

Revision Class

Song Liu (song.liu@bristol.ac.uk)

General Stuff

What might be given?

- 2 by 2 Matrix Inversion Formula.
 - Formulas of
 - Radial Basis Function;
 - Polynomial Kernel function;
 - Radial Basis Kernel function.
 - Comp. complexity of matrix inversion.
-
- You should assume no other information will be given (at least for this part of SPS).

Facts of Exam

- Multiple choices:
 - **Concepts**: e.g. which of ... is true/false
 - **Calculation**: e.g. given info, calculate sth.
 - **Practical**: e.g. given a problem setting, which one of the following XXX should be used...
- Part III is new this year!
 - No previous exam available!
- Test yourself using all the mock questions (marked as "**M**" in this presentation).
- Live demonstrations happened off the slides will **not** be tested

Overview

- Feature Transform

- Different Types of Feature Transforms
- Variance and Bias Decomposition
- Kernel Methods

- Feature Redundancy Removal

- PCA and FDA

- Feature Dependency Modelling

- Markov Net
- Bayesian Net

Focus



Prerequisites

- What is Least squares?
 - How to solve it?
- What is Training data/Testing data?
 - What is training error/testing error?
- What is overfitting?

Feature Transforms

Lecture 1.

Key Messages

- Polynomial Transform
 - What is "Polynomial feature transform, with degree $b=X$ "?
 - How choices of b affect classification boundary?
- RBF Transform
 - What is "RBF feature transform, with number of basis, $b=X$ "?
 - How do you select centroids?
 - What does the hyper para. σ do?

Polynomial Transform

- Let $f(x)$ be polynomial functions:
- When $x \in R$, $f(x) := [x^0, x^1, x^2, \dots, x^b]$.
 - b is called the degree of f .
 - $f(x) = [0, x, x^2]$ is called a degree 2 polynomial trans. on x .

Polynomial Transform

- When $\mathbf{x} \in R^d$,
 - $\mathbf{f}(\mathbf{x}) := [\mathbf{h}(x^{(1)}), \mathbf{h}(x^{(2)}), \dots, \mathbf{h}(x^{(d)})]$.
 - $\mathbf{h}(t) := [t^0, t^1, t^2, \dots, t^b] \in R^{b+1}$.
 - $\mathbf{f}(\mathbf{x}) \in R^{d(b+1)}$, which means $\boldsymbol{\beta} \in R^{d(b+1)}$.

Polynomial Transform on Data Matrix

- $X \in R^{n \times d}$ is data matrix with n observations and d dimensions.

- $f(X) := \begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_n) \end{bmatrix} \in R^{n \times d(b+1)}.$

- We expanded our data matrix.
 - from d to $d(b + 1)$

LS Solution

- $\hat{\beta} := \arg \min \sum_{i=1}^n (y_i - \langle \beta, f(x_i) \rangle)^2$
- $\hat{\beta} := (f(X)^\top f(X))^{-1} f(X)^\top y$
- **M**: what is the computational complexity of calculating $\hat{\beta}$?

Radial Basis Function (RBF)

- RBF is another widely used basis function for function approximation.
- $f^{(i)}(x) := \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right)$
 - $\sigma > 0$ is called **width** and **is a hyperparameter**.
 - σ is determined **before** fitting
 - A practice is setting σ as the median of all pairwise distances of x in your data.

