

Capturing Dependency of Data using Graphical Models

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

Objectives

- Understand **equivalence of conditional independence of R.Vs and factorizations** of their probability distribution over a graph.
- Simple **undirected graphical models**:
 - Gaussian Markov Network
 - Logistic Model

Example: Scores of Units

- Imagine a table of unit scores.

| Name | SPS | Math | Python | Mach. Learn. |
|----------|-----|------|--------|-----------------|
| Song | 80 | 70 | 50 | 60 |
| Harry | 50 | 40 | 70 | 80 |
| Ron | 50 | 50 | ... | 45 |
| Hermione | 90 | 100 | ... | 100 |
| ... | ... | ... | ... | ... |

Dependency of Datasets and Its Graphical Representation

- Scores of units are **dependent!**
 - Student with **high** Math, Python score is likely to receive **high** SPS score.
 - Student with **high** SPS score is likely to receive a **high** Mach. Learn. score.

Problem Formulation

- Given a dataset $\{\mathbf{x}_i\}_{i=1}^n$,
 - $\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)} \dots x_i^{(d)}] \in R^d$
 - \mathbf{x}_i is a vector of a student i 's scores.
 - e.g., $x^{(1)}$ is SPS, $x^{(2)}$ is Math...
- **What does $p(x^{(1)}, x^{(2)} \dots x^{(d)})$ look like?**

Independence of R.V.s

- Let's look at how independence between R.V.s are **expressed in probability**:
- R.V. X is **independent** of Y :
 - $X \perp Y$
 - $\Leftrightarrow p(X, Y) = p(X)p(Y)$
 - Factorization
 - $\Leftrightarrow p(X|Y) = p(X) \Leftrightarrow p(Y|X) = p(Y)$
 - No Information flows between X and Y .

Example: Likelihood with Independent Datapoints:

- Likelihood over the dataset
 - Factorizes into product over each x_i
 - $p(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n p(x_i; \theta)$
 - We can do so as $x_1 \dots x_n$ are independent.
- Maximum Likelihood Estimation
 - $\max_{\theta} \prod_{i=1}^n p(x_i; \theta)$
 - **Lab sheet 4.1**

Conditional Independence of R.V.s

- R.V. X is independent of Y **given** Z
 - $X \perp Y|Z$
 - $\Leftrightarrow p(X, Y|Z) = p(X|Z)p(Y|Z)$
 - $\Leftrightarrow p(X, Y, Z) \propto g_1(X, Z) \cdot g_2(Y, Z)$
 - Factorization
 - $\Leftrightarrow p(X|Y, Z) = p(X|Z)$
 - Information flow: Y does not give any additional info which changes the prob. of X given Z .
 - $\Leftrightarrow p(Y|X, Z) = p(Y|Z)$

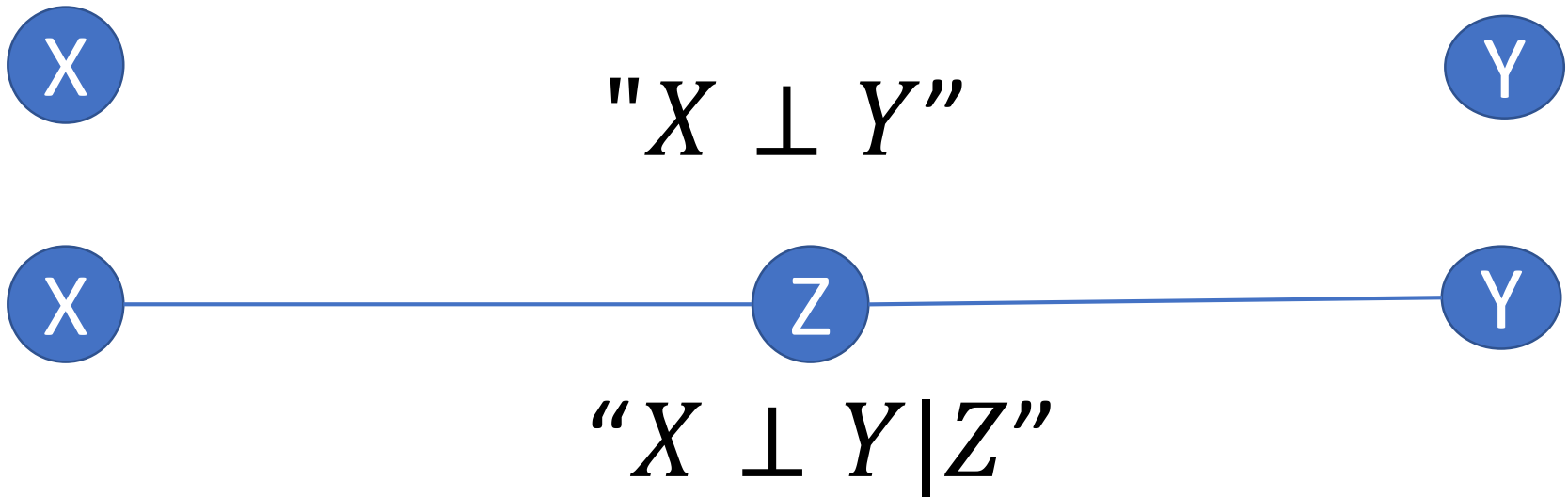
(Conditional) Independence and Information Flow

- (Conditional) Independence tells how information **flows** between R.V.s
 - $X \perp Y \Leftrightarrow$ no information flows in-between X and Y .
 - $X \perp Y | Z \Leftrightarrow$ information **flows between** X and Y **via** Z .



Representing (Conditional) Independence by Graph

- Given many R.Vs, listing all (cond.) independence can be cumbersome.
- A **graphical representation** is helpful:



Representing Conditional Independence by Graph

- Given a graph $G = \langle E, V \rangle$, and three random variables $X, Y, Z \subseteq V$
 - if X and Y are completely “**blocked**” by Z , we say $X \perp Y | Z$ is represented by G .

Example: Encoding (cond.) indep. by graph

$\text{Math} \perp \text{ML} \mid \text{SPS}$

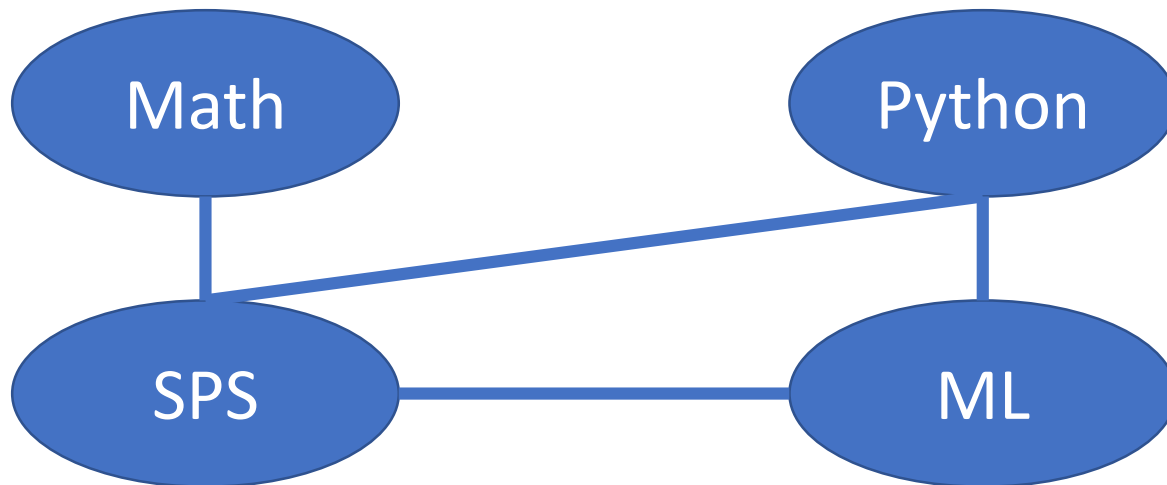
$\text{Math} \perp \text{Python} \mid \text{SPS}$

$\text{Math} \perp \text{ML} \mid \text{SPS}, \text{Python}$

$\text{Math} \perp \text{Python}, \text{ML} \mid \text{SPS}$

$\text{Math} \perp \text{Python} \mid \text{SPS}, \text{ML}$

List of
conditional
independen
ce encoded
by Graph!





