dimension of transformed representation of PCA.

1. True, True
2. True, False
3. False, True
4. False, False

Q2-2: Are these statements true or false?
(A) The principal component with the largest eigenvalue maximizes the reconstruction error.
(B) The dimension of original data representation is always higher than the dimension of transformed representation of PCA.

1. True, True
2. True, False
3. False, True
4. False, False ⟵

(A) The principal component with the largest eigenvalue captures the maximum amount of variability which is equivalent to minimum reconstruction error.
(B) If the matrix XX^T is full-rank, they can be of the same dimension.

Thanks Everyone!

Some of the slides in these lectures have been adapted/borrowed from materials developed by Misha Khodak, Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, Pedro Domingos, Jerry Zhu, Yingyu Liang, Volodymyr Kuleshov, Kirthy Kandasamy