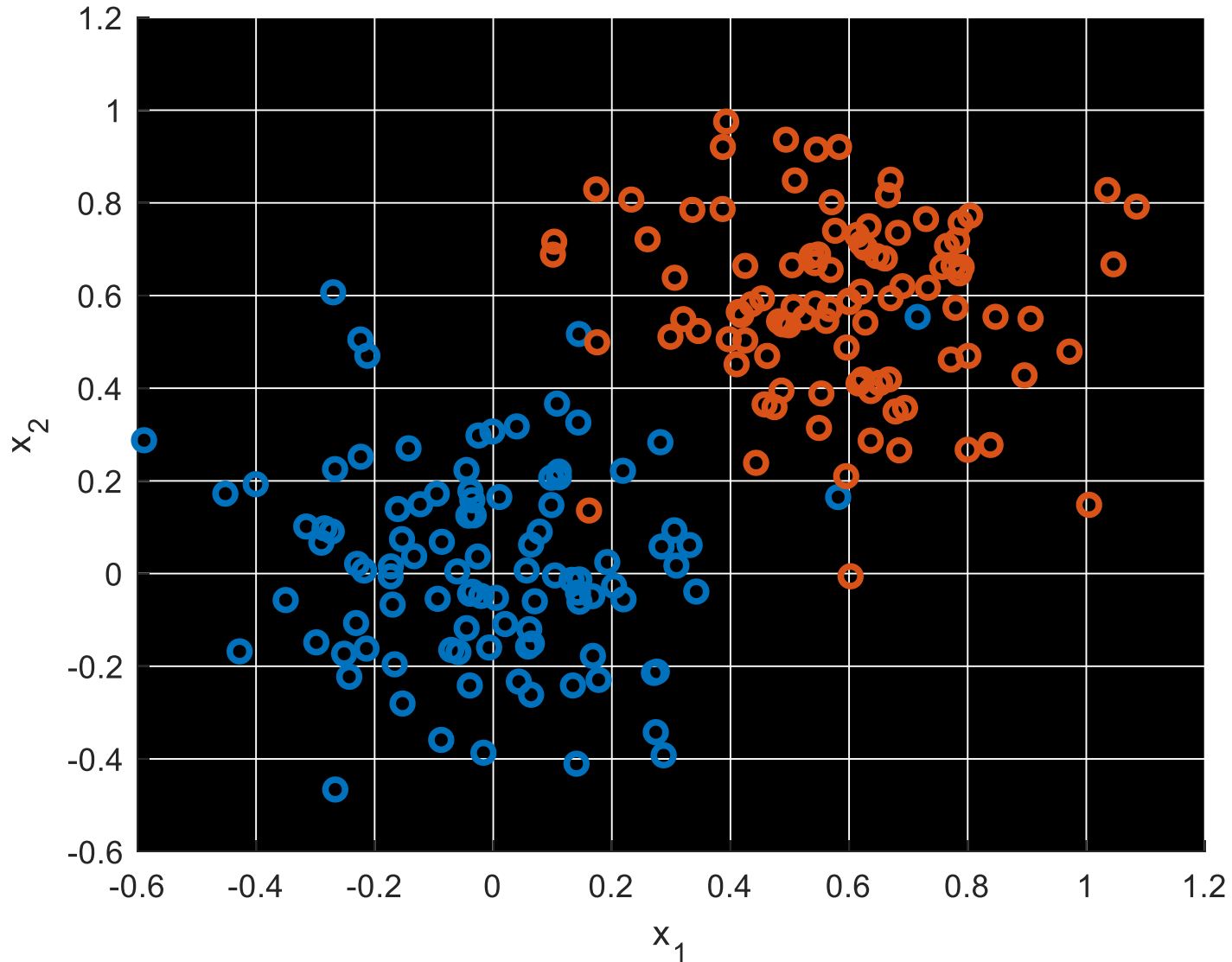
Radial Basis Function (RBF)

- x_i are called **RBF centroids**.
- x_i can be **randomly chosen** from the x in your dataset
- $f(x) := [\textcolor{red}{1}, f^{(1)}(x), f^{(2)}(x), \dots, f^{(b)}(x)]$
 - Do not forget 1!

M: How to choose f given data

- Given a dataset D (see next slide), what f should you use for classification? **Hint:** consider computational cost and overfitting
 - Polynomial, $b = 1$
 - Polynomial, $b = 2$
 - Polynomial, $b = 3$
 - RBF, $b = 100$
- Use an f that is **just enough** for doing your job without causing heavy computation/overfitting!

M: How to choose f given data



Feature Transforms

Lecture 2. Bias and variance decomposition

Key Messages

- How the choices of b in feature transform affects training and testing error?
 - Training error \rightarrow goes down as b increases.
 - Testing error \rightarrow goes down and then raise up as b increases.
- What is the expected error at a data point x_i in regression problem?
 - How does it decompose?
 - Remember the decomposition formulas.

Expected Square Error Decomposition

- Given dataset, $D = \{(x_i, y_i)\}$,
- $y_i = g(x_i) + \epsilon$, $\epsilon \sim N(0, \sigma^2)$

- Bias and Variance Decomposition:

$$\begin{aligned} & \mathbb{E}_{\epsilon}[(y - \hat{y}_i)^2 | \mathbf{x}_i] \\ &= \underbrace{\text{var}[\epsilon]}_{\text{Irreducible error}} + \underbrace{[g(x) - \mathbb{E}_{\epsilon}[\hat{y}_i | \mathbf{x}_i]]^2}_{\text{bias}} + \underbrace{\text{var}[\hat{y}_i | \mathbf{x}_i]}_{\text{variance}} \end{aligned}$$

- “Variance and Bias decomposition”

M: Calculate Variance.