Representing Prob. Distribution Factorization by Graph

- Factorizing a probability dist. greatly reduces complexity of modelling and computation of a probability dist.
 - Think about that Maximum Likelihood example you did in Lab!

Representing Prob. Distribution Factorization by Graph

- Writing the factorization of a probability distribution of many factors can be cumbersome.
- Can we also use graph to help??

 $P(X, Y) = P(X)P(Y)$ 



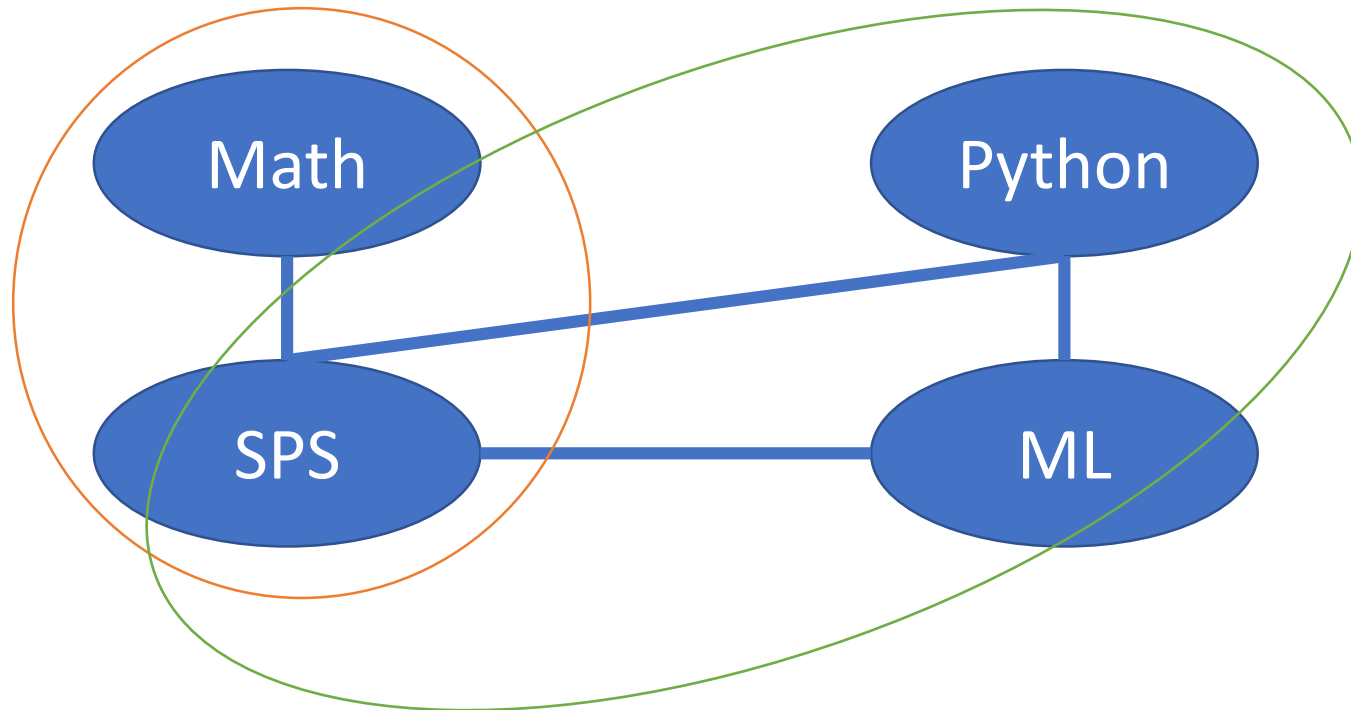
$P(X, Y, Z) \propto g_1(X, Z)g_2(Y, Z)$

Representing Prob. Distribution Factorization by Graph

- Given a graph $G = \langle E, V \rangle$,
- We say $p(X)$ factorizes over G :
- If $p(X) \propto \prod_{c \in \mathcal{C}} g_c(X^{(c)})$
 - where \mathcal{C} is set of all **cliques** in G .
 - Clique: fully connected subgraph.
 - g_c is a function defined on $X^{(c)}$, which is the subset of X **restricted on** c .

Example

$$p(Ma, SPS, Py, ML) \\ \propto g_1(Ma, SPS) \cdot g_2(Py, ML, SPS).$$



Equivalency between Factorization and Conditional Independence over G

- Using graph represent a factorization of a probability distribution
- Using graph represent a list of conditional independence
- Remarkably, these two seemingly irrelevant notions are **equivalent!**

Equivalency between Factorization and Conditional Independence over G

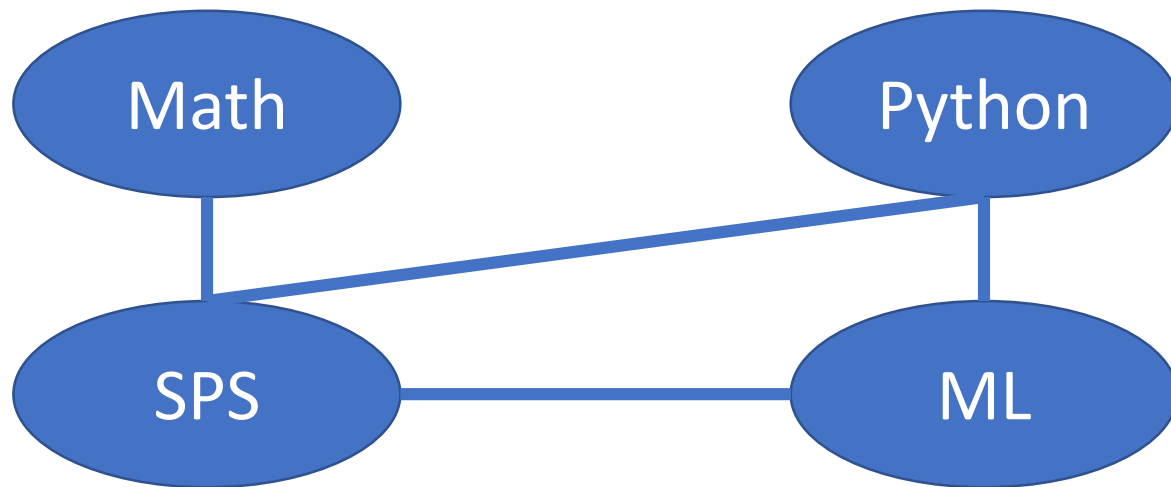
- If p factorizes over G , p satisfies all conditional independence represented by G .
- If p satisfies all conditional independence represented by G , then p factorizes over G .

