

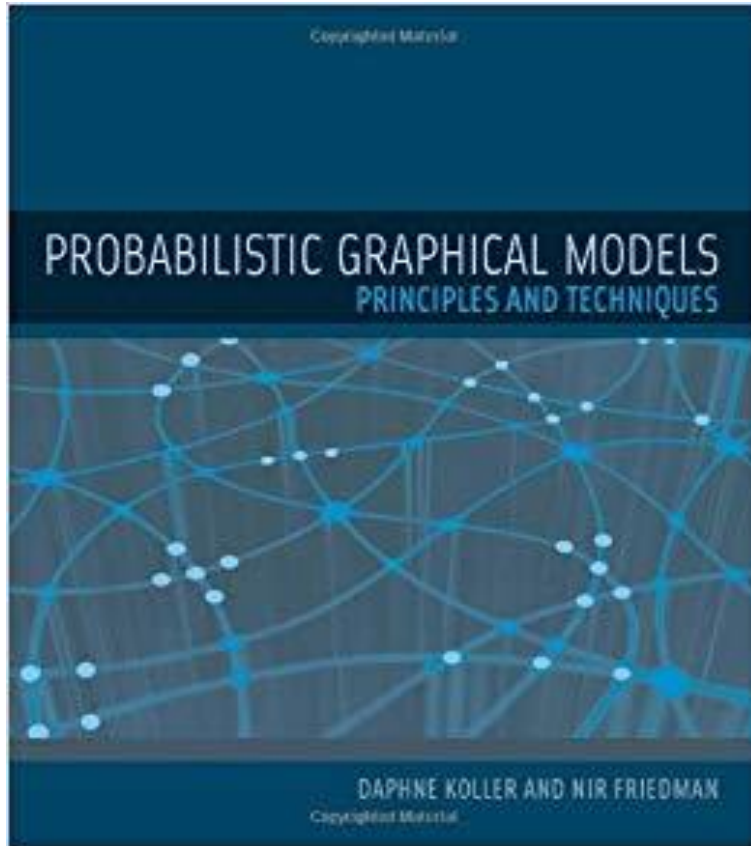
# Capturing Dependency of Data using Graphical Models

Song Liu  
([song.liu@bristol.ac.uk](mailto: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

# References



- Today's class roughly follows Chapter 2.14 and Chapter 4 in Probabilistic Graphical Models by Koller and Friedman.

# Example: Scores of Units

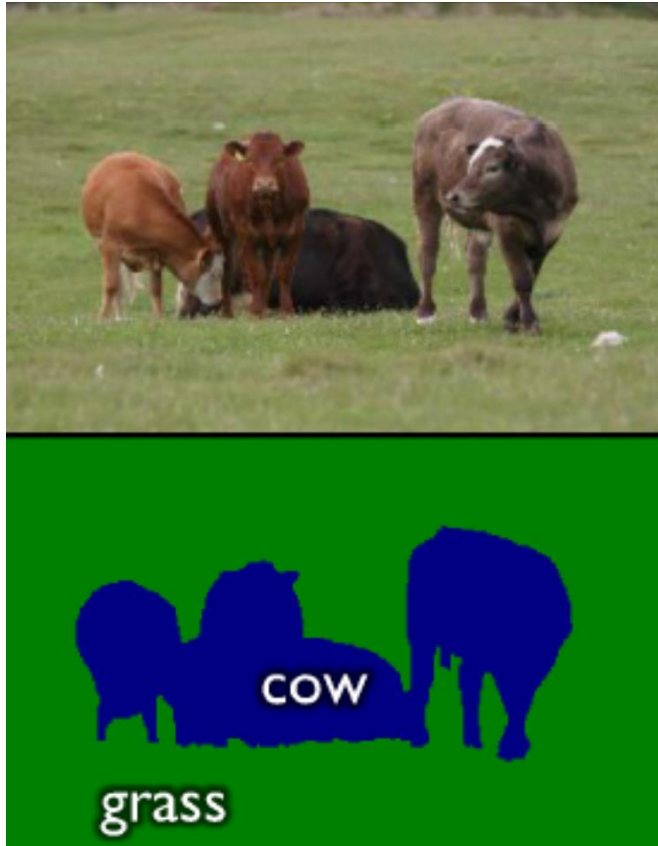
- Imagine a table of unit scores.