- Given a data generation scheme, $y_i = x_i + \epsilon$, $\epsilon \sim N(0, \sigma^2)$, $\sum_{i=1} x_i^2 = C$ and a regression model $\hat{y} = \hat{\beta} \cdot x$, where $\hat{\beta}$ is calculated using least squares.
- 1. Write down bias and irreducible error.
 - irr. error = σ^2 , bias = 0
- 2. Calculate variance term at a data point $x = 1$. (see next slide for a cheat)

A Closer Look at In Sample $\text{var}[\hat{y}]$

- $\text{var}[\hat{y}|\mathbf{x}_i] = \langle h(\mathbf{x}_i), h(\mathbf{x}_i) \rangle \cdot \sigma^2$
 - Where $h(\mathbf{x}_i) :=$
$$\mathbf{f}(\mathbf{x}_i)(\mathbf{f}(\mathbf{X})^\top \mathbf{f}(\mathbf{X}))^{-1} \mathbf{f}(\mathbf{X})^\top$$
- Figure out what is $\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{X})$ and $g(\mathbf{x})$ in this example, then you can use this formula to calculate the result.
- $\text{var}[\hat{y}|\mathbf{x}_i] = \frac{\sigma^2}{c}$

Feature Transforms

Lecture 3. Kernel methods

Key Messages

- How do we perform kernel least squares?
- Prediction rule: $\hat{\mathbf{y}} := \mathbf{k}(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}$
 - What are $\mathbf{k}, \mathbf{K}, \mathbf{I}, \mathbf{y}, \lambda$?
 - How do you use this rule to make a prediction?
 - **Remember this prediction rule.**
- What is
 - Linear kernel function
 - Polynomial kernel function
 - RBF kernel function?

M: Example

- Given a dataset $\{(y_1 = 1, x_1 = 1), (y_2 = -1, x_2 = -1)\}$, calculate \mathbf{K} in the kernel least square prediction rule using
 - Linear kernel
 - $\mathbf{K} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
 - Polynomial kernel $k(x, x') := (\langle x, x' \rangle + 1)^2$.
 - $\mathbf{K} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
- Calculate \mathbf{k} for a prediction \hat{y} at data point $x = 2$ using
 - Linear kernel: $\mathbf{k} = [2, -2]$
 - Polynomial kernel: $\mathbf{k} = [9, 1]$

Feature Redundancy

Lecture 4. PCA

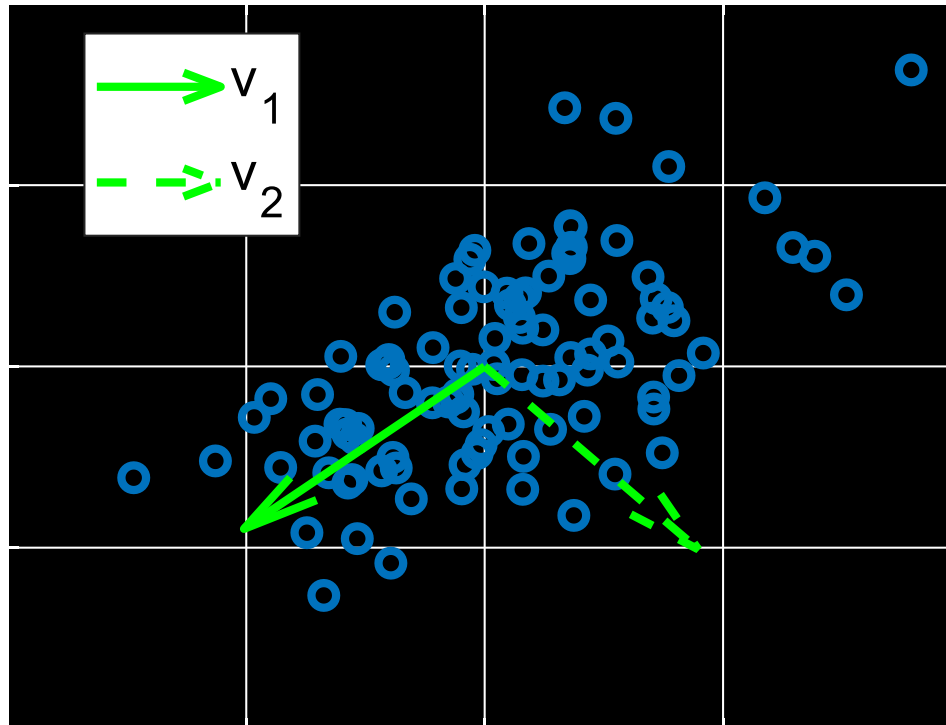
Key Messages

- What is curse of dimensionality?
 - The performance of machine learning algorithm degrades when the dimensionality of dataset increases.
- What kind of information is most likely preserved in a PCA projection?