Equivalency between Factorization and Conditional Independence over G

- Verify this on Scores of Units
example!
- Live demonstration.

Example

$$p(Ma, SPS, Py, ML) \\ \propto g_1(Ma, SPS) \cdot g_2(Py, ML, SPS).$$



$$\text{Hint: } X \perp Y|Z \Leftrightarrow p(X, Y, Z) \propto g_1(X, Z) \cdot g_2(Y, Z) \\ X \perp Y, W|Z \Rightarrow X \perp Y|Z$$

Example

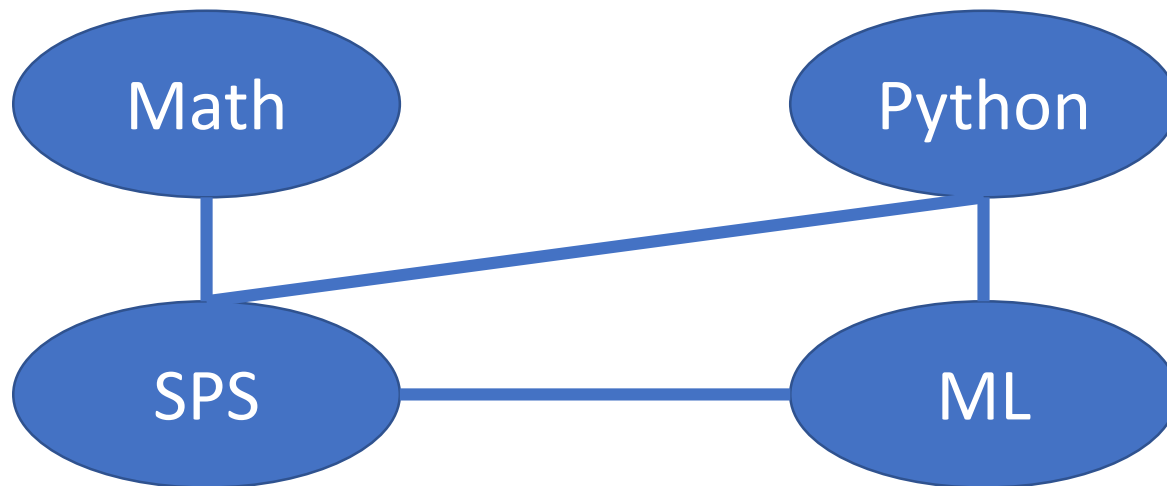
Math \perp ML | SPS

Math \perp Python | SPS

Math \perp ML | SPS, Python

Math \perp Python, ML | SPS

Math \perp Python | SPS, ML



Hint: $X \perp Y | Z \Leftrightarrow p(X, Y, Z) \propto g_1(X, Z) \cdot g_2(Y, Z)$

Markov Network

- A probability distribution $p(X)$ which uses undirected graph representing its conditional independence, is called an **undirected graphical model**, or a **Markov network**.

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 $\Theta = (\Sigma)^{-1}$.

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

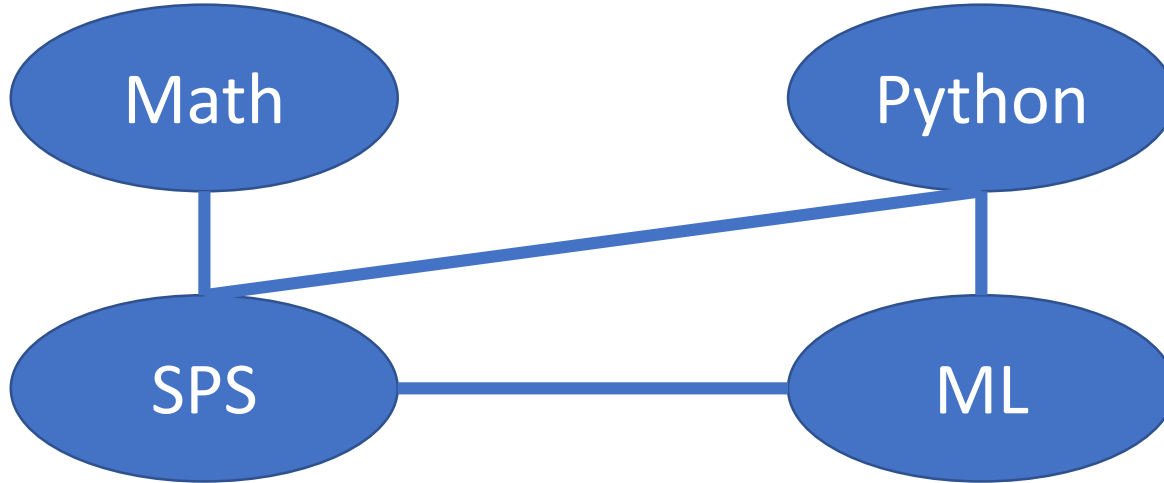
Gaussian Markov Network

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

Gaussian Markov Network

- Knowing a graph G that encodes all conditional independence of your dataset, I can use its adjacency matrix G to construct Θ !
 - Use sparsity of the adjacency matrix
 - NOT its actual values!
 - Θ must be positive definite!!

Example



• $x^{(1)}:\text{Math}; x^{(2)}:\text{Py}; x^{(3)}:\text{SPS}; x^{(4)}:\text{ML}$

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

Quiz

- Suppose graph G encodes all cond. indep. in your probability dist. G contains **three edges, five nodes**. How many **non-zero elements** are there in Θ , the parameter to your Gaussian Markov Net?
- A.3
- B.8
- C.6
- D.10
- E.11