Name	SM1	Math	Python	Mach. Learn.
Song	80	70	50	60
Harry	50	40	70	80
Ron	50	50	...	45
Hermione	90	100	...	100
...	...	...	...	...

# Dependency of R.V.s and Probabilistic Models

- How do you construct a good  $p(D|\theta)$  as the likelihood of this dataset?
- Scores of units are **dependent!**
  - Student with **high** Math, Python score is likely to receive **high** SM1 score.
  - Student with **high** SM1 score is likely to receive a **high** Mach. Learn. score.

# Example: Pixel Correlation



- The likelihood of one pixel being “Cow” is dependent with labels of **adjacent pixels**.

How the dependencies between R.V.s would affect likelihood modelling?

# Dependency and Likelihood

- If we assume  $x_1 \dots x_n$  are IID.
- Likelihood 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
  - $\max_{\theta} \prod_{i=1}^n p(x_i | \theta)$
  - **First Lecture!**



# Dependency and Likelihood

- IIDness is an extremely simple assumption.
- What about complicated dependencies?
  - How do we factorize our likelihood?
- To solve this problem, we **first** need to convert our dependences into a graphical representation, **then** use the graph to guide our factorization.
- Study of factorization of prob. distributions and dependencies of R.V.s is called **graphical modelling**.

# Review: Independence and Conditional Independence

# Independence of R.V.s

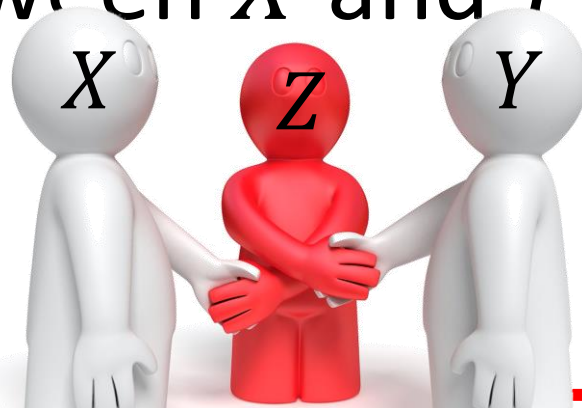
- Let's look at how independence between R.V.s are **expressed in probability distribution**:
- R.V.  $X$  is **independent** of  $Y$ :
  - $X \perp Y$
  - $\Leftrightarrow p(X, Y) = p(X)p(Y)$ 
    - Factorization
  - $\Leftrightarrow p(X|Y) = p(X) \Leftrightarrow p(Y|X) = p(Y)$ 
    - No Information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

# Conditional Independence of R.V.s

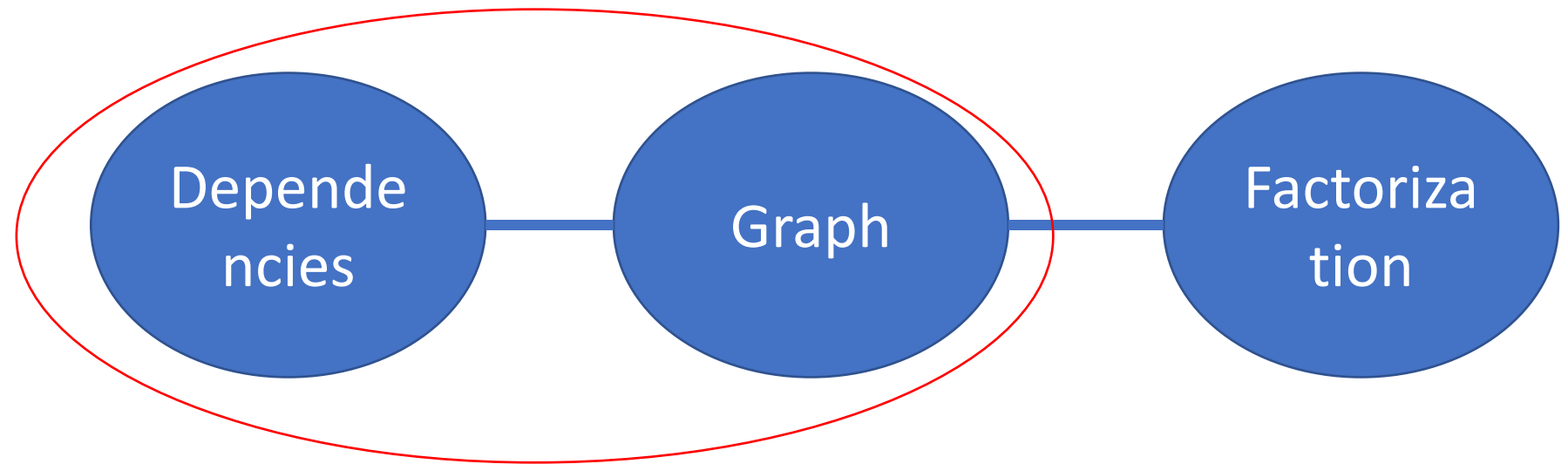
- R.V.  $X$  is independent of  $Y$  **given**  $Z$ 
  - $X \perp Y | Z$
  - $\Leftrightarrow p(X, Y | Z) = p(X | Z)p(Y | Z)$
  - $\Leftrightarrow p(X, Y, Z) \propto g_1(X, Z) \cdot g_2(Y, Z)$ 
    - Factorization
  - $\Leftrightarrow p(X | Y, Z) = p(X | Z)$ 
    - $Y$  does not give any additional info which changes the prob. of  $X$  given  $Z$ .
    - No **direct** information exchange between  $X$  and  $Y$
  - $\Leftrightarrow p(Y | X, Z) = p(Y | Z)$

