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
…	…	…	…	…

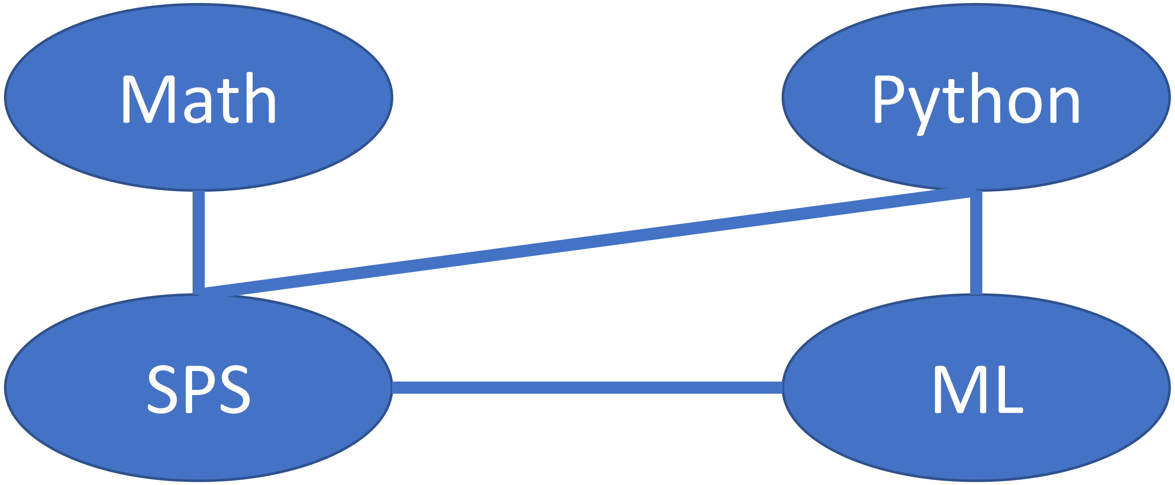
Example: Scores of Units

- Given a dataset $\{\boldsymbol{x}_i\}_{i=1}^{n}$,
  - $\boldsymbol{x}_i = \left[x_i^{(1)}, x_i^{(2)} \dots x_i^{(d)}\right] \in R^d$
  - $\boldsymbol{x}_i$ is a vector of a student i's scores.
  - $x^{(1)}$ is SPS score, $x^{(2)}$ is Math score, … $x^{(d)}$ is Mach. Learn. score.

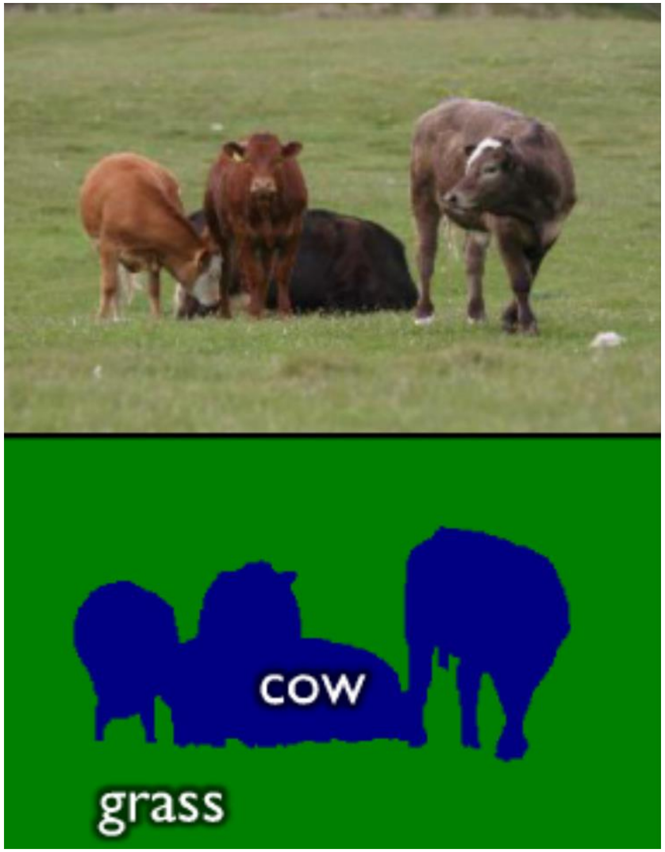
- **What does $p\left(x^{(1)}, x^{(2)} \dots\ x^{(d)}\right)$ look like?**

An (undirected) Graphical Representation of Dependency

- Scores of units are **dependent**!
  - Student with **high** Math, Python score is likely to receive **high** SPS score.
- A graphical representation:



Example: Image Segmentation



- The probability of one pixel being labelled as "Cow" is correlated with **adjacent pixels**.
    - A pixel is more likely to be a Cow pixel if surrounding pixels are all Cow pixels