<https://bit.ly/2uIFZUu>

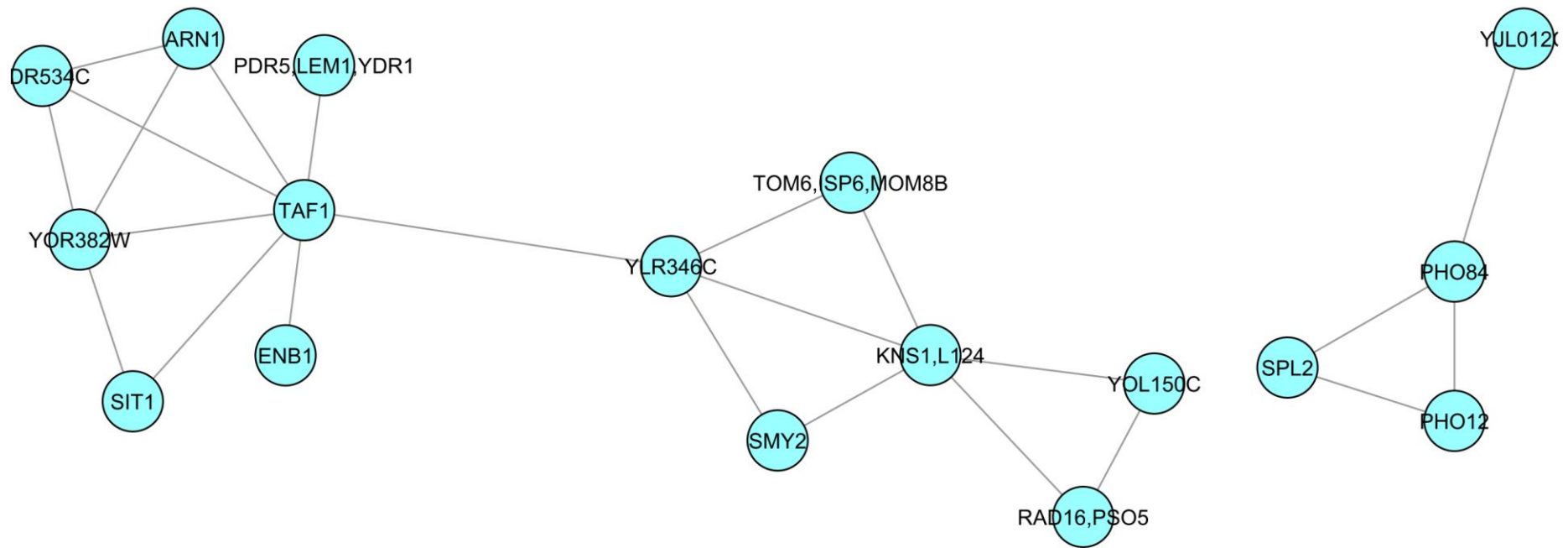
Gaussian Markov Network

- **Not knowing** G encodes all cond. independence of $p(\mathbf{x})$. Given dataset D , we can fit a sparse $\hat{\Theta}$.
 - Using MLE: $\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log p(D; \Theta)$
 - The sparsity of $\hat{\Theta}$ gives a graphical representation of $p(\mathbf{x})$!
 - Such representation reveals how random variables “interacts” with each other!

Example: Gene Expression Data

| Time stamp | Gene1 | Gene2 | Gene3 | Gene4 |
|------------|-------|-------|-------|-------|
| t1 | .1 | .2 | .5 | .2 |
| t2 | .5 | .4 | .7 | .8 |
| t3 | .5 | .5 | ... | .45 |
| t4 | .9 | .2 | ... | .01 |
| ... | ... | ... | ... | ... |

Gene Network (Banerjee et al., 2008)



Exponential Family Distribution

- Gaussian Markov network belongs to a wider **family** of distributions, which are defined using a generic form:

- $$p(\mathbf{x}; \boldsymbol{\theta}) := \frac{\exp(\langle \boldsymbol{\theta}, \mathbf{f}(\mathbf{x}) \rangle)}{Z(\boldsymbol{\theta})}$$

- $\mathbf{f}(\mathbf{x})$ is a feature transform on \mathbf{x} .

- $$Z(\boldsymbol{\theta}) := \int \exp(\langle \boldsymbol{\theta}, \mathbf{f}(\mathbf{x}) \rangle) d\mathbf{x}$$

- PC: show when \mathbf{f} is 2nd degree poly. transform with pairwise terms, $p(\mathbf{x}; \boldsymbol{\theta})$ is a multivariate Gaussian distribution.

Conditional Markov Network

- In many tasks, the conditional distribution is the key interest.
 - $p(Y|X)$ measures the randomness on Y given X and help us make a prediction.
 - Both regression and classification requires a **conditional** model.
- How to factorize a conditional distribution over G ?

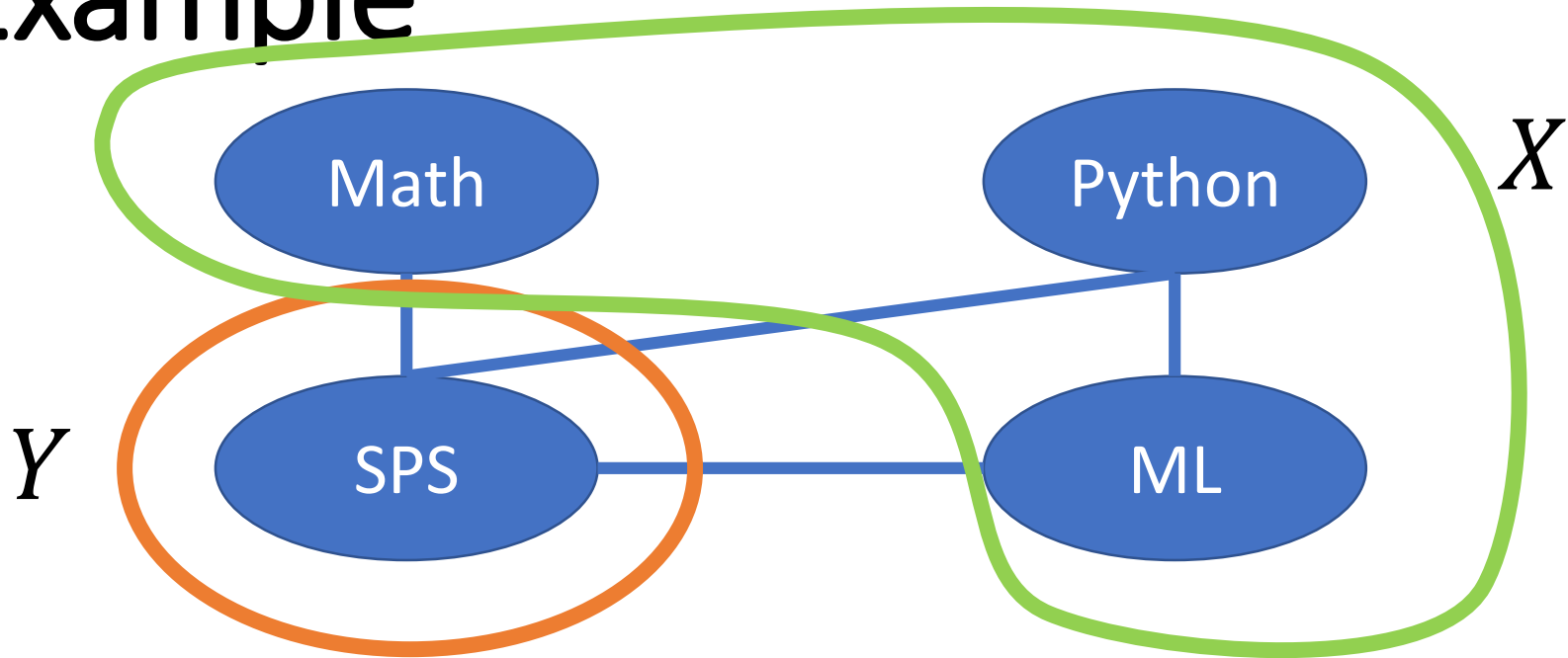
Conditional Markov Network

- We say a conditional probability distribution $P(Y|X)$ factorizes over G whose nodes $V = X \cup Y$, if
- $p(Y|X) = \frac{1}{N(X)} \prod_{c \in \mathcal{C}} g_c(Z), Z \subseteq X \cup Y$
- $N(X) := \int \prod_{c \in \mathcal{C}} g_c(Z) dY$
- Normalizing constant:
 - It normalizes the distribution to 1 over the domain of the random variable (Y).

Conditional Markov Network

- PC: show $Z \not\subseteq X$
 - $p(Y|X)$ does not include factors on conditioning variable X !