# (Conditional) Independence and Information Exchange

- (Conditional) Independ. tells how information **exchange** between R.V.s
  - $X \perp Y \Leftrightarrow$  no information exchanges in-between  $X$  and  $Y$ .
  - $X \perp Y | Z \Leftrightarrow$  no **direct** information exchanges between  $X$  and  $Y$

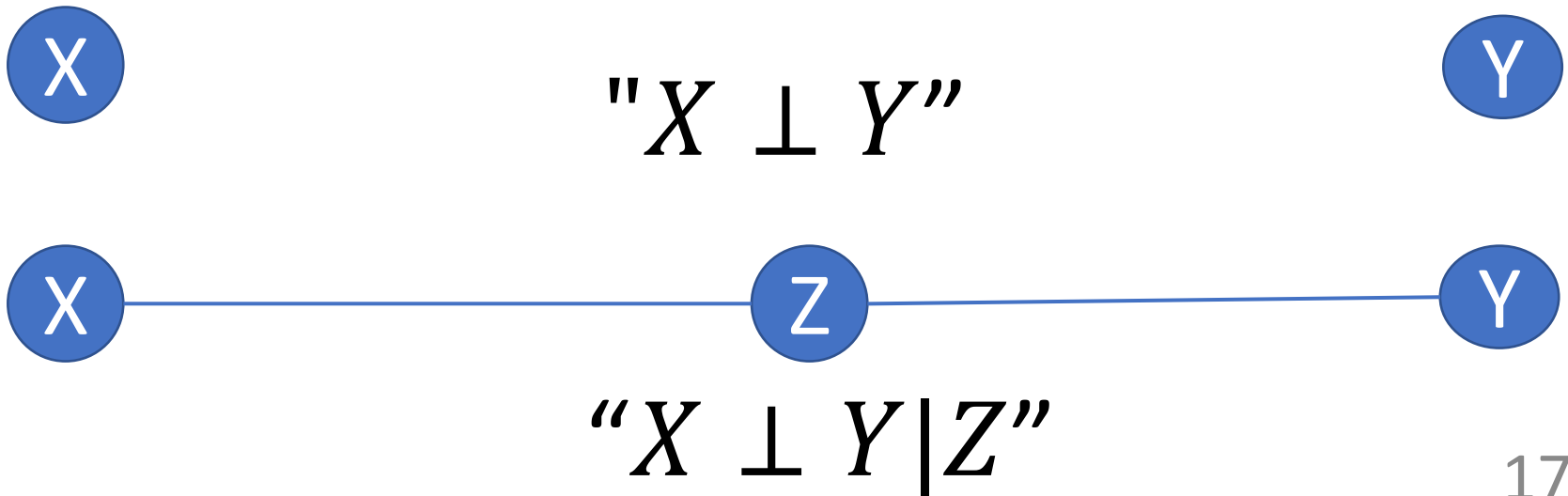


# Creating a Graph of Independence



# 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  $G = \langle E, V \rangle$ ,
  - $V$  contains all the R.V.
- Given three subsets of R.V.:  $X, Y, Z \subseteq V$ 
  - if  $X$  and  $Y$  are completely “**blocked**” in the graph by  $Z$ , we say  $X \perp Y | Z$  is represented by  $G$ .



# Example: Encoding (cond.) indep. by graph

Math  $\perp$  ML | SM1

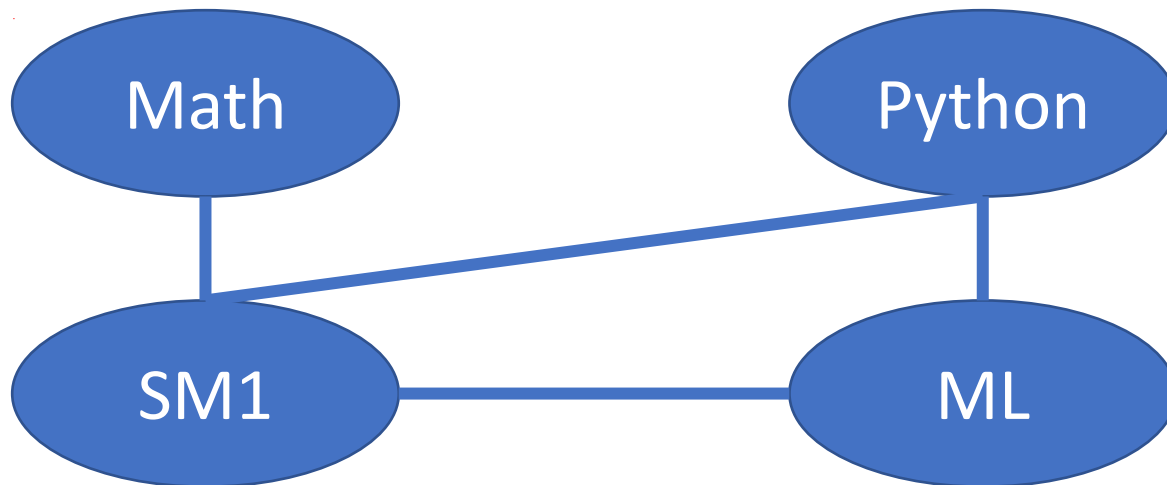
Math  $\perp$  Python | SM1

Math  $\perp$  ML | SM1, Python

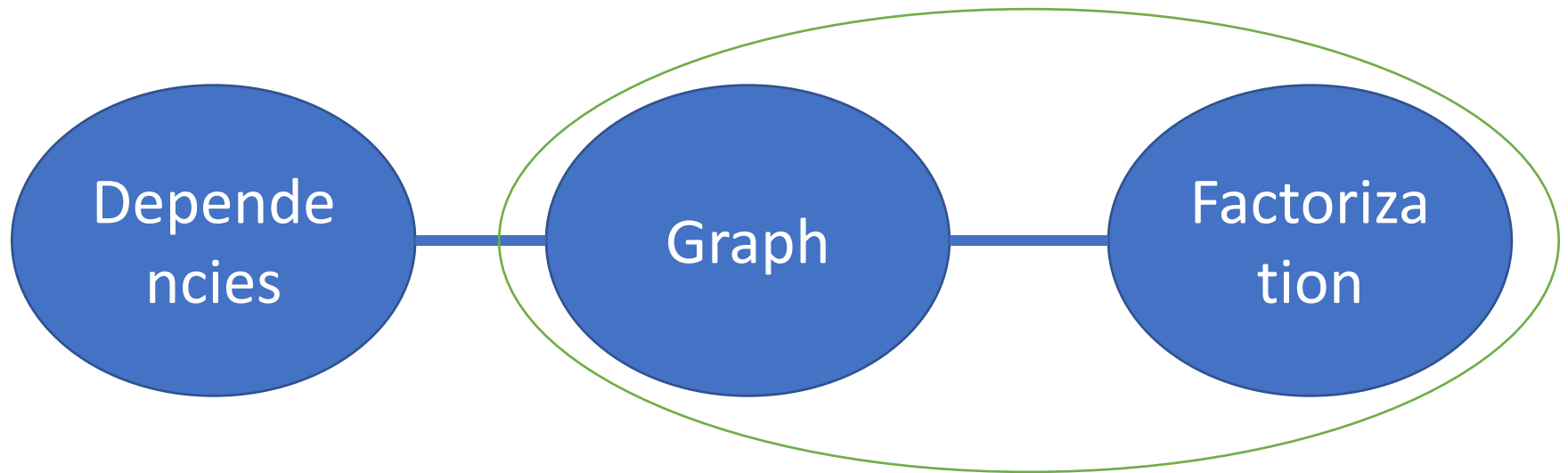
Math  $\perp$  Python, ML | SM1

Math  $\perp$  Python | SM1, ML

List of  
conditional  
independen  
ce encoded  
by Graph!



# Graph and Factorization

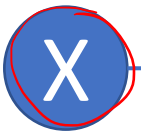


# Representing Prob. Distribution Factorization by Graph

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



$$\underline{P(X, Y)} = \underline{P(X)} \underline{P(Y)}$$



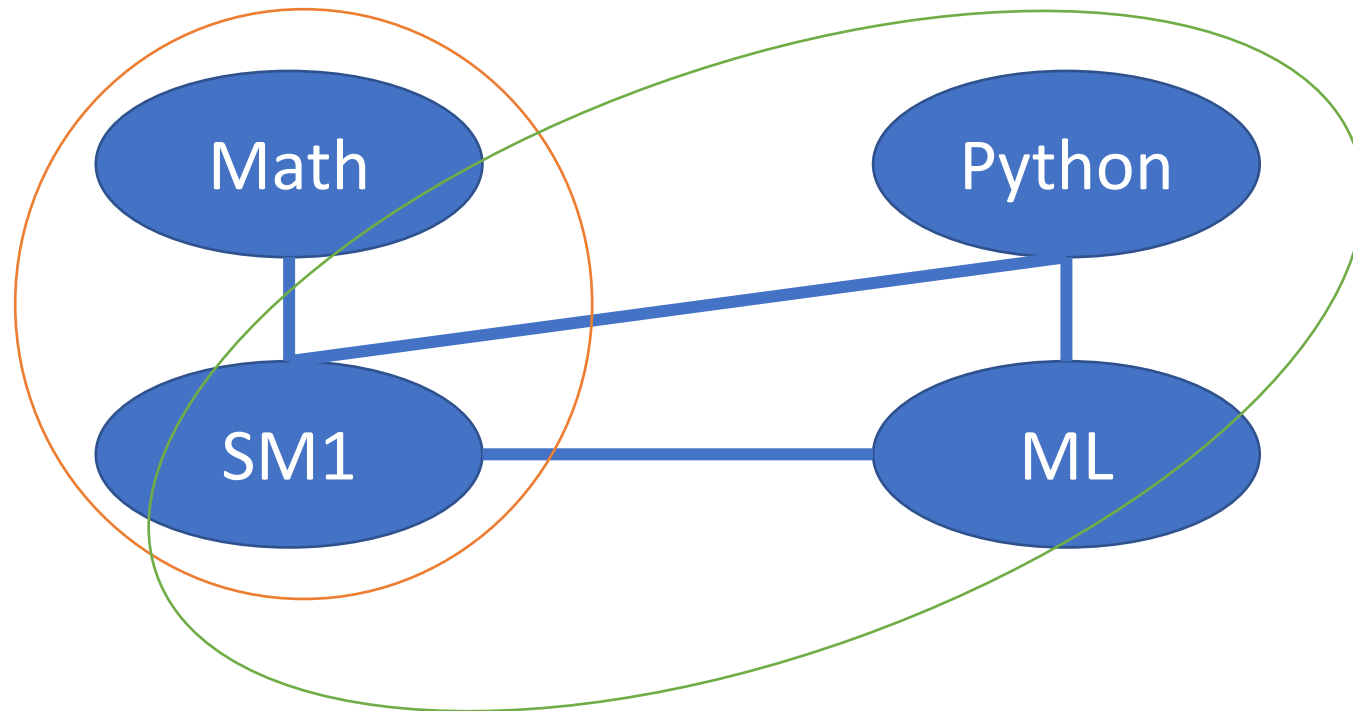
$$P(X, Y, Z) \propto \underline{g_1(X, Z)} \underline{g_2(Y, Z)}$$

# 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

$$p(Ma, SM1, Py, ML) \\ \propto g_1(Ma, SM1) \cdot g_2(Py, ML, SM1).$$



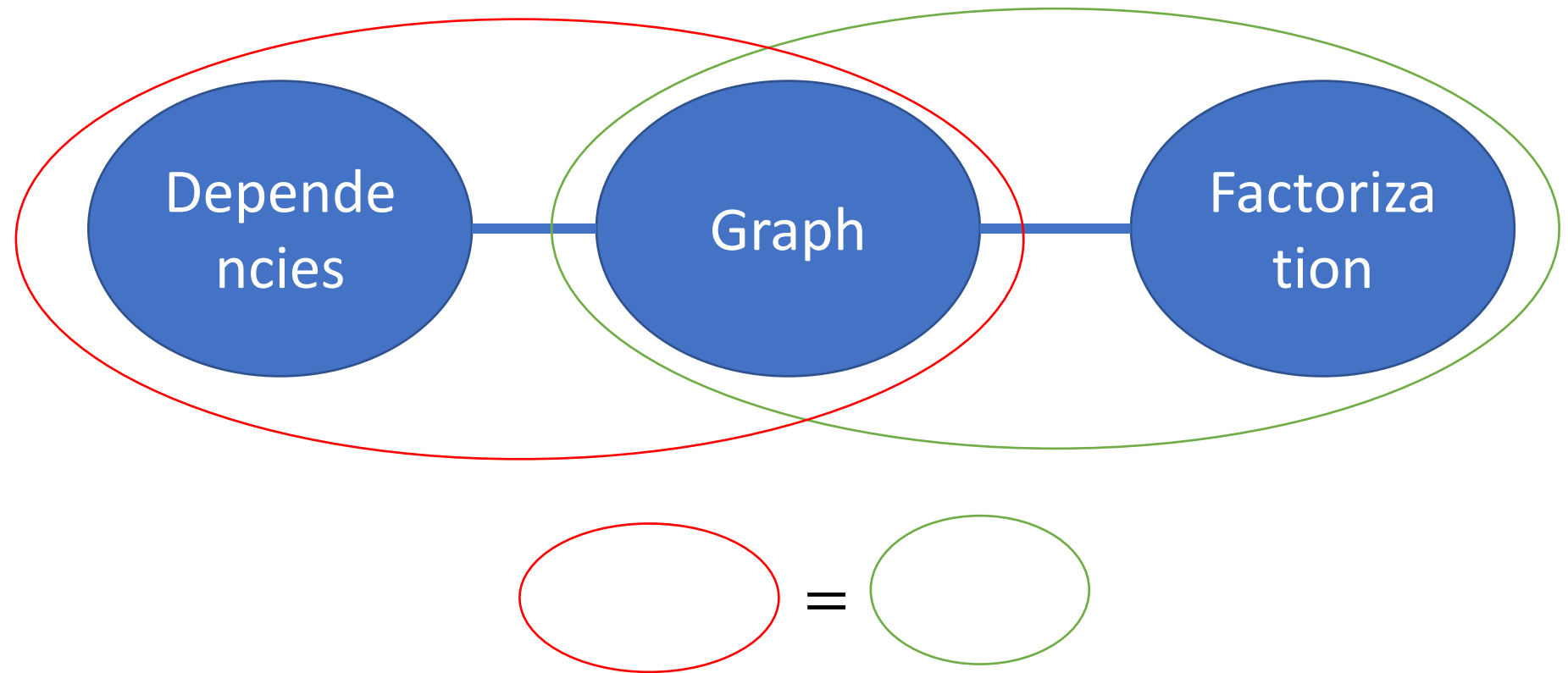
# Equivalency between Factorization and Conditional Independence over $G$

- 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  $G$ .

# Dependencies, Graph, Factorization





# Equivalency between Factorization and Conditional Independence over $G$

- Verify this on Scores of Units example!
- **Homework:**
  - Create  $G$  using factorization on page 24 and check if the graph encodes all conditional Independence of  $p(Ma, SM1, Py, ML)$ .
  - Create  $G$  using all conditional independence on page 19 and check if it encodes the factorization of  $p(Ma, SM1, Py, ML)$ .

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

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

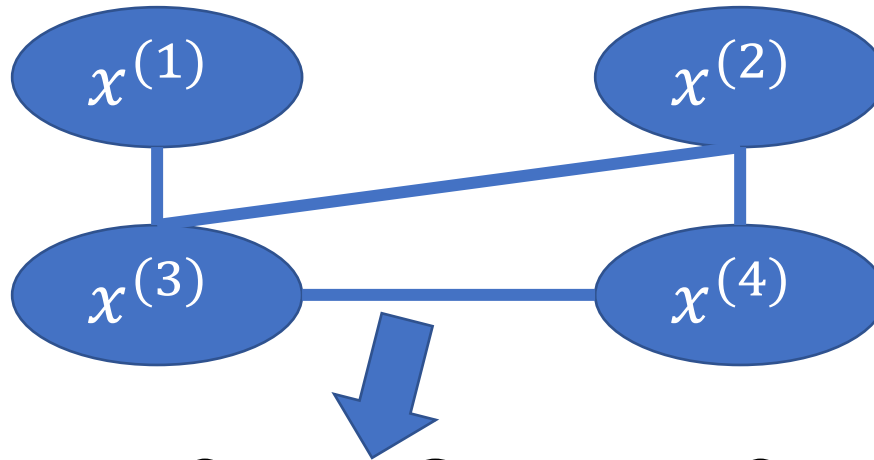
# Gaussian Markov Network

- $p(\mathbf{x}) \propto \prod_{u,v; \Theta(u,v) \neq 0} g_{u,v}(x^{(u)}, x^{(v)})$
- $p(\mathbf{x})$  **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?)

# Gaussian Markov Network

- Give me a graph  $G$  that encodes all conditional independence of your Gaussian R.V., I can infer the sparsity of your  $\Theta$ .

# Example

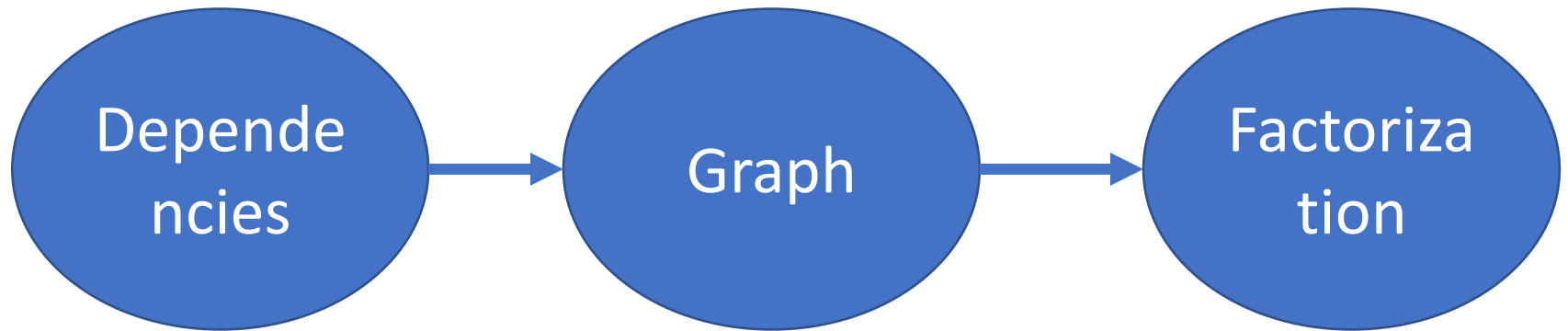


$$\bullet \Theta = \begin{bmatrix} \Theta_{11} & 0 & \Theta_{13} & 0 \\ 0 & \Theta_{22} & \Theta_{23} & \Theta_{24} \\ \Theta_{13} & \Theta_{23} & \Theta_{33} & \Theta_{34} \\ 0 & \Theta_{24} & \Theta_{34} & \Theta_{44} \end{bmatrix}$$

sparsity  
of  $\Theta$  =  
sparsity  
of the  $G$ !

- If I know dependency of my R.V.s, I can easily write down my Gaussian model.

# From Dependency to Factorization



# Homework question:

