香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP 5212
Machine Learning
Lecture 13

Dimensionality Reduction

Junxian He
Mar 15, 2024

Midterm Exam

- Location: 2465 (Lift 25-26)
- Next Wed (March 20), 3pm-420pm

Recap: The General EM Algorithm

Repeat until convergence {

(E-step) For each i, set

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta).$$

Based on current θ, model parameters does not change in E-step

(M-step) Set

$$\theta := \arg\max_\theta \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i, \theta)$$

$$= \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}.$$

$Q(z)$ is not relevant to θ, and $Q(z)$does not change in the M-step

}

E-step is maximizing ELBO over Q(z), M-step is maximizing ELBO overθ

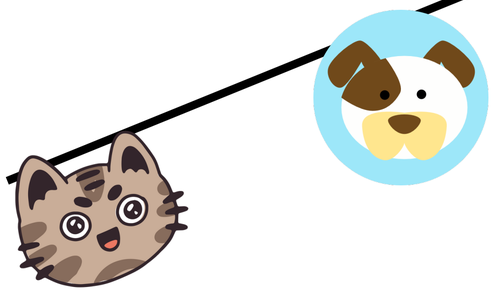
Why is maximizing lower-bound sufficient?

Recap: EM is Hill Climbing

$\log p(x;\theta)$ Only related to θ, no z

ELBO

Larger

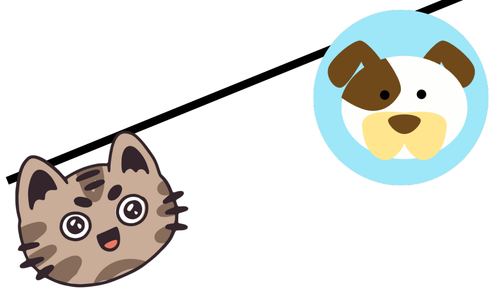


Recap: EM is Hill Climbing

$\log p(x;\theta)$

ELBO

Larger



E-step: $Q(z) = p(z \mid x;\theta)$, making ELBO tight

"dog" doesn't change, because θ does not change

Recap: EM is Hill Climbing

$\log p(x;\theta)$

ELBO

Larger

M-step: $\max_{\theta} ELBO$

ELBO becomes larger, and it is not tight anymore because posterior changes

Recap: EM is Hill Climbing

$\log p(x;\theta)$

ELBO

Recap: EM is Hill Climbing

$\log p(x;\theta)$

ELBO

E-step: $Q(z) = p(z \mid x;\theta)$, making ELBO tight

"dog" doesn't change, because θ does not change

Recap: EM is Hill Climbing

$\log p(x;\theta)$

ELBO

M-step: $\max_{\theta} ELBO$

ELBO becomes larger, and it is not tight anymore because posterior changes

$\log p(x;\theta)$ is monotonically increasing!

We are doing MLE implicitly!

