Topics in Graphical Models

$p(X_1 .. X_n) = \prod_{i=1}^{n} p(X_i | pa(X_i))$

- Representation
  – Which joint probability distributions does a graphical model represent?

conditional independence
- local Markov assump
- D-separation
- Markov Blanket

- Inference
  – How to answer questions about the joint probability distribution?
    - Marginal distribution of a node variable
    - Most likely assignment of node variables

$p(x_1, x_3)$ or $p(x_2 | x_4)$

Variable elimination

$\max_{x_1, x_3} p(x_1, x_3)$

- Learning
  – How to learn the parameters and $\boxed{\text{structure}}$ of a graphical model?

$D = \{X_1^{(i)} ... X_n^{(i)}\}_{i=1}^{m}$ – m data points

Max Likelihood score for graph structure

$D = \{x_1^{(j)} \ldots x_n^{(j)}\}_{j=1}^{m}$ m - data points, n - # variables

$\hat{\theta}_G$ - MLE or MAP estimates of parameters given G

ML score for graph structure $\mathcal{G}$

$$\underbrace{\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G})} = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

$\hat{P}(D \mid \hat{\theta}_G, G) = \prod_{j=1}^{m} \underbrace{P(x_1^{(j)}, \ldots, x_n^{(j)})}_{\prod_{i=1}^{n} p(x_i^{(j)} \mid pa(x_i^{(j)}))}$

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log \widehat{P}\left(x_i^{(j)} \mid \mathbf{x}^{(j)}_{\mathbf{Pa}_{X_i}}\right)$$

$$= m \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \widehat{P}(x_i, \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log \widehat{P}\left(x_i \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

$H(z) = -\sum_{z} p(z) \log p(z)$

$$= -m \sum_{i=1}^{n} \widehat{H}(X_i \mid \mathbf{Pa}_{X_i})$$

$$= m \sum_{i=1}^{n} [\widehat{I}(X_i, \mathbf{Pa}_{X_i}) - \underbrace{\widehat{H}(X_i)}]$$

Doesn't depend on graph structure $\mathcal{G}$

ML score is Decomposable

- Log data likelihood

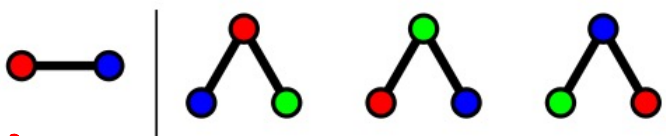
$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = m \sum_{i=1}^{n} [\widehat{I}(X_i, \mathbf{Pa}_{X_i}) - \widehat{H}(X_i)]$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!

How many trees are there?

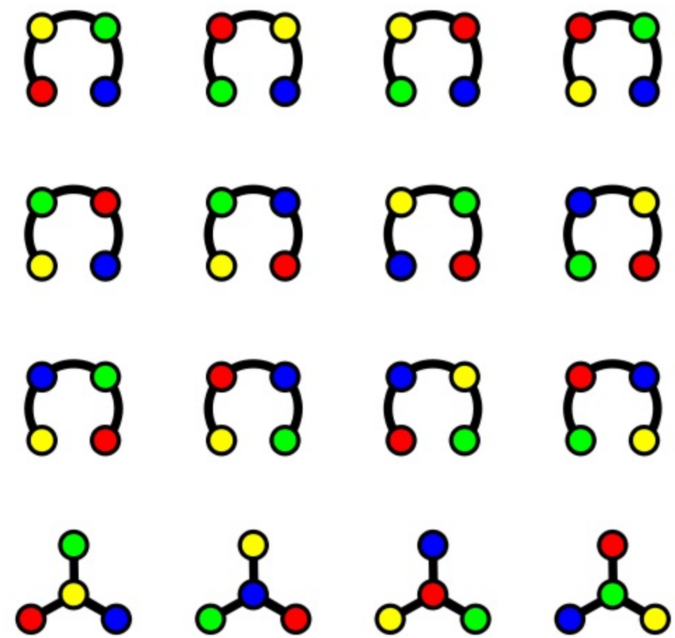
- Trees – every node has at most one parent
- n^{n-2} possible trees (Cayley's Theorem)

n=2 n=3



$2^{2-2} = 2^0 = 1$ $3^{3-2} = 3$

n=4



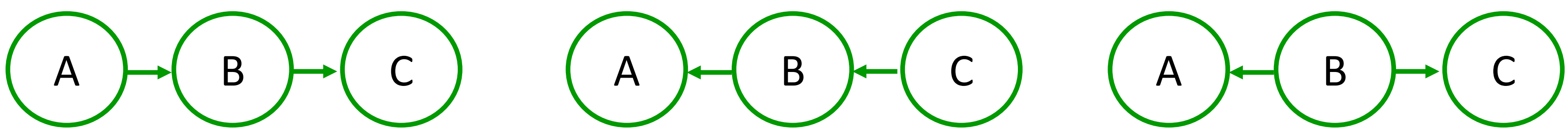
$4^{4-2} = 16$

Nonetheless – Efficient optimal algorithm finds best tree!

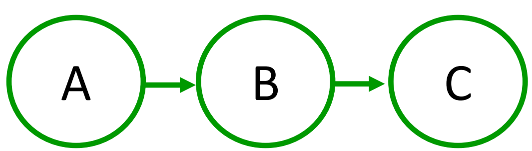
Scoring a tree

$$\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

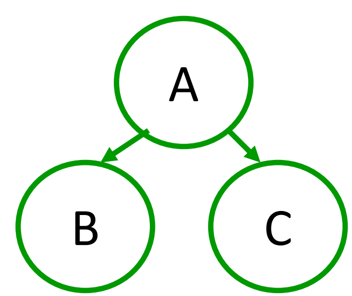
Equivalent Trees (same score): I(A,B) + I(B,C)



Score provides indication of structure:



I(A,B) + I(B,C)



I(A,B) + I(A,C)

Chow-Liu algorithm

$I(X_i, pa(X_i))$

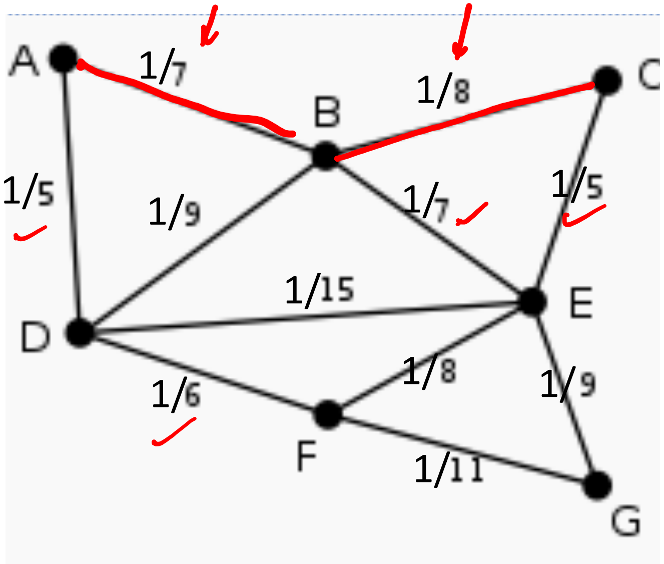
- For each pair of variables X_i, X_j
  - Compute empirical distribution: $\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$ ✓
  - Compute mutual information:

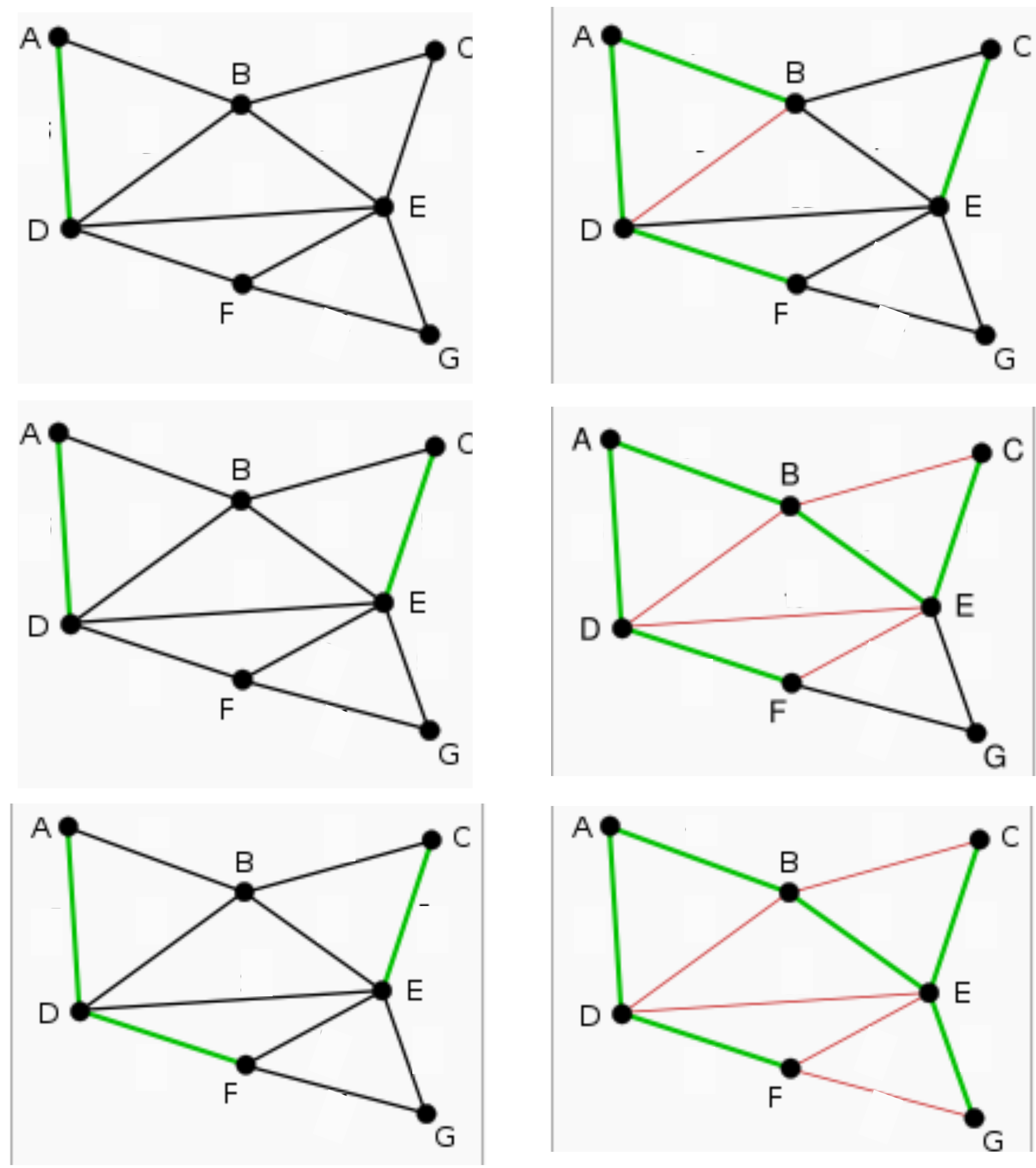
$$\widehat{I}(X_i, X_j) = \sum_{x_i, x_j} \widehat{P}(x_i, x_j) \log \frac{\widehat{P}(x_i, x_j)}{\widehat{P}(x_i)\widehat{P}(x_j)}$$

- Define a graph
  - Nodes $X_1,\ldots,X_n$
  - Edge (i,j) gets weight $\widehat{I}(X_i, X_j)$
- Optimal tree BN
  - Compute maximum weight spanning tree (e.g. Prim's, Kruskal's algorithm O(nlog n)) ✓✓
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Chow-Liu algorithm example

$\{A^{(j)} \ldots G^{(j)}\}_{j=1}^{m}$





Scoring general graphical models

- Graph that maximizes ML score -> complete graph!
- Information never hurts

  $H(A|B) \geq H(A|B,C)$

- Adding a parent always increases ML score

  $I(A,B,C) \geq I(A,B)$

- The more edges, the fewer independence assumptions, the higher the likelihood of the data, but will overfit…
- Why does ML for trees work?

  Restricted model space – tree graph

Regularizing

- Model selection
  - Use MDL (Minimum description length) score
  - BIC score (Bayesian Information criterion)
- Still NP –hard

  Theorem: The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d>1* (Note: tree d=1)
- Mostly heuristic (exploit score decomposition)
- Chow-Liu: provides best tree approximation to any distribution.
- Start with Chow-Liu tree. Add, delete, invert edges. Evaluate BIC score