Minimizing Projection Error

- $\min_{B, \mathbf{B}\mathbf{B}^\top = I} \sum_{i=1}^n \left\| \mathbf{x}_i^\top - \mathbf{B}^\top \mathbf{B} \mathbf{x}_i^\top \right\|^2$
 - We minimize square error between original data points and its projection.

Example



v_1 always points at the direction where your dataset has the largest variance!

Feature Redundancy

Lecture 5. FDA

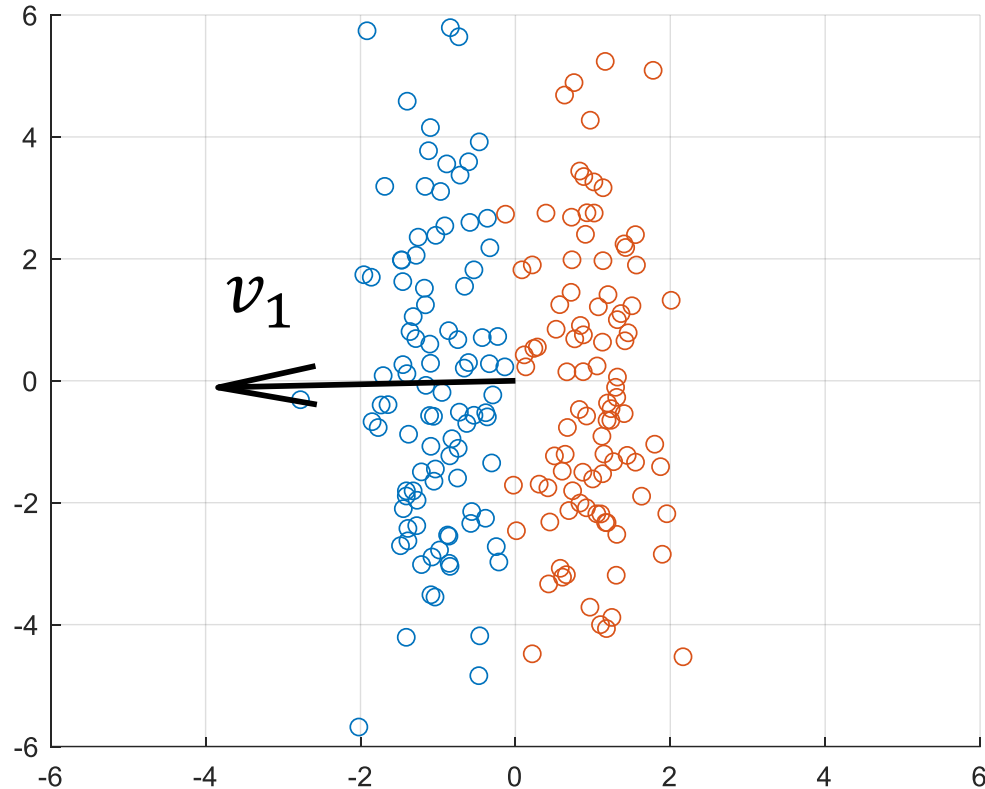
Key Messages

- Why PCA does **NOT** preserve cluster/class information?
 - It does not take class information into account
- What is within class scatterness?
- What is between class scatterness?
- What **kind of information** is most likely preserved in a FDA projection?