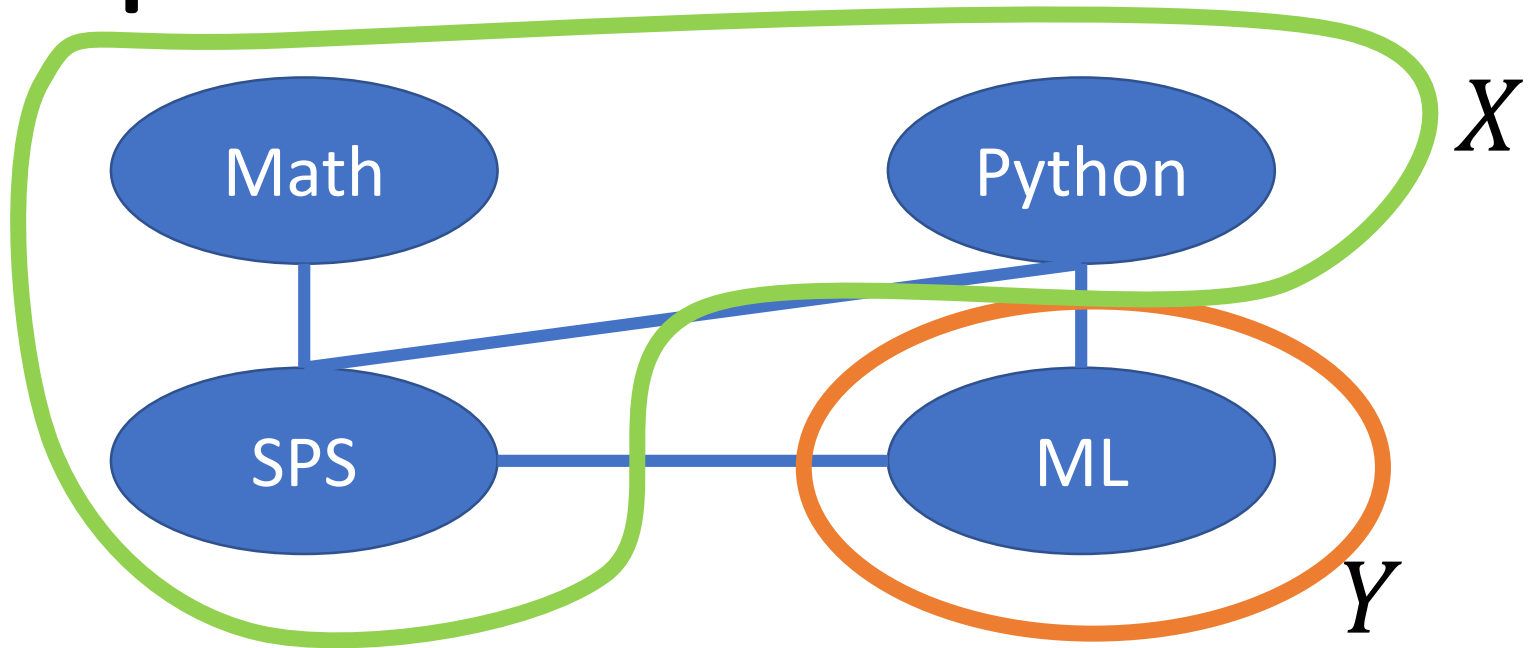
Example



$$\begin{aligned} & \bullet p(\text{SPS} | \text{Ma}, \text{Py}, \text{ML}) \\ &= \frac{1}{Z(\text{Ma}, \text{Py}, \text{ML})} g_1(\text{SPS}, \text{Py}, \text{ML}) g_2(\text{SPS}, \text{Ma}) \end{aligned}$$

$$\begin{aligned} & \bullet Z(\text{Ma}, \text{Py}, \text{ML}) = \\ & \int g_1(\text{SPS}, \text{Py}, \text{ML}) g_2(\text{SPS}, \text{Ma}) d\text{SPS} \end{aligned}$$

Example



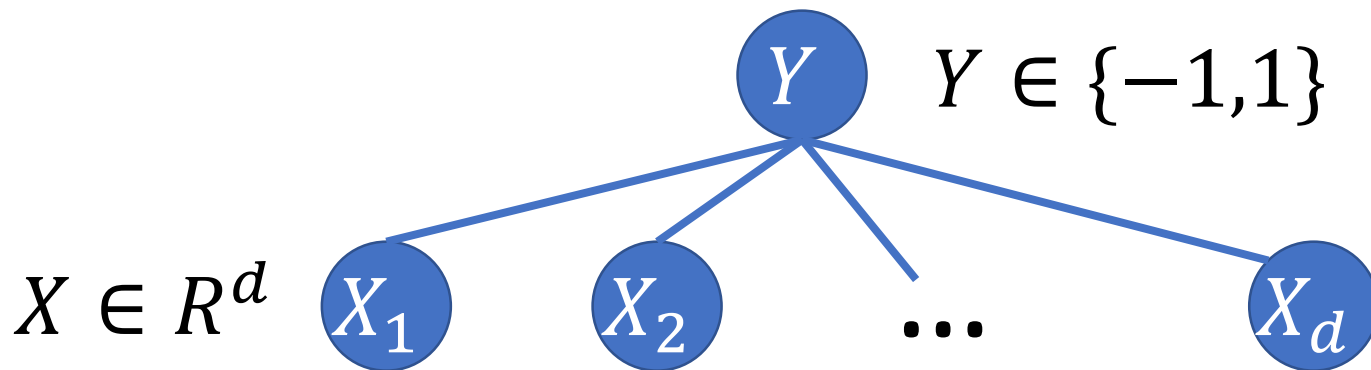
- $p_1(ML|Ma, Py, SPS)$
$$= \frac{1}{Z(Ma, Py, SPS)} g_1(SPS, Py, ML)$$

- $Z(Ma, Py, SPS) = \int g_1(SPS, Py, ML) dML$

- g_2 is gone!

Logistic Regression

- The way of constructing a conditional P.D. gives us a simple classification tool: Logistic Regression.
- Consider a simple Markov Net

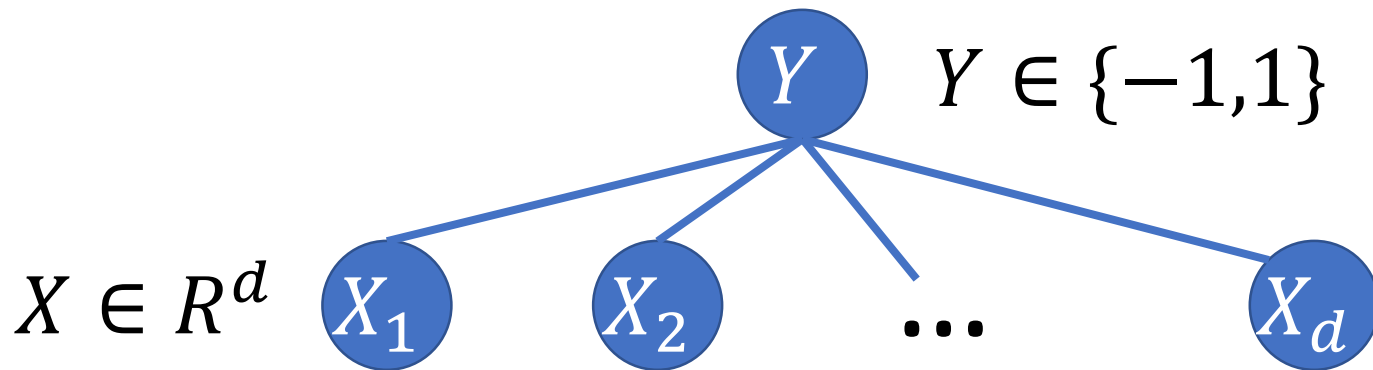


Logistic Model

- Using the factorization rule above,

- $p(Y|X) = \frac{1}{N(X)} \prod_i g_i(Y, X^{(i)})$

- $N(X) = \sum_{c \in \{-1, 1\}} \prod_i g_i(Y, X^{(i)})$



Logistic Model

- Let us construct a model of $p(Y|X)$!