What you should know

- Learning BNs ← directed graphical model
  - Maximum likelihood or MAP learns parameters
  - ML score
    - Decomposable score
    - Information theoretic interpretation (Mutual information)
  - Best tree (Chow-Liu)
  - Other BNs, usually local search with BIC score

regularized ML score

Unsupervised Learning

Aka Learning without labels

$y \downarrow$

$p(x_1 \ldots x_n)$

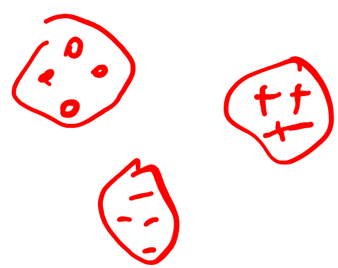
- Learning and inference using probability distributions & densities
    - MLE/MAP
    - Graphical models

- Dimensionality Reduction

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} \longrightarrow \tilde{X} = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_d \end{bmatrix}$$

$d << D$

- Clustering



Dimensionality Reduction
PCA

Aarti Singh

Machine Learning 10-701
April 19, 2023

Slides Courtesy: Tom Mitchell, Eric Xing, Lawrence Saul

High-Dimensional data

- High-Dimensions = Lot of Features

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

High-Dimensional data

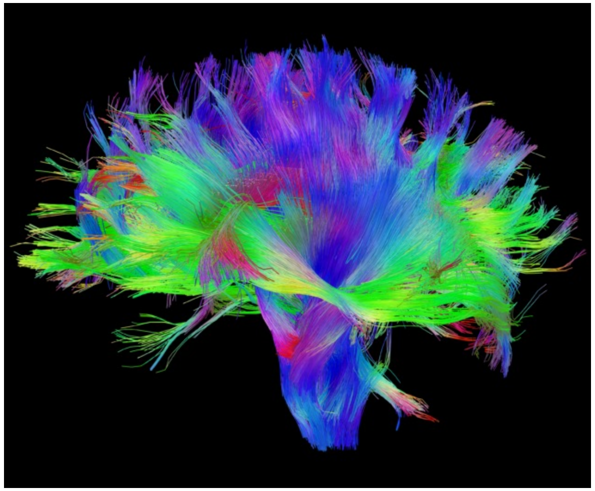
- High-Dimensions = Lot of Features

High resolution images

millions of pixels

Diffusion scans of Brain

300,000 brain fibers

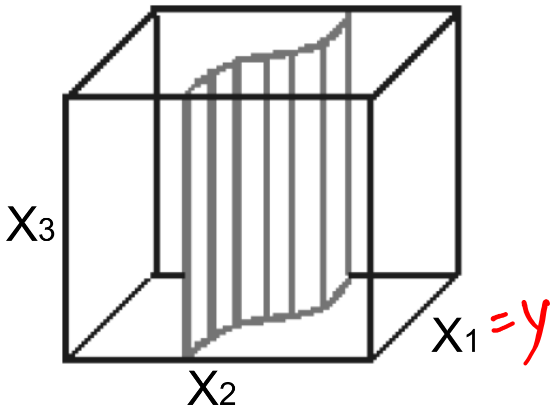


Curse of Dimensionality

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

Dimensionality Reduction

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

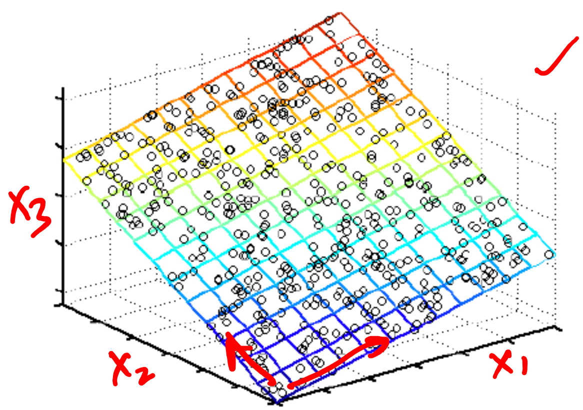


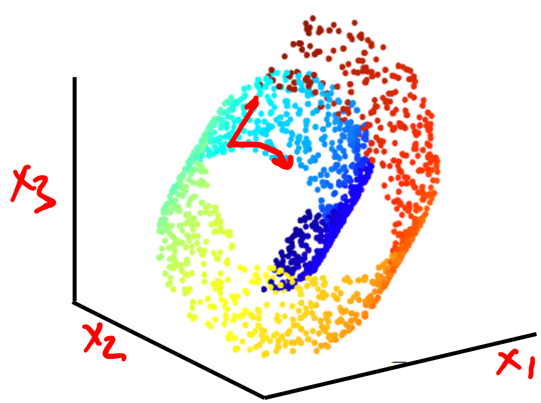
X3 - Irrelevant

l_1 penalty
$\parallel$
lasso

$y \leftarrow w_2 x_2 + w_3 x_3$

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





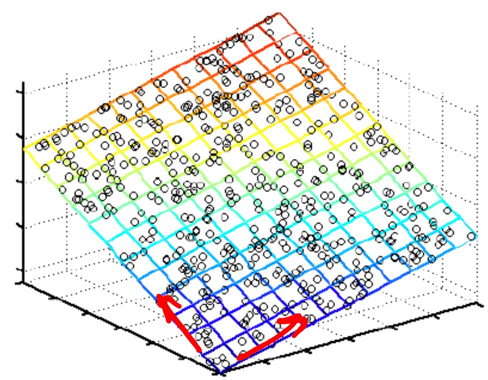
Latent Features

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

Often may not have physical meaning

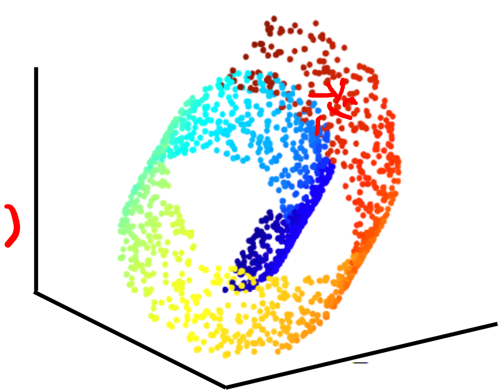


- Linear
  - **Principal Component Analysis (PCA)** ✓
  - Factor Analysis
  - Independent Component Analysis (ICA)
- Nonlinear
  - Kernel PCA
  - Laplacian Eigenmaps, ISOMAP, LLE
  - Autoencoders

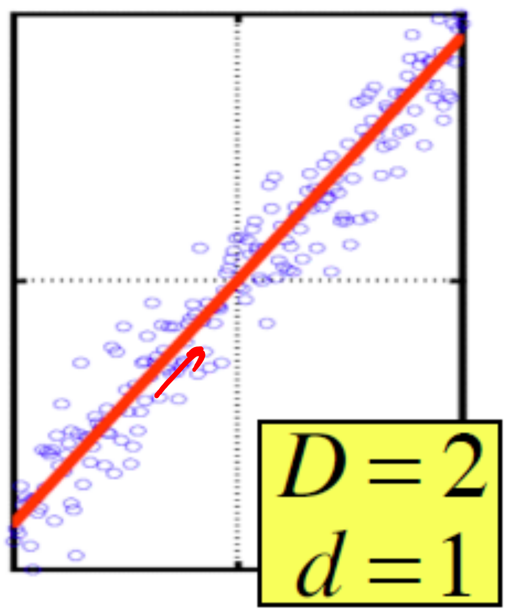
$d \ll D$

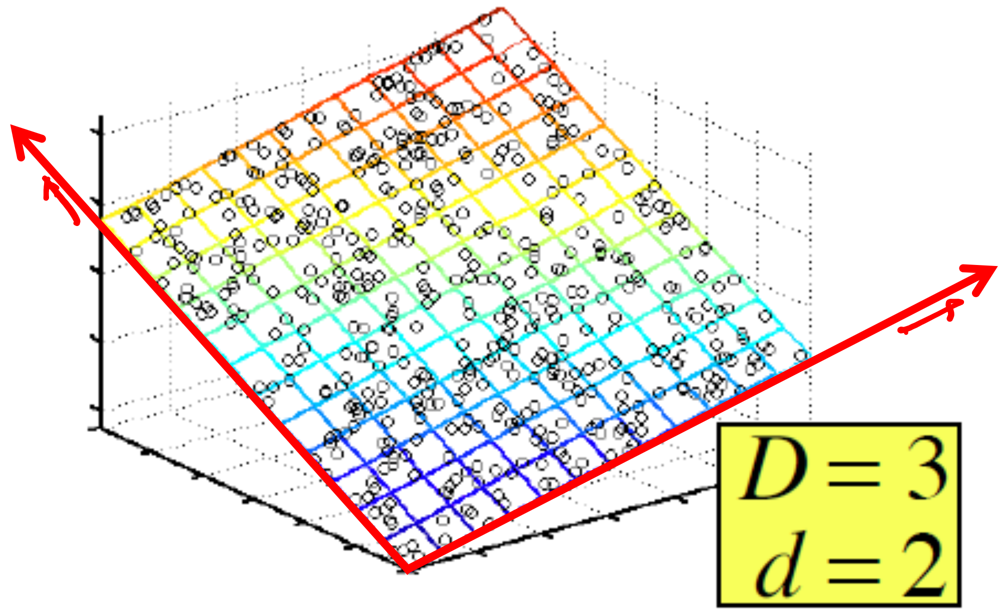
x $\hat{x}$

$MSE(x, \hat{x})$



Principal Component Analysis (PCA)





When 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

Data for PCA

Data X = $[x_1, x_2, \ldots, x_n]$ where each data point x_i is D-dimensional vector

X is D x n matrix

Assume data are centered i.e. sample mean $\frac{1}{n}\sum_{i=1}^{n} x_i = 0$

(1) What if data is not centered?

Subtract off sample mean from each data point ✓

Since data matrix is centered, sample covariance matrix can be written as

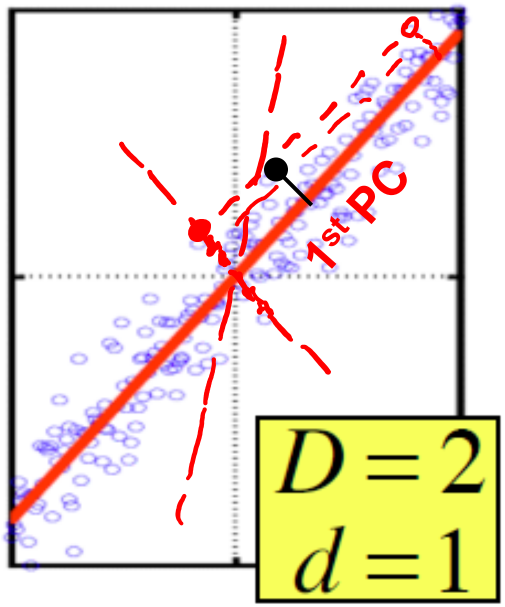
(2)

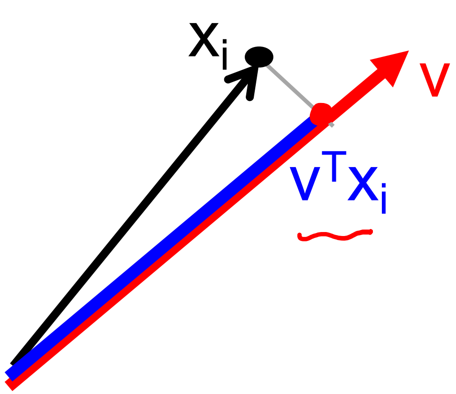
$$S = \frac{1}{n} X X^\top$$

$E[(z_i - Ez)(z_i - Ez)^T]$

$E \to \frac{1}{n}\sum_{i=1}^{n}$, $\quad Ez \to 0$, $\quad Ez \to \frac{1}{n}\sum_{i=1}^{n} z_i$, $\quad Ez \to 0$

Principal Component Analysis (PCA)





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

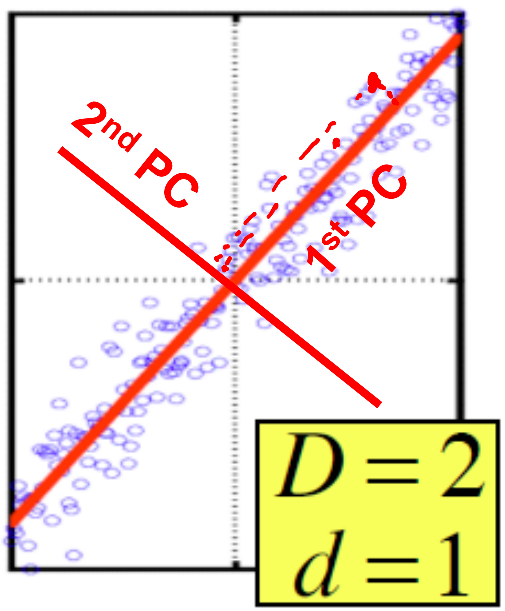
1st PC – direction of greatest variability in data

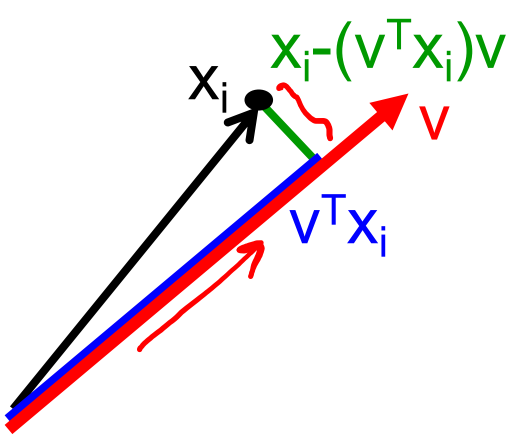
Projection of data points along 1st PC discriminate the data most along any one direction

Take a data point x_i (D-dimensional vector)

Projection of x_i onto the 1st PC v is $v^T x_i$

Principal Component Analysis (PCA)





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

1st PC – direction of greatest variability in data

2nd PC – Next orthogonal (uncorrelated) direction of greatest variability

(remove all variability in first direction, then find next direction of greatest variability)

And so on …

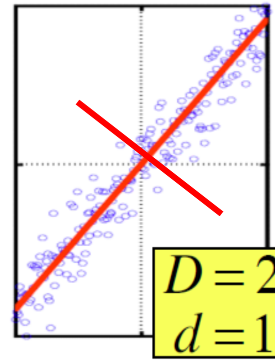
$D \to d \ll D$

Principal Component Analysis (PCA)

Let $v_1, v_2, \ldots, v_d$ denote the principal components

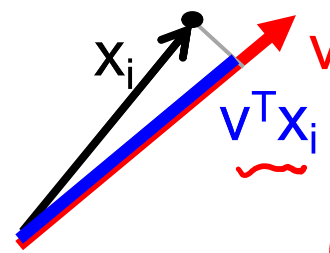
Orthogonal and unit norm $\quad v_i^T v_j = 0 \quad i \neq j$ ✓

$v_i^T v_i = 1$ ✓



Find vector that maximizes sample variance of projection

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \frac{\underset{1\times d}{\mathbf{v}^T}\,\boxed{\underset{d\times d}{\mathbf{X}\mathbf{X}^T}}\,\underset{d\times 1}{\mathbf{v}}}{n}$$



$E[z^2]$

$E[(z-Ez)^2]$

var(z)

$\frac{1}{n}\sum_{i=1}^{n} z_i^2$

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

$\sum_i v_i^2 = 1$

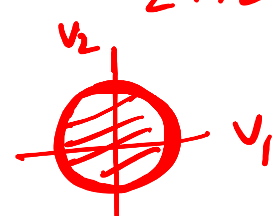
convex

non-convex set

Poll:

$z^T M z \geq 0$

➢ Is this a convex optimization problem?

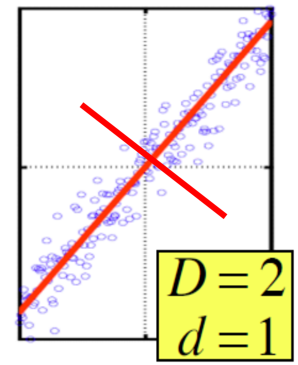


$\|v\| \leq 1$

$v^T v \leq 1$

Principal Component Analysis (PCA)

Let $v_1, v_2, \dots, v_d$ denote the principal components

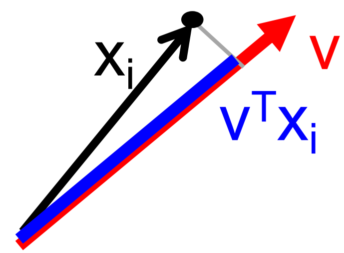


Orthogonal and unit norm $\quad v_i^T v_j = 0 \quad i \neq j$

$v_i^T v_i = 1$

Find vector that maximizes sample variance of projection

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



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

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

Wrap constraints into the objective function

$2XX^Tv - 2\lambda v = 0$

$\text{evec}(XX^T)$

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

$\text{eval}(XX^T)$

Sample var when projecting on v $\quad v^T X X^T v = v^T(\lambda v) = \lambda v^T v = \lambda$