Objective of FDA

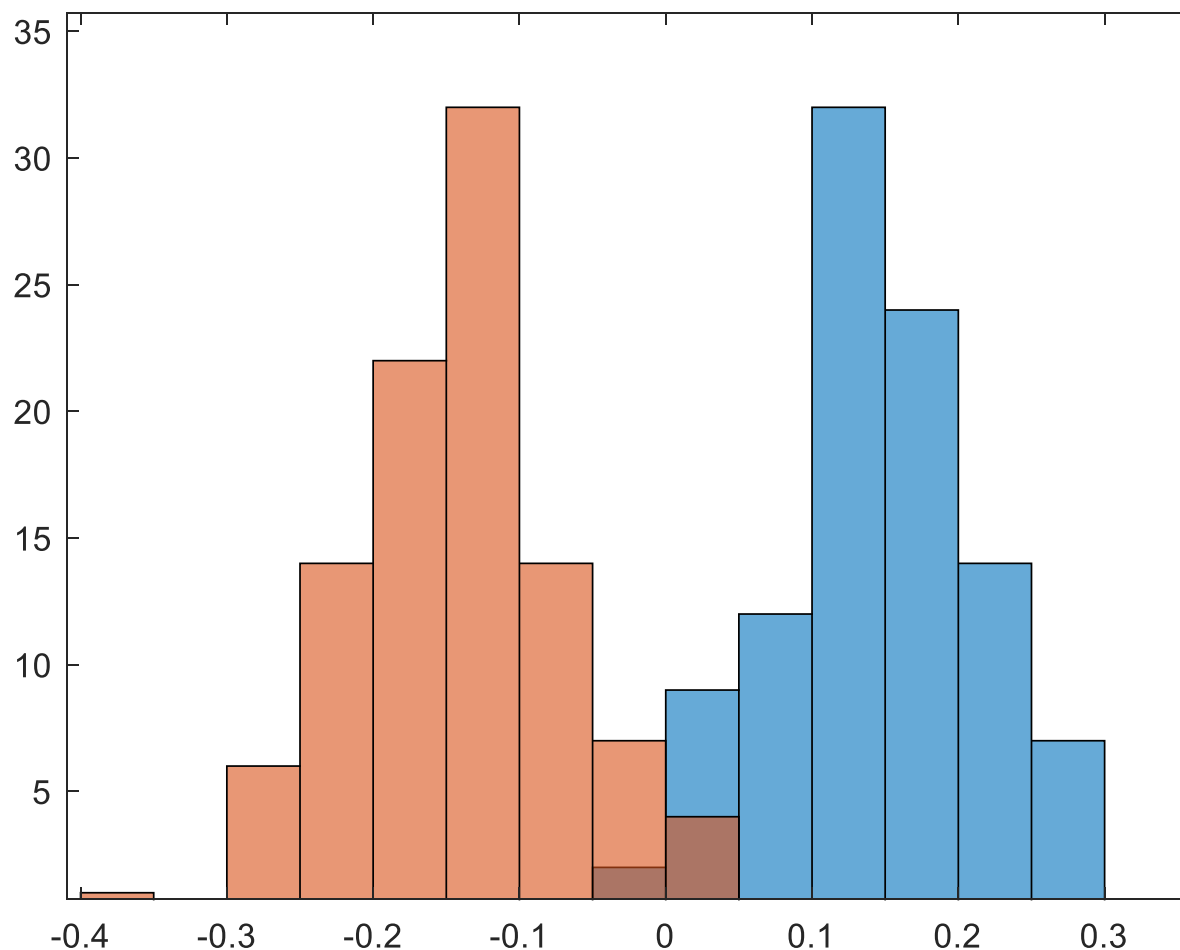
- **Maximizing** between class scatterness \forall_k .
 - **Minimize** within class scatterness \forall_k .

Example: Binary Classification Dataset



FDA embeds samples to a subspace that is the most **linearly** separable.

Example: embedding, $v_1^T x^T$



Class separation is preserved
after embedding.

Feature Dependency

Lecture 6. Markov Net

Key Messages

- Cond. independence in a distribution can be encoded by a graph.
- The density of such a distribution factorizes over the same graph.
- What is **Gaussian Markov net**?

Gaussian Markov Network

- Multivariate Gaussian distribution:

- $\mathbf{x} \in R^d, \mathbf{x} \sim N(\mathbf{0}, \Sigma)$

- $p(\mathbf{x}) \propto \exp \left[-\frac{\mathbf{x}(\Sigma)^{-1} \mathbf{x}^T}{2} \right]$ Let $\Theta = (\Sigma)^{-1}$.