Convergence is guaranteed

High-Dimensional Data

- High-Dimensions = Lot of Features

Document classification

Features per document =

thousands of words/unigrams

millions of bigrams, contextual information

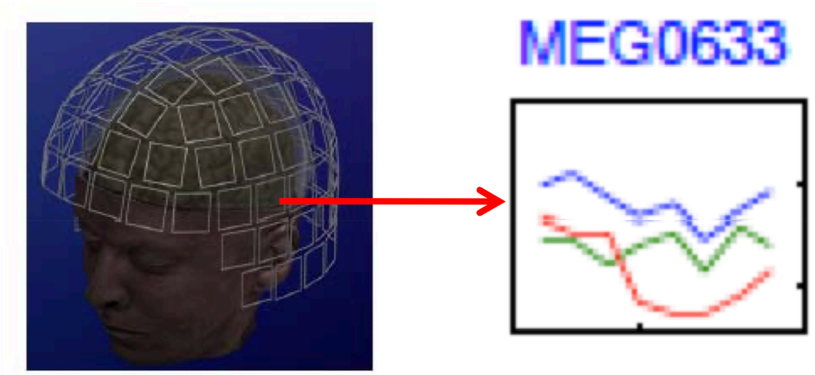
High-Dimensional Data

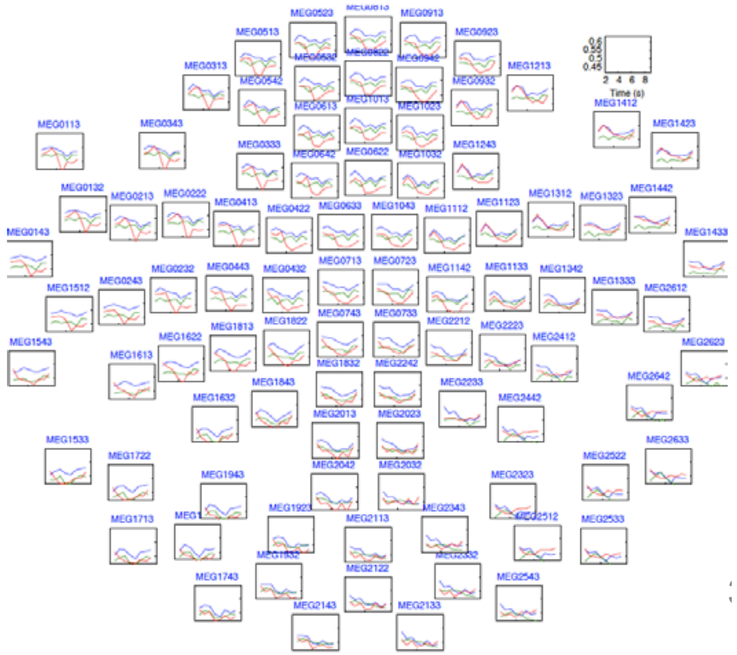
- High-Dimensions = Lot of Features

MEG Brain Imaging

120 locations x 500 time points

x 20 objects

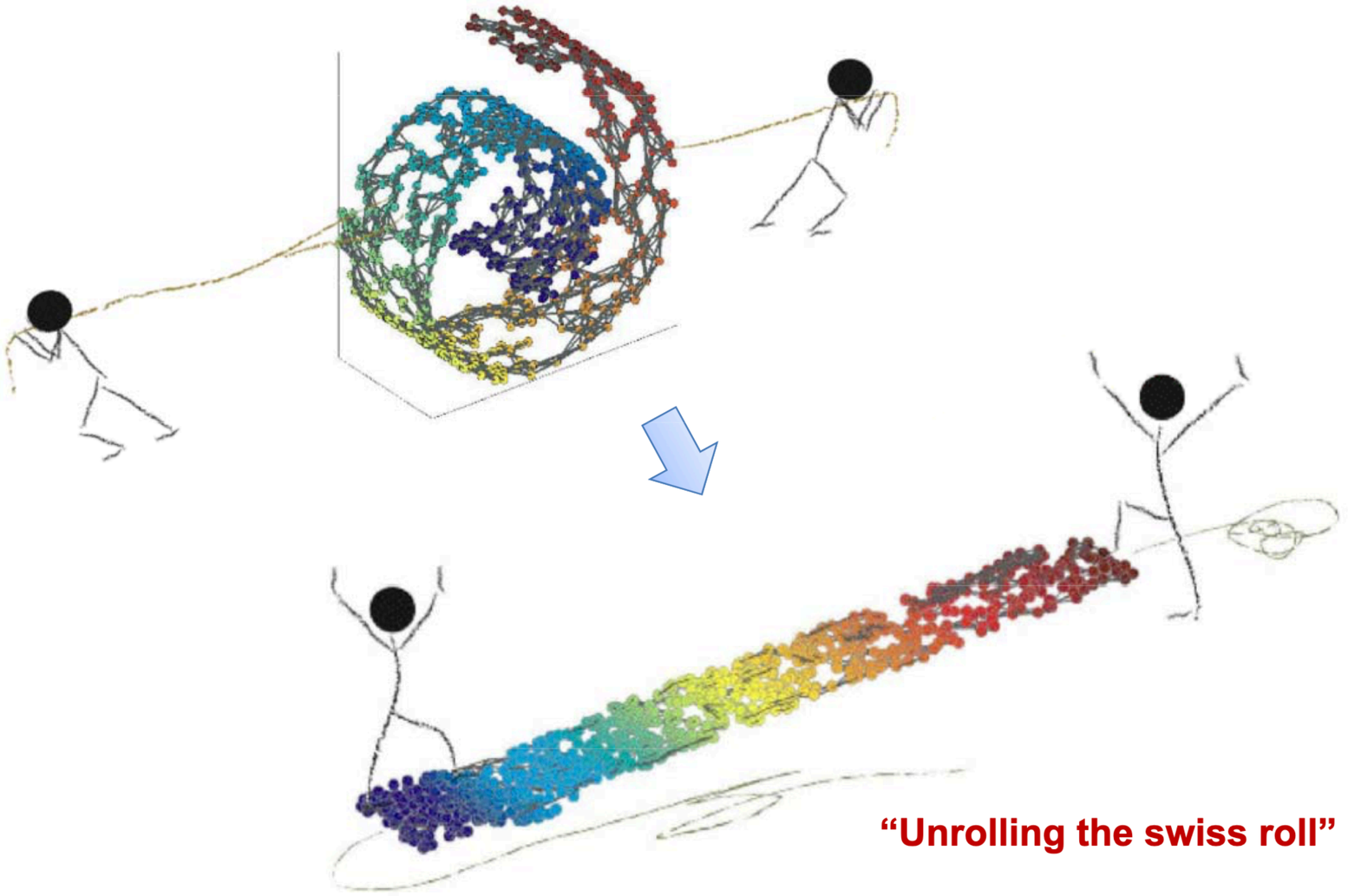




Curse of Dimensionality

- Why are more features bad?