- By setting

$$g_i(Y = y, X_i = x^{(i)}; \beta_i) := \exp(\beta_i \cdot y x^{(i)})$$

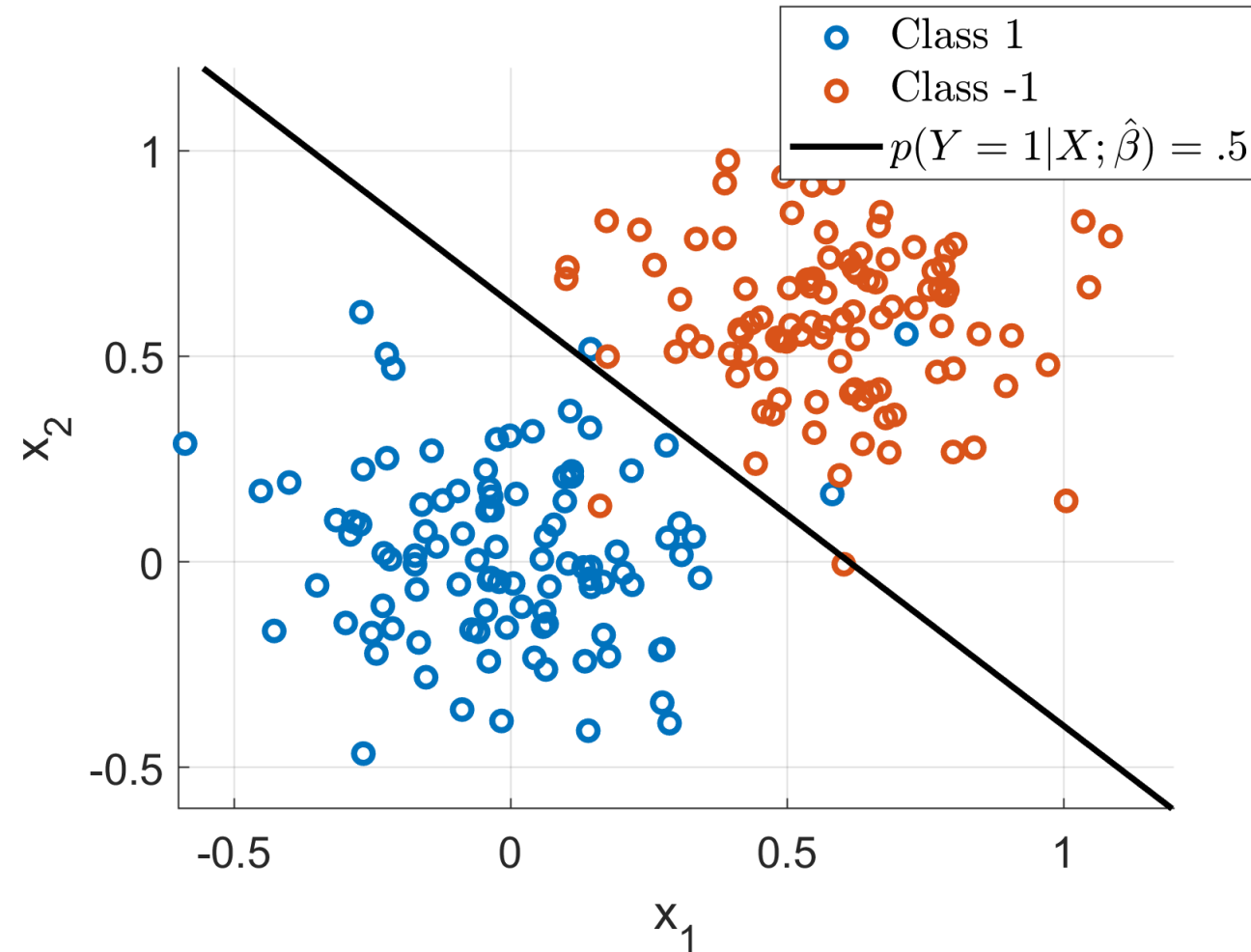
- $$p(y|\mathbf{x}; \boldsymbol{\beta}) = \frac{1}{N(X)} \exp\left(\sum_i \beta^{(i)} \cdot y x^{(i)}\right)$$
$$= \frac{1}{N(X)} \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle y).$$

- $$N(X; \boldsymbol{\beta}) = \sum_{y \in \{1, -1\}} \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle y)$$

Logistic Regression

- Logistic model:
- $p(y|x; \boldsymbol{\beta}) = \frac{1}{N(x)} \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle y)$
- $N(x) = \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle) + \exp(-\langle \boldsymbol{\beta}, \mathbf{x} \rangle)$
- $\boldsymbol{\beta}$ can be fitted using MLE.
 - $\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} \sum_{i=1}^n \log p(y_i | \mathbf{x}_i; \boldsymbol{\beta})$
 - The process of fitting $\boldsymbol{\beta}$ using MLE is called Logistic Regression.
 - `sklearn.linear_model.LogisticRegression`

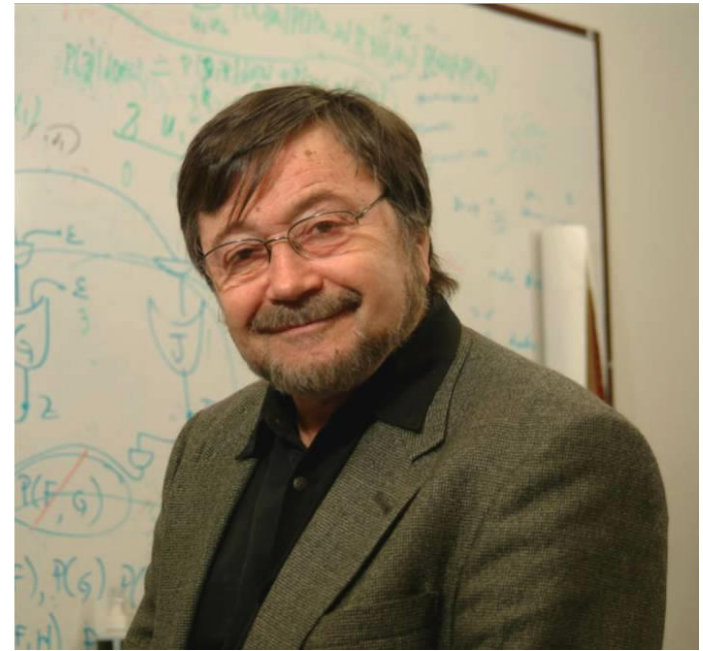
Example



- Unlike least squares classifier, logistic classifier is a probabilistic classifier, which outputs $p(Y|X; \hat{\beta})$, which is more interpretable!

Conclusion

- Markov network uses an **undirected graph** to represent conditional independencies and factorizations of a probability distribution.
- Two examples of Markov network
 - Gaussian Markov network factorizes over the graph defined by its **inverse covariance**.
 - Logistic model is a conditional P.D. model factorizes over a classification network.