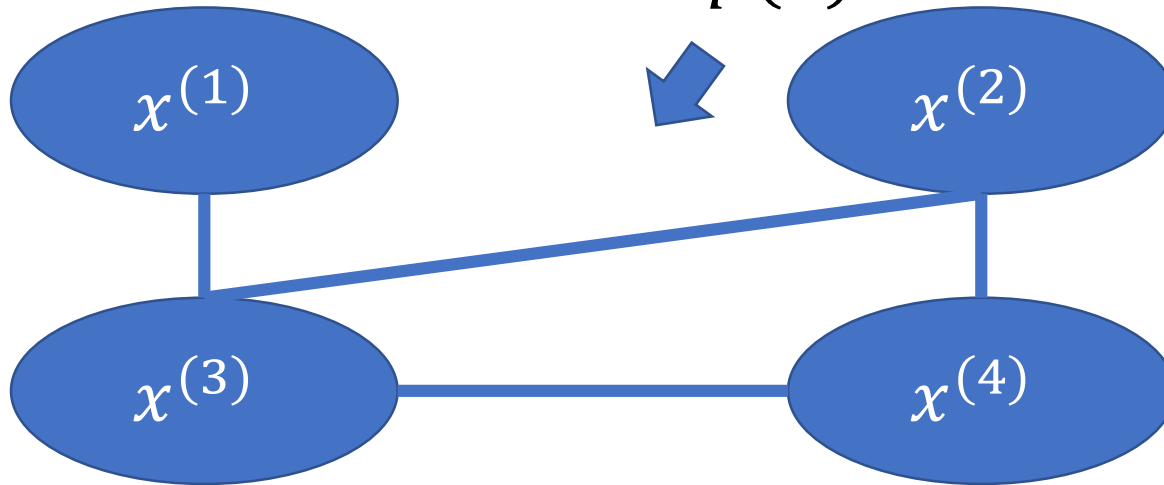
$$\propto \exp \left[-\frac{\sum_{u,v} \Theta^{(u,v)} x^{(u)} x^{(v)}}{2} \right]$$
$$\propto \prod_{u,v; \Theta^{(u,v)} \neq 0} \exp(-\Theta^{(u,v)} x^{(u)} x^{(v)})$$

Gaussian Markov Network

- $p(\mathbf{x}) \propto \prod_{u,v; \Theta(u,v) \neq 0} g_{u,v}(x^{(u)}, x^{(v)})$
- $p(\mathbf{x})$ **factorizes over G !**
 - G defined by the adjacency matrix A
$$A^{(u,v)} = \begin{cases} 0, & \Theta(u,v) == 0 \\ 1, & \Theta(u,v) \neq 0 \end{cases}$$
 - G must be an undirected graph (why?)
- $\Leftrightarrow p(\mathbf{x})$ satisfies the conditional independence encoded in G .

Example

This G encodes cond.
independence in a Gaussian
MN $p(\mathbf{x})$



$$\bullet \Theta = \begin{bmatrix} \Theta_{11} & 0 & \Theta_{13} & 0 \\ 0 & \Theta_{22} & \Theta_{23} & \Theta_{24} \\ \Theta_{13} & \Theta_{23} & \Theta_{33} & \Theta_{34} \\ 0 & \Theta_{24} & \Theta_{34} & \Theta_{44} \end{bmatrix}$$

Notice how
the sparsity of
 G translates
into the
sparsity of Θ !

Diagonal must
be filled!

M

- Suppose graph G encodes all cond. indep. in your Gaussian MN p . G contains **three edges, five nodes**. How many **non-zero elements** are there in **inverse covariance** matrix of p ?
 - A.3
 - B.8
 - C.6
 - D.10
 - **E.11**
- #Edges *2 + #Vertices
Understand why #Edges times 2
Understand why vertices must be non-zero

Feature Dependency

Lecture 7. Bayesian Net

Important Concepts

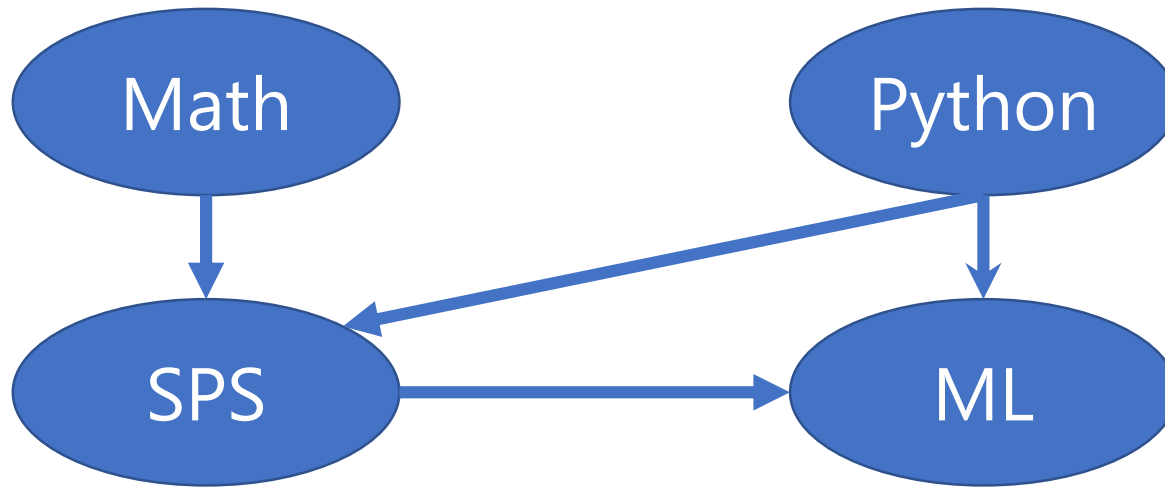
- What is a DAG?
- How a density is represented by a DAG? (Chain rule)
- How do you read conditional independence from a DAG?
- How Naïve Bayes Classifier is derived from a Bayesian net?