  - Redundant features (not all words are useful to classify a document) more noise added than signal
  - Hard to store and process data (computationally challenging)
  - Hard to interpret and visualize
  - Complexity of decision rule tends to grow with # features

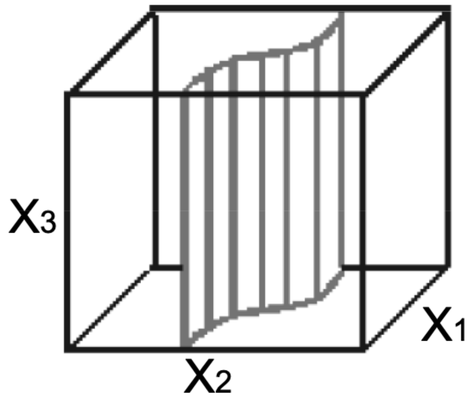
Dimensionality Reduction



“Unrolling the swiss roll”

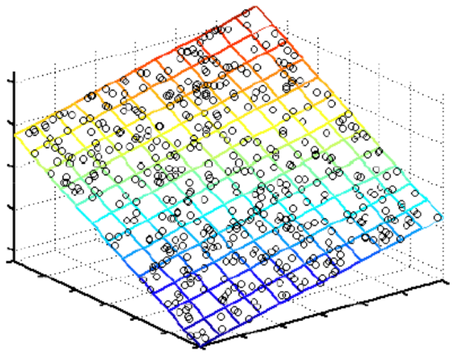
Dimensionality Reduction

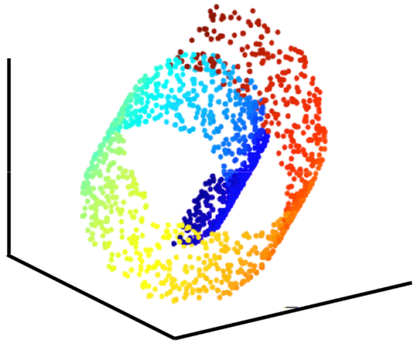
- Feature Selection – Only a few features are relevant to the learning task