Jamie Shotton et. al. IJCV 2009

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
  - $\Leftrightarrow p(X|Y) = p(X) \Leftrightarrow p(Y|X) = p(Y)$
    - Information Flow

Example: Likelihood with Independent Datapoints:

- Likelihood over the dataset
  - Factorizes into product over each x_i
  - $p(x_1, x_2, \dots x_n; \theta) = \prod_{i=1}^{n} p(x_i; \theta)$

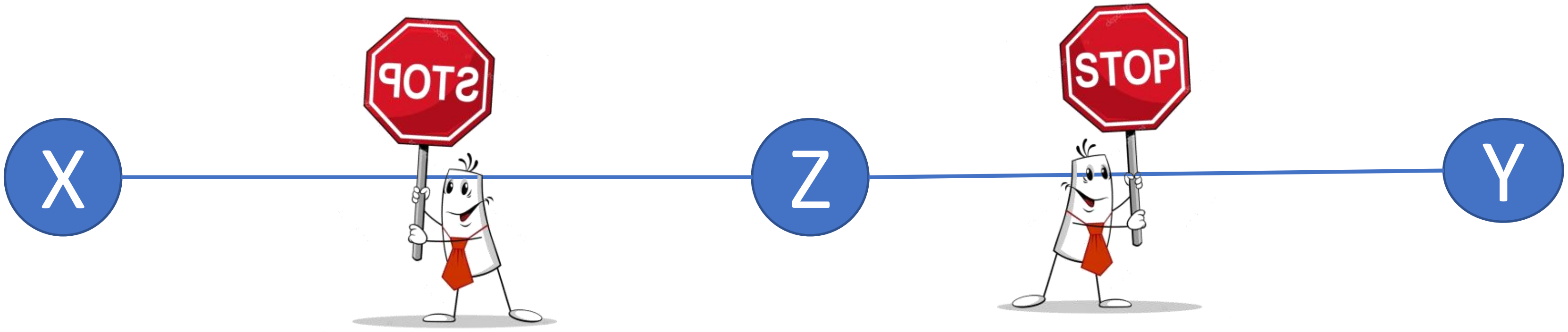
- Maximum Likelihood Estimation
  - $\max_{\theta} \prod_{i=1}^{n} p(x_i; \theta)$
  - **Lab sheet 4.1**

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: **Given** Z, Y does not give any additional info which changes the prob. of X.
  - $\Leftrightarrow p(Y | X, Z) = p(Y | Z)$

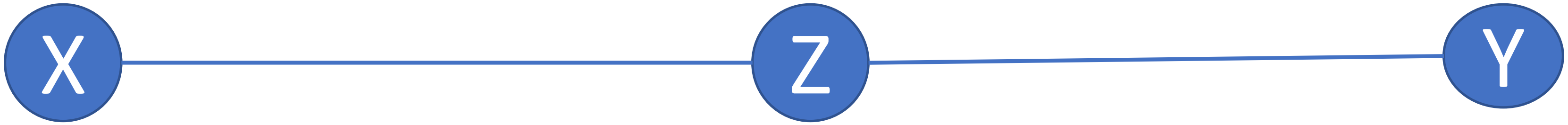
Conditional Independence of R.V.s

- Conditional Independence is interesting since it tells how information **flows** between R.V.s
  - $X \perp Y|Z$ tells information **flows into** X from Y are “bottlenecked” by Z.
  - vice versa.



Representing Conditional Independence by Graph

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



$$“X \perp Y | Z”$$

Representing Conditional Independence by Graph

- Given a graph $G =< E, V >$, and three random variables $X, Y, Z \in V$
  - if X and Y are completely "**blocked**" by Z, we say $X \perp Y|Z$ is represented by G.
  - X, Y, Z can also be a group of R.Vs.