Representing Factorization using DAG

- DAG can also be used to represent the factorization of a probability dist.
- We say a probability dist. $p(X)$ factorizes over a DAG G if
- $p(X) = \prod_{v \in V} p(X_v | X_{\text{parent}(X_v)})$

M: Expressing Density using DAG

- Write down the Bayesian net represented by this graph:



$$\begin{aligned} & p(Ma, Py, SPS, ML) \\ &= p(Ma)p(Py)p(SPS|Ma, Py)p(ML|SPS, Py) \end{aligned}$$

Represent Cond. Indep. using DAG

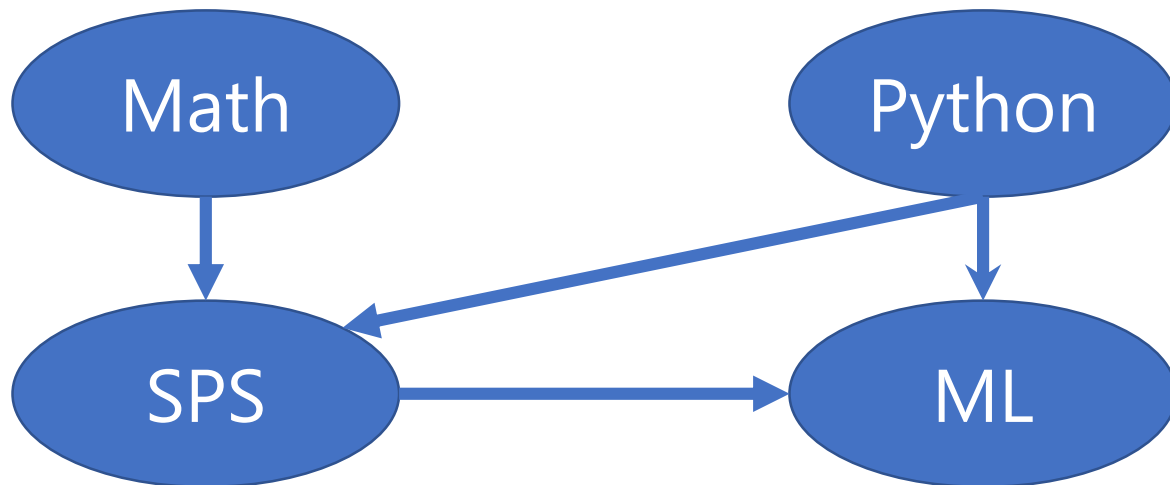
- Given DAG G .

- X_v is independent of $X_{\text{non-desc}(X_v)}$ given $X_{\text{parent}(X_v)}$, $\forall v$.

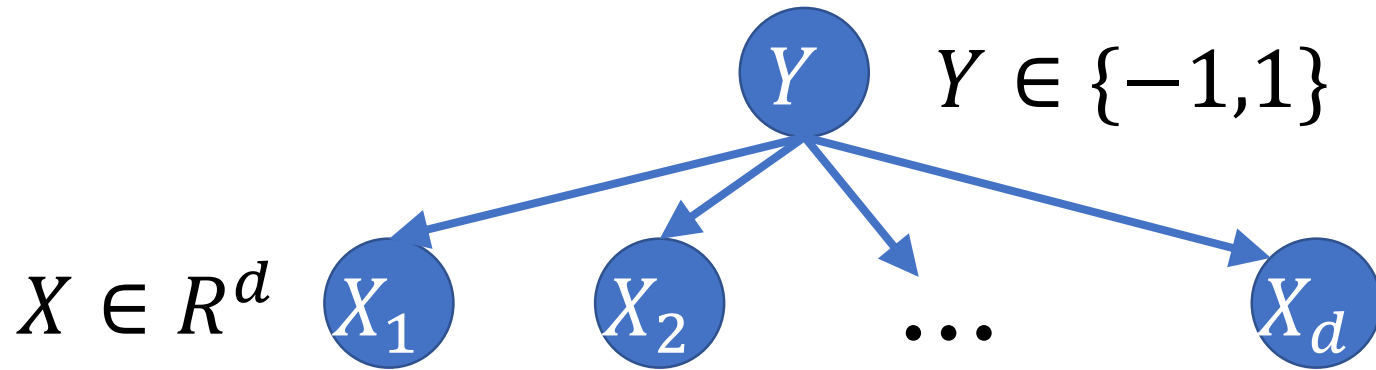
- This is an analogy to Markov net, as X_v and all non-descendants of X_v are “blocked” by the parents of X_v .
- Knowing $X_{\text{parent}(X_v)}$, $X_{\text{non-desc}(X_v)}$ tell us nothing new about X_v .