Judea Pearl

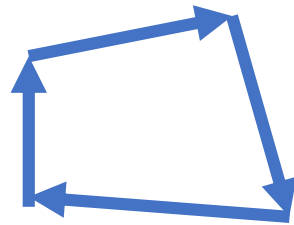
Bayesian Network

A Directed Graphical Model

- Markov network is a **undirected graphical model**.
 - which encodes **cond. indep.**
 - and **factorization** of a probability dist.
- Can we use a **directed graphical model** to do the same job?
 - Some dependency are better addressed using a directed model.

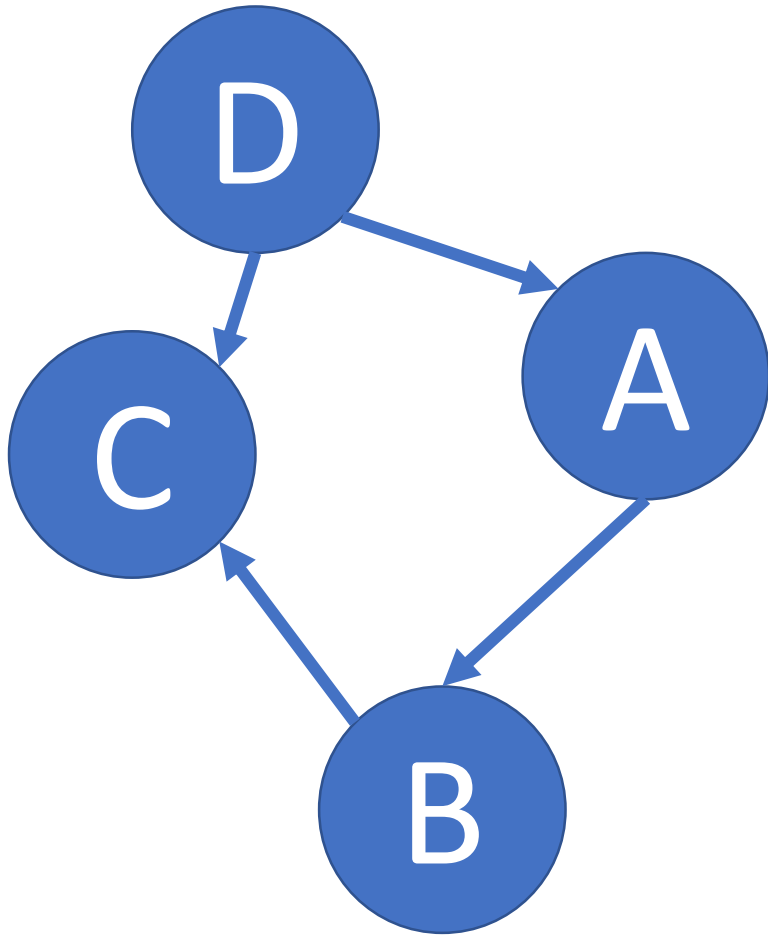
Directed Acyclic Graph

- The directed graphical model uses Directed Acyclic Graph (DAG) as its graphical representation.
 - $G := \langle E, V \rangle$, E is directed edge set.
 - DAG: G **without directed cycles.**



A directed cycle

Parents, Children, Descendants



One node may have
more than one parent
or child!

If there exists an
directed edge $A \rightarrow B$:
 A is the parent of B and
 B is the child of A .

If there exists an
directed path $A \rightarrow B$: B
is the descendant of A .

Parent(A): D

Children(A): B

Descendants(A): B,C

Example



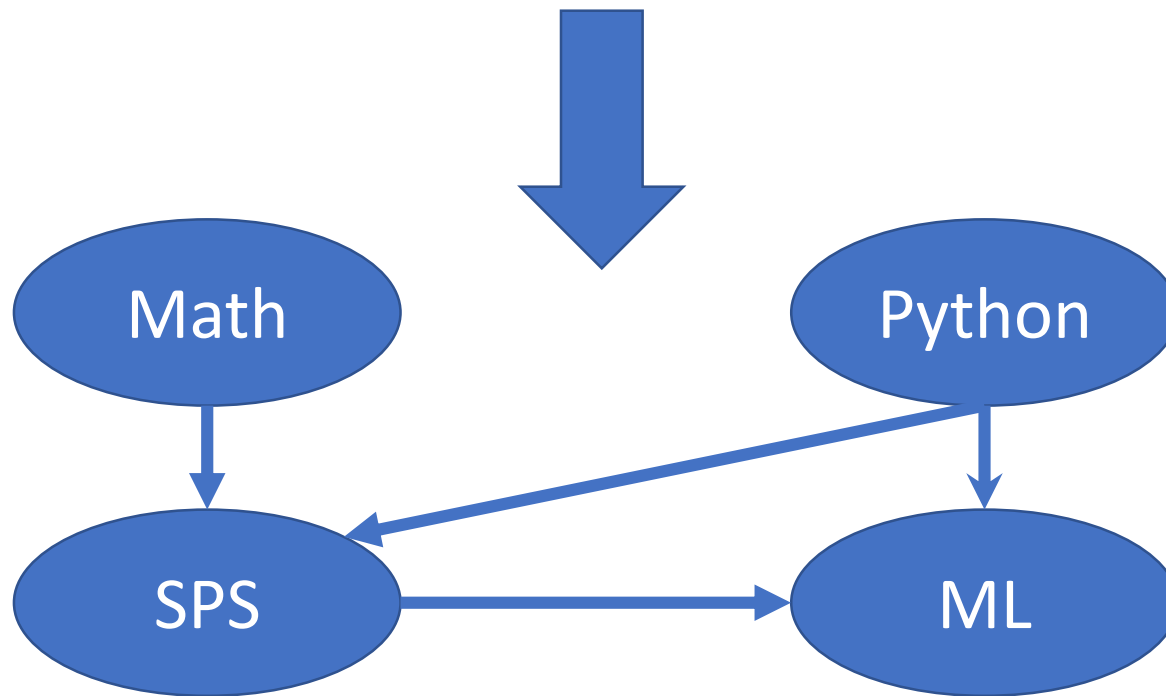
- DAG is usually used to represent causal relationship.
- e.g. high temp yesterday causes high temp today, not **vice versa**!

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
- $p(X) = \prod_v p(X_v | X_{\text{parent}(X_v)})$

Example

- $p(Ma, Py, SPS, ML) = p(Ma)p(Py)p(SPS|Ma, Py)p(ML|SPS, Py)$

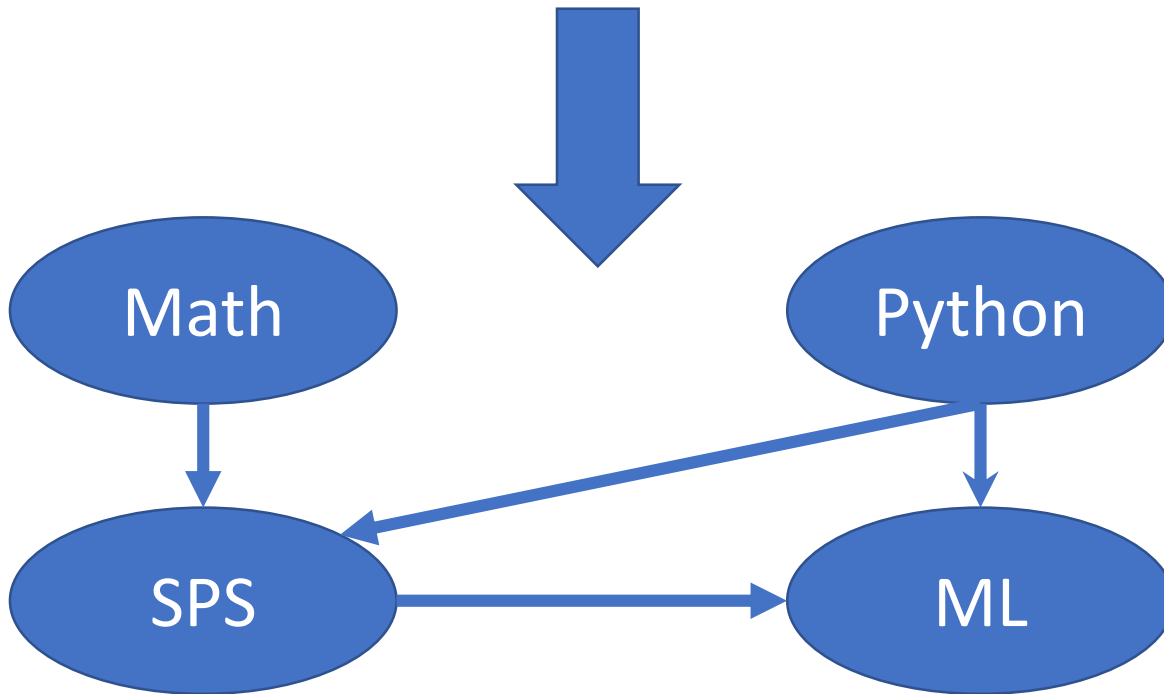


Represent Cond. Indep. using DAG

- 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-descendant 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 .

Example

- $ML \perp Math \mid SPS, Python$
- $Math \perp Python$



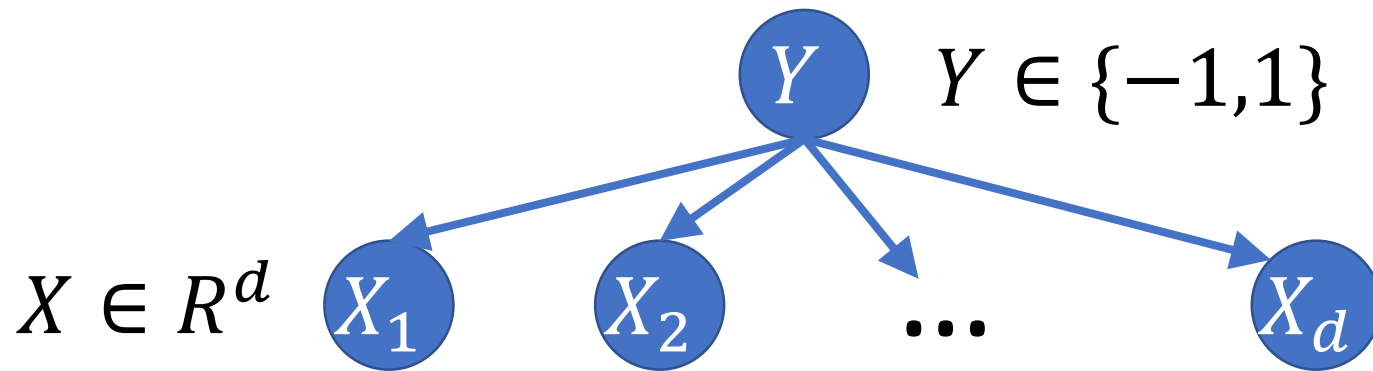
Equivalency between Factorization and Conditional Independence over DAG G

- If p factorizes over G , p satisfies all conditional independence represented by G .
- If p satisfies all conditional independence represented by G , then p factorizes over G .
- **PC**: verify this on unit score example.

Bayesian Network

- A probability dist. $p(x)$ factorizes over a DAG G is called Bayesian network.

Bayesian Network for Classification



- Similar to a logistic model, only now it is a directed graphical model.

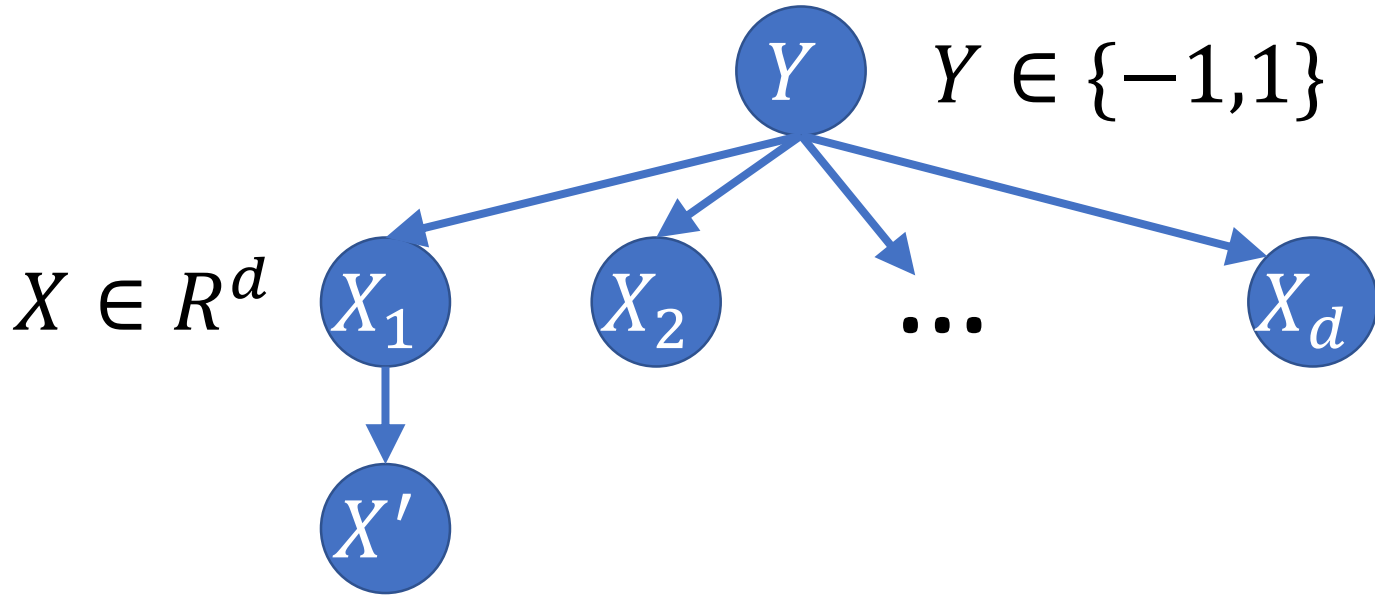
Bayesian Network for Classification

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

Bayesian Network for Classification

- **PC:** Compare NB and Logistic model from the following perspectives:
 - The graphical structure
 - The factorization
 - The probabilistic model
 - The training/fitting of a classifier
 - The usage of a classifier

Question



- PC: Given this Bayesian Net for a classification task, should you include feature X' for training? Why?

Conclusion

- Bayesian Net uses a **DAG** to represent factorization and conditional independence of a probability distribution .
- **Naïve Bayes** is derived from a simplified Bayesian network for conditional probability $P(Y|X)$.