Example: Encoding cond. ind. by graph

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

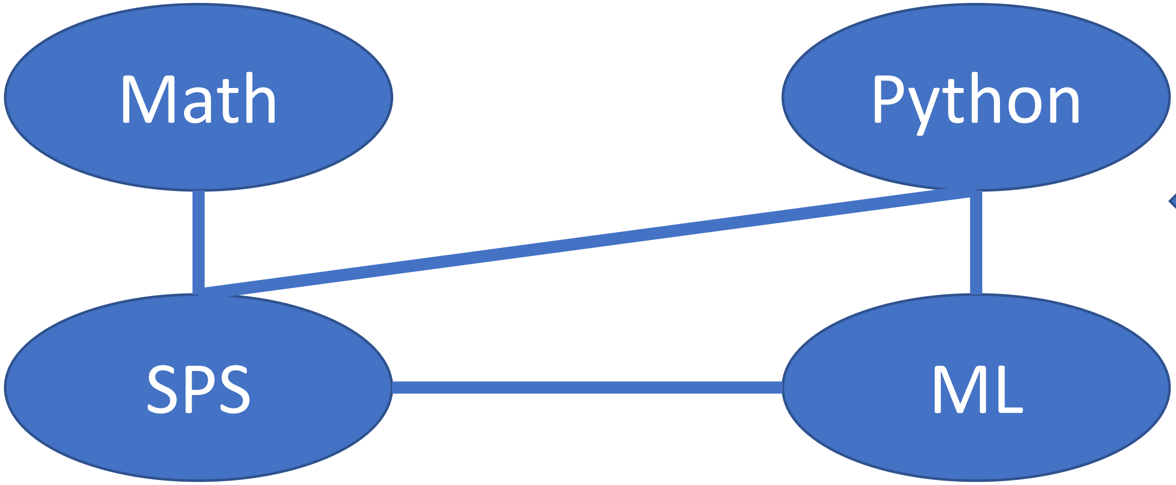
$\text{Math} \perp \text{ML} \mid \text{SPS}$

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

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

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

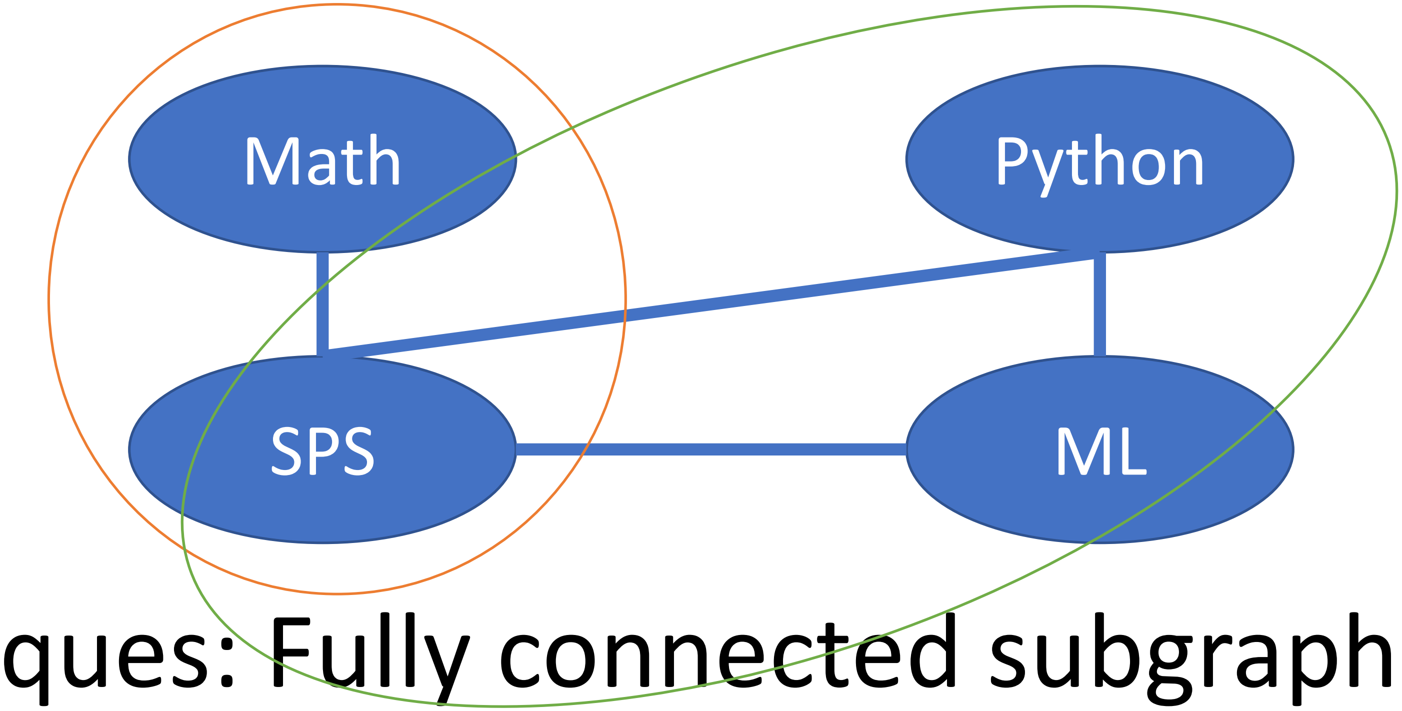
List of conditional independence encoded by Graph!