X_3 - Irrelevant

- Latent features – Some linear/nonlinear combination of features provides a more efficient representation than observed features





Latent Feature Extraction

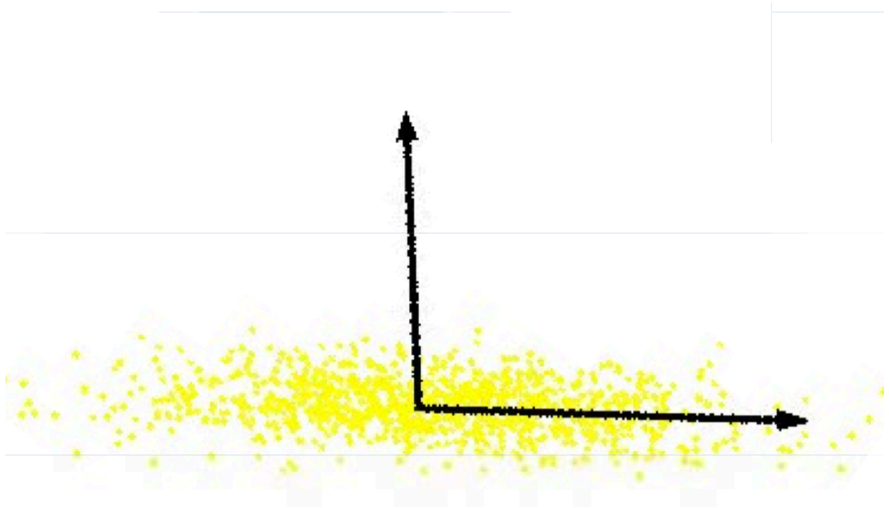
Combinations of observed features provide more efficient representation, and capture underlying relations that govern the data

E.g. Ego, personality and intelligence are hidden attributes that characterize human behavior instead of survey questions

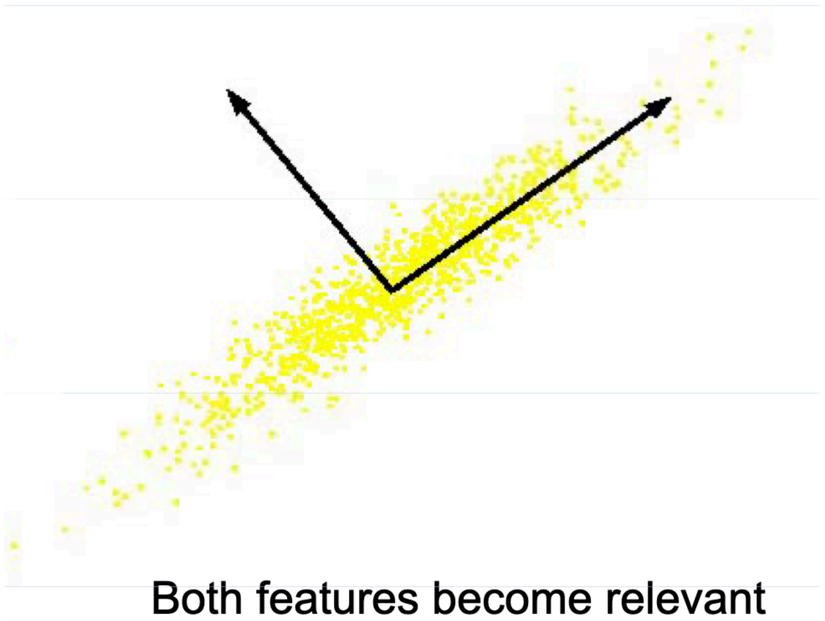
Topics (sports, science, news, etc.) instead of documents

- Linear
  - Principal Component Analysis (PCA)
  - Factor Analysis
  - Independent Component Analysis (ICA)
- Nonlinear
  - ISOMAP
  - Local Linear Embedding (LLE)
  - Laplacian Eigenmaps

Principal Component Analysis (PCA)



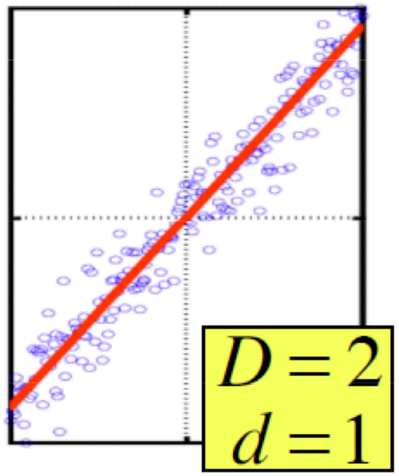
Only one relevant feature

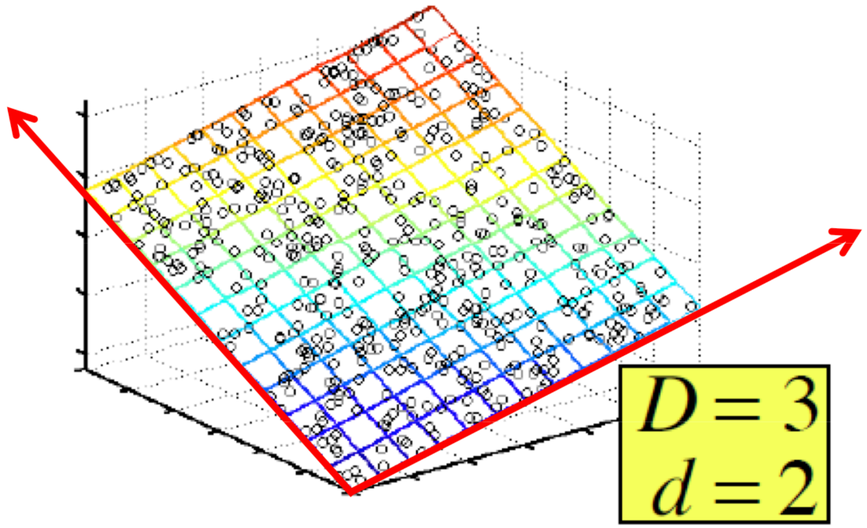


Both features become relevant

Can we transform the features so that we only need to preserve one latent feature? Find linear projection so that projected data is uncorrelated.

Principal Component Analysis (PCA)