M: Expressing Cond. Indep. Using DAG

- Which of the following Cond. Indep. is **not** encoded by the graph?
 - $ML \perp Math \mid SPS, Python$
 - $Math \perp Python$
 - $SPS \perp ML \mid Math$



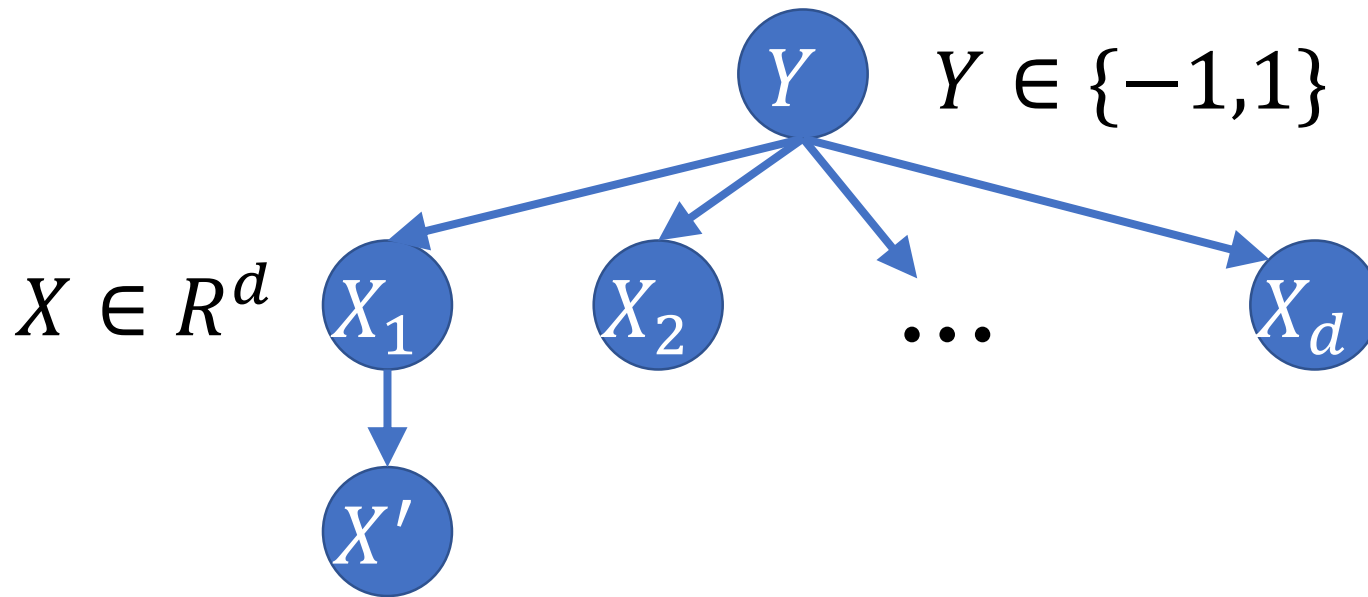
Bayesian Network for Classification



Bayesian Network for Classification

- Write down the conditional probability $P(Y|X)$.
- $$P(Y|X) = \frac{\prod_i P(X_i|Y)P(Y)}{P(X)}$$
- This is how Naïve Bayes is derived!

M: “useless feature”



- Given this Bayesian Net for a classification task, should you include feature X' for training? Why?
- $P(Y|X) = \frac{\prod_i P(X_i|Y)P(Y)}{P(X)} p(X'|X)$
- $\hat{y} := \operatorname{argmax}_y p \left(\frac{\prod_i P(X_i|Y)P(Y)}{P(X)} \overset{\text{Does not depends on } y}{p(X'|X)} \right)$, for a specific x !
- You should not include X' for training.

In Conclusion...

- Take your time to do all questions.
- Bring a Calculator!
- Office Hour:
 - next week Tuesday 3-5pm;
 - next week Thursday 3-5pm.