Representing Prob. Distribution Factorization by Graph

- Writing a factorization of a probability distribution with many R.V. can be messy.
- Given a graph $G =< E, V >$,
- We say $p(X)$ factorizes over G:
- If $p(X) \propto \prod_{c \in C} g_c\left(X^{(c)}\right)$
  - where C is set of all **cliques** in G.
  - g_c is a function defined on $X^{(c)}$, which is the subset of X **restricted on** c.

Example



- Cliques: Fully connected subgraphs.
- Maximum cliques:
  - Math-SPS, SPS-Python-ML
- $p(Ma, SPS, Py, ML) \propto g_1(Ma, SPS) \cdot g_2(Py, ML, SPS)$.

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.
  - Hint: $X \perp Y, W | Z \Rightarrow X \perp Y | Z$

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:
- $\boldsymbol{x} \in R^d, \boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$
- Let $\boldsymbol{\Theta}$ is be the inverse of $\boldsymbol{\Sigma}$.
- $p(\boldsymbol{x}) \propto \exp\left[-\frac{\boldsymbol{x}(\boldsymbol{\Sigma})^{-1}\boldsymbol{x}^{\top}}{2}\right]$

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

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

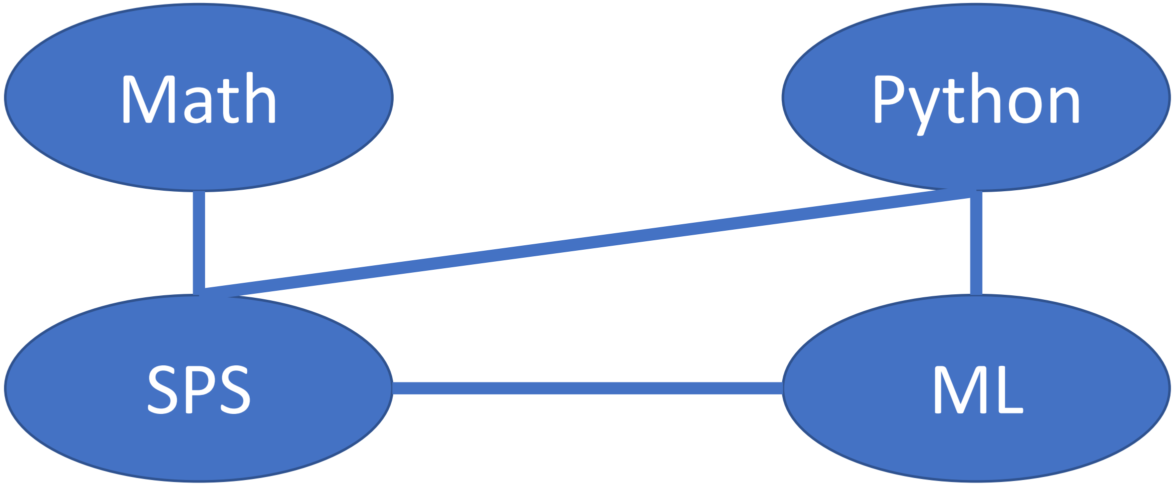
Gaussian Markov Network

- $p(\boldsymbol{x}) \propto \prod_{u,v} g_{u,v}\left(x^{(u)}, x^{(v)}\right)$
    - Edge $\left(X^{(u)}, X^{(v)}\right)$ is a clique!
- $p(\boldsymbol{x})$ **factorizes over structure of $\boldsymbol{\Theta}$!**
    - $\Leftrightarrow$ factorizes over G defined by the adjacency matrix $A_{i,j} = \begin{cases} 0, \Theta_{i,j} == 0 \\ 1, \Theta_{i,j} \neq 0 \end{cases}$
    - G must be an undirected graph (why?)
    - $\Leftrightarrow$ satisfies the conditional independence encoded in G.

Gaussian Markov Network

- **Knowing structure** of $p(\boldsymbol{x})$ in advance, we can hand-craft a Gaussian Markov network model by specifying $\boldsymbol{\Theta}$.
  - **Careful:** $\boldsymbol{\Theta}$ must be **positive definite!**

Example



- $x^{(1)}$:Math; $x^{(2)}$:Py; $x^{(3)}$:SPS; $x^{(4)}$:ML
- $\mathbf{\Theta} = \begin{bmatrix} \Theta_{11} & 0 & \Theta_{13} & 0 \\ 0 & \Theta_{22} & \Theta_{23} & \Theta_{24} \\ \Theta_{13} & \Theta_{23} & \Theta_{33} & \Theta_{34} \\ 0 & \Theta_{24} & \Theta_{34} & \Theta_{44} \end{bmatrix}$

Constructing Likelihood

- **PC:** If $(x_0, \boldsymbol{x})$ are drawn from a joint Gaussian $p(x_0, \boldsymbol{x})$, show log likelihood $\log p(x_0|\boldsymbol{x})$ has the form:
  - $-(x_0 - \sum_i \beta_i x_i)^2/b$, where $\beta_i \neq 0$ iff (X_0, X_i) is an edge in the Markov network structure of p.
  - How does it help us select good features in least squares fitting?

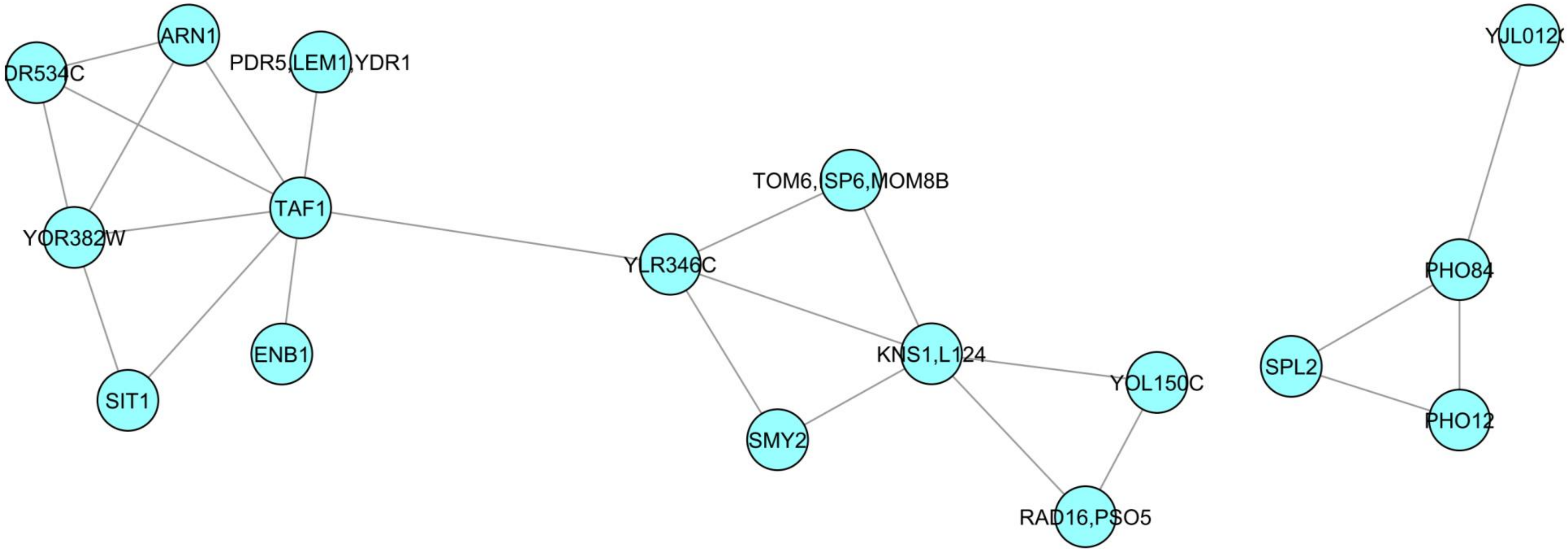
Gaussian Markov Network

- **Not knowing** structure of $p(\boldsymbol{x})$, given dataset D, we can fit a sparse $\boldsymbol{\Theta}$.
  - Using MLE: $\max_{\boldsymbol{\Theta}} \log p(D; \boldsymbol{\Theta})$
  - The sparsity of $\boldsymbol{\Theta}$ gives a graphical representation of $p(\boldsymbol{x})$!
  - Such representation reveals how random variables “interacts” with each other!

Example: Gene Expression Data

Name of Genes	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:
- $p(\boldsymbol{x};\boldsymbol{\theta}) := \dfrac{\exp(\langle\boldsymbol{\theta},\boldsymbol{f}(\boldsymbol{x})\rangle)}{Z(\boldsymbol{\theta})}$
  - $\boldsymbol{f}(\boldsymbol{x})$ is a feature transform on $\boldsymbol{x}$.
  - $Z(\boldsymbol{\theta}) := \int \exp(\langle\boldsymbol{\theta},\boldsymbol{f}(\boldsymbol{x})\rangle) d\boldsymbol{x}$
- PC: show when $\boldsymbol{f}$ is 2nd degree poly. transform with pairwise terms, $p(\boldsymbol{x};\boldsymbol{\theta})$ is a multivariate Gaussian distribution.

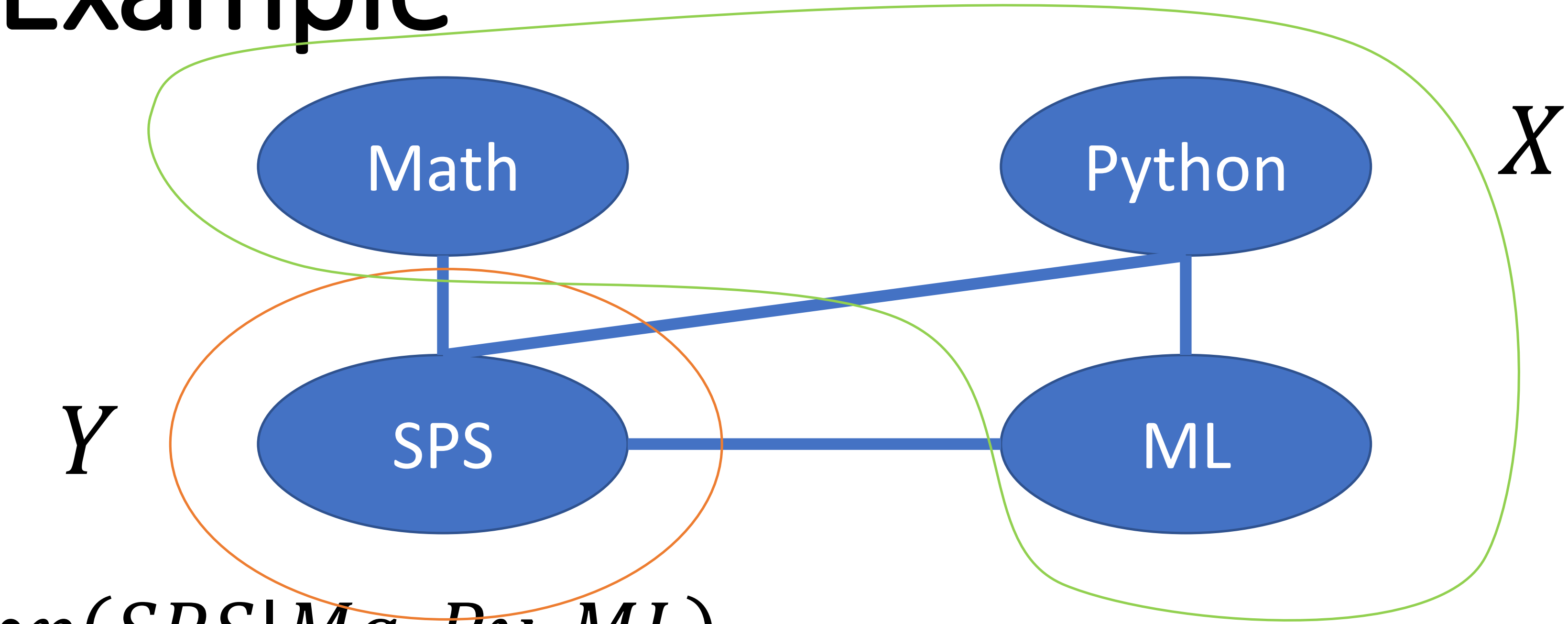
Conditional Markov Network

- 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
- $p(Y|X) = \frac{1}{Z(X)} \prod_{c \in C} g_c(Z)\,, Z \subseteq X \cup Y$
- $Z(X) \coloneqq \int \prod_{c \in C} g_c(Z)\, dY$
- PC: show $Z \nsubseteq X$
  - $p(Y|X) =$does not include factors on conditioning variable X!

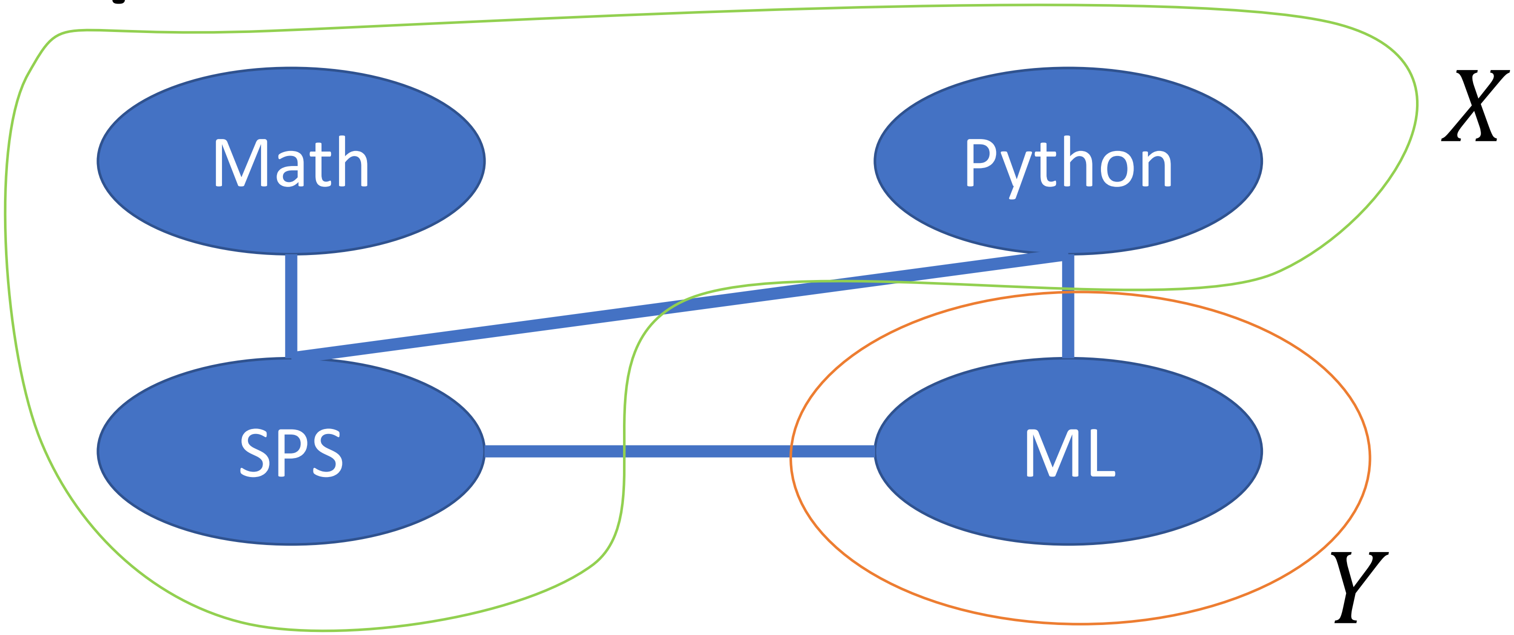
Example



- $p(SPS|Ma,Py,ML) = \frac{1}{Z(Ma,Py,ML)} g_1(SPS,Py,ML) g_2(SPS,Ma)$

- $Z(Ma,Py,ML) = \int g_1(SPS,Py,ML) g_2(SPS,Ma) dSPS$

Example



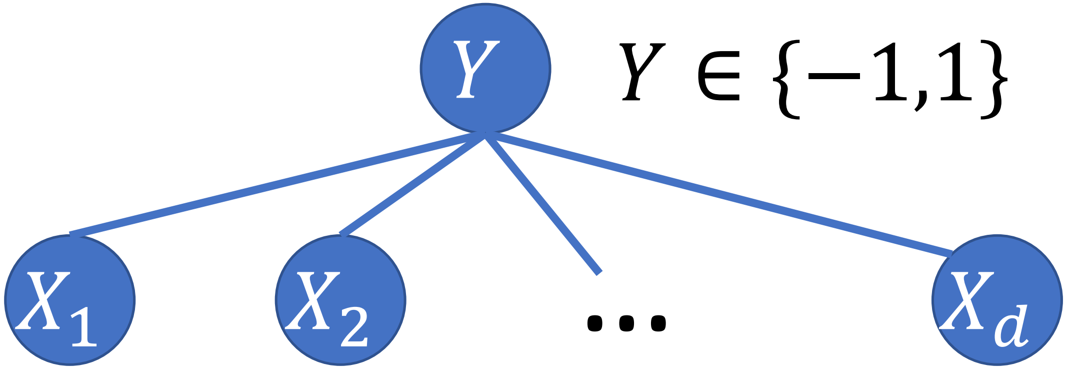
- $p(ML|Ma, Py, SPS)$
$$= \frac{1}{Z(Ma, Py, SPS)} g_1(SPS, Py, ML)$$

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



$Y \in \{-1,1\}$

$X \in R^d$

Logistic Regression

- Using the factorization rule above,
    - $p(Y|X) = \frac{1}{Z(X)} \prod_i g_i(Y, X^{(i)})$
    - $Z(X) = \sum_{c \in \{-1,1\}} \prod_i g_i(Y, X^{(i)})$
- Let $g_i\left(Y = y, X_i = x^{(i)}\right) := \exp\left(\beta_i \cdot yx^{(i)}\right)$
    - $p(y|\boldsymbol{x}) = \frac{1}{Z(X)} \exp\left(\sum_i \beta^{(i)} \cdot yx^{(i)}\right)$

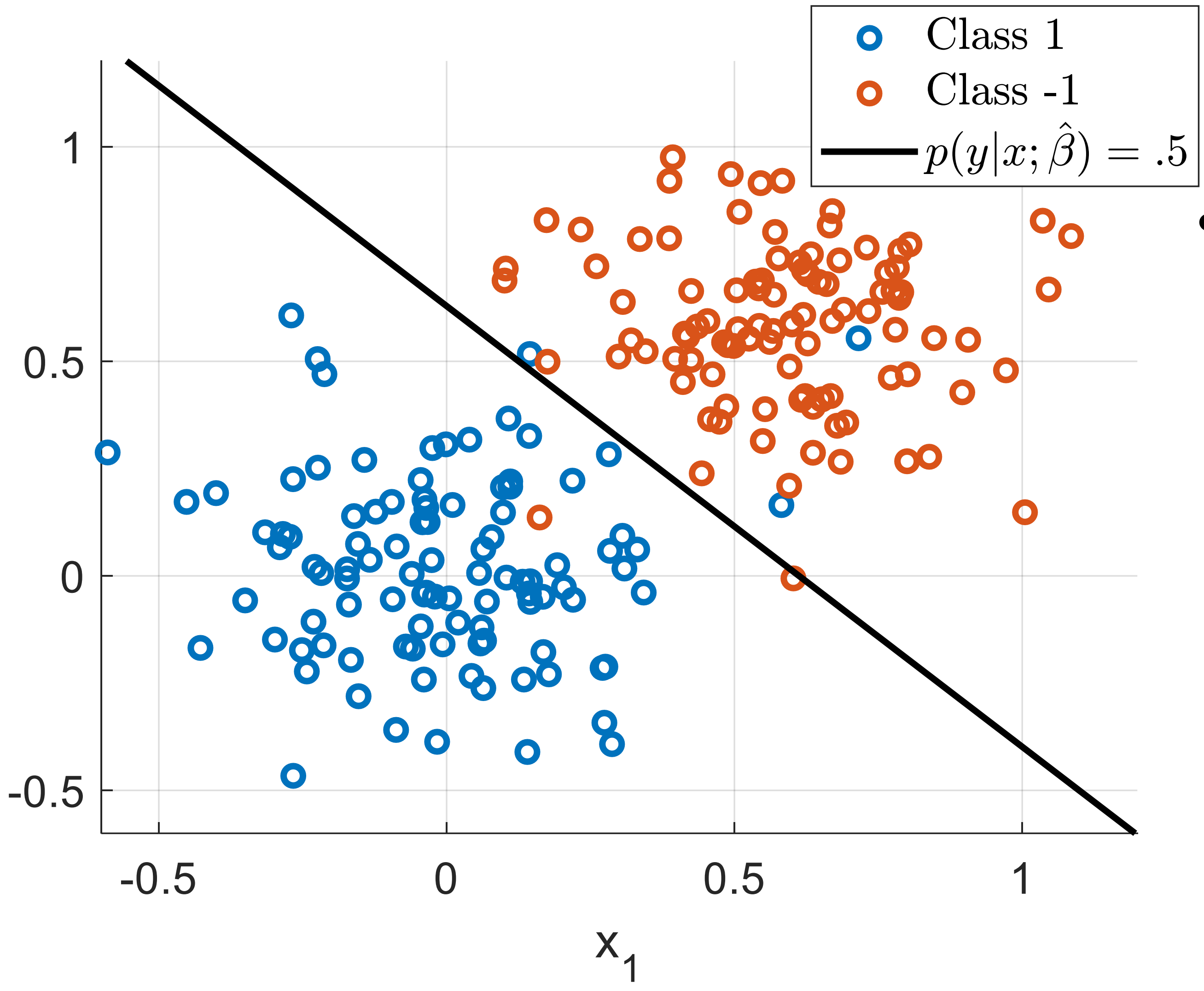
      $= \frac{1}{Z(X)} \exp(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle y).$
    - $Z(X) = \exp(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) + \exp(-\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)$

Logistic Regression

- Logistic model:
- $p(y|\boldsymbol{x};\boldsymbol{\beta}) = \frac{1}{Z(X)}\exp(\langle\boldsymbol{\beta},\boldsymbol{x}\rangle y)$
- $Z(x) = \exp(\langle\boldsymbol{\beta},\boldsymbol{x}\rangle) + \exp(-\langle\boldsymbol{\beta},\boldsymbol{x}\rangle)$

- $\boldsymbol{\beta}$ can be fitted using MLE.
  - $\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log p(y_i|\boldsymbol{x}_i;\boldsymbol{\beta})$
  - 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 $p\left(y\middle|x;\hat{\beta}\right)$, which is more interpretable!

Conclusion

- Markov network uses a graph to represent its conditional independencies.
  - It visualizes interactions of R.V.s in a P.D.

- Two examples of Markov network
  - Gaussian Makov network factorizes over the graph defined by its **inverse covariance**.
  - Logistic model is a conditional P.D. model factorizes over a classification model