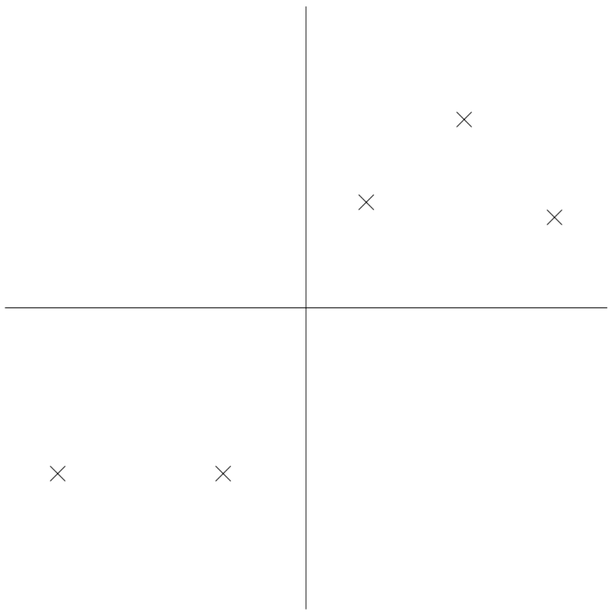


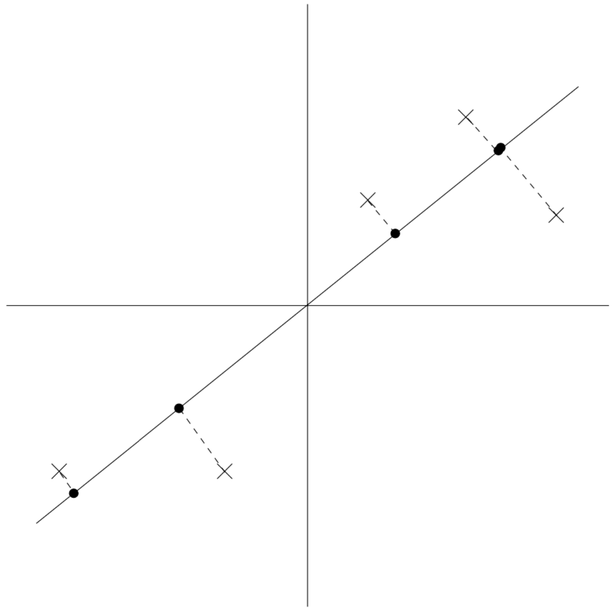
Assumption: Data lies on or near a low d-dimensional linear subspace.

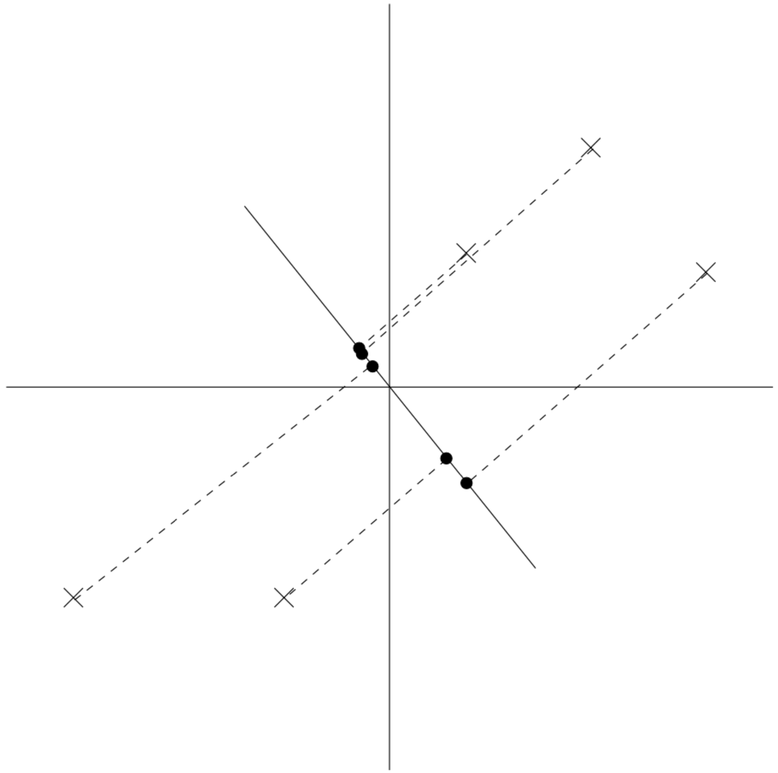
Axes of this subspace are an effective representation of the data

Identifying the axes is known as Principal Components Analysis, and can be obtained by Eigen or Singular value decomposition

Principal Component Analysis (PCA)







Project the data onto different directions

Which projection is better?

We want the low-dim features that can discriminate the data the most

Normalizing Data

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_j^{(i)} \qquad \sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} - \mu_j)^2$$

Different features may have different scales

After normalization, each feature has 0 mean and variance 1

Principal Component Analysis (PCA)

Let v be the principal component

Find vector that maximizes sample variance of projection

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

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

Lagrangian: $\max_{\mathrm{v}} \mathrm{v}^T X X^T \mathrm{v} - \lambda(\mathrm{v}^T\mathrm{v} - 1)$

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

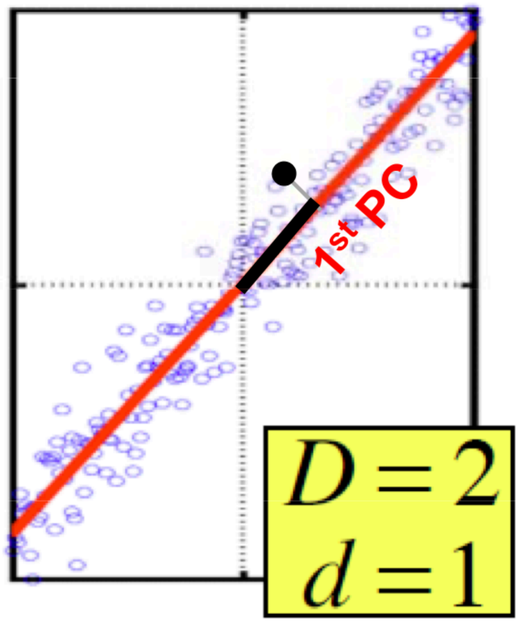
Definition of eigenvectors

K-dimensional Cases

If we project our data into a k-dimensional subspace (k<d), we should choose $v_1, v_2, \ldots, v_k$ to be the top k eigenvectors of XX^T

For symmetric matrices, eigenvectors for distinct eigenvalues are orthogonal

Principal Component Analysis (PCA)



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

1st PC – direction of greatest variability in data

Projection of data points along 1st PC discriminate the data most along any one direction

Principal Component Analysis (PCA)

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

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

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

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

And so on …

Computing the Principal Components (PCs)

Eigenvectors are solutions of the following equation:

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

Non-zero solution v ≠ 0 possible only if

$$\det(\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I}) = 0$$

We can compute the eigenvalues from this equation

This is a D^{th} order equation in λ, can have at most D distinct solutions (roots of the characteristic equation)

Once eigenvalues are computed, solve for eigenvectors (Principal Components) using

$$(\mathbf{X}\mathbf{X}^T - \lambda\mathbf{I})\mathbf{v} = 0$$

Another Interpretation

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

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

Dimensionality Reduction using PCA

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

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

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

Original Representation

data point

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

(D-dimensional vector)

Transformed representation

projections

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

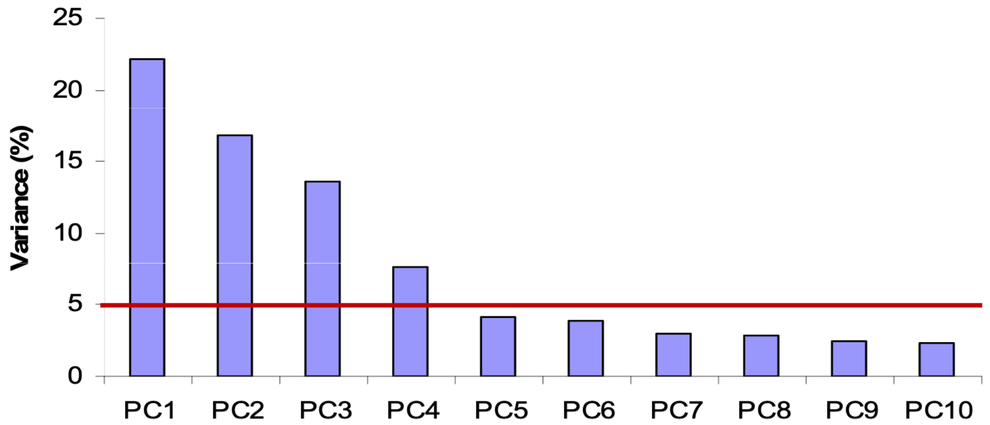
(d-dimensional vector)

Dimensionality Reduction using PCA

Usually data lies near a linear subspace, as noise introduces small variability

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

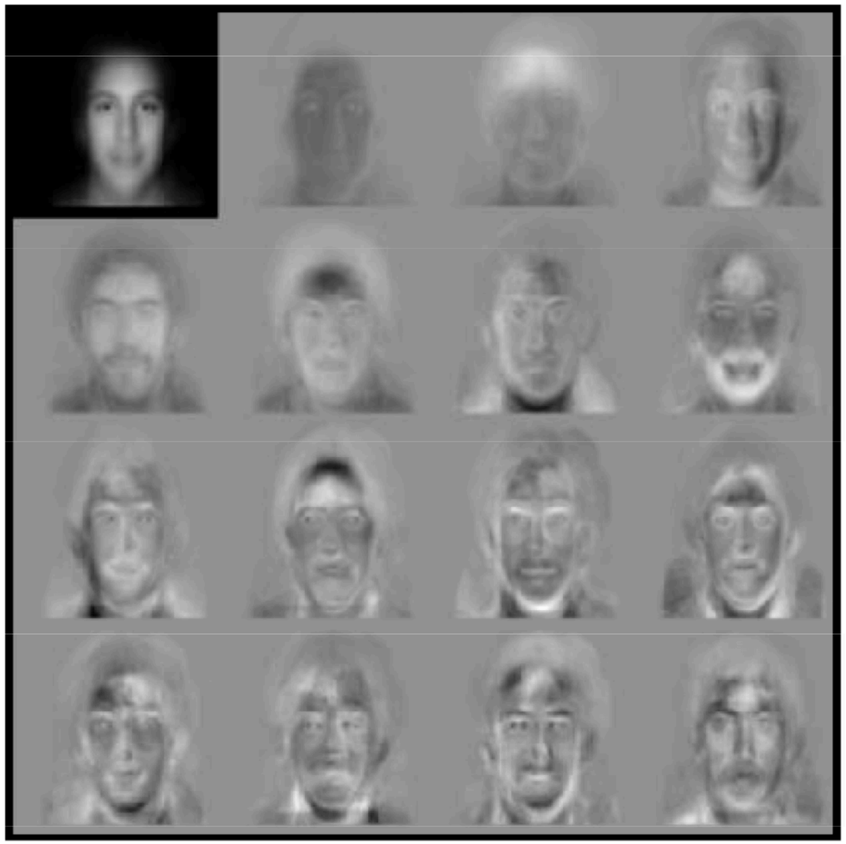
Can *ignore* the components of lesser significance.



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

It is not lossless compression

Example: faces



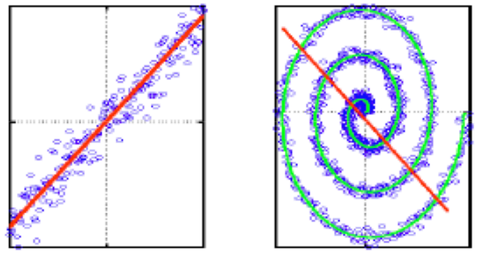
Eigenfaces from 7562 images:

top left image is linear combination of rest.

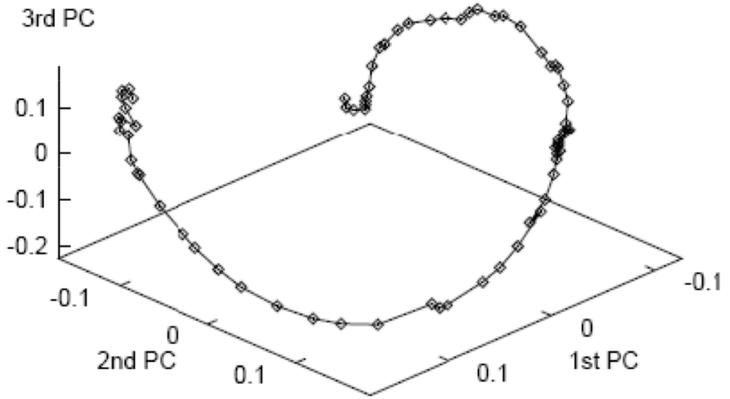
Sirovich & Kirby (1987)
Turk & Pentland (1991)

Properties of PCA

- **Strengths**
  - Eigenvector method
  - No tuning parameters
  - Non-iterative
  - No local optima
- **Weaknesses**
  - Limited to second order statistics
  - Limited to linear projections



Nonlinear example



Thank You!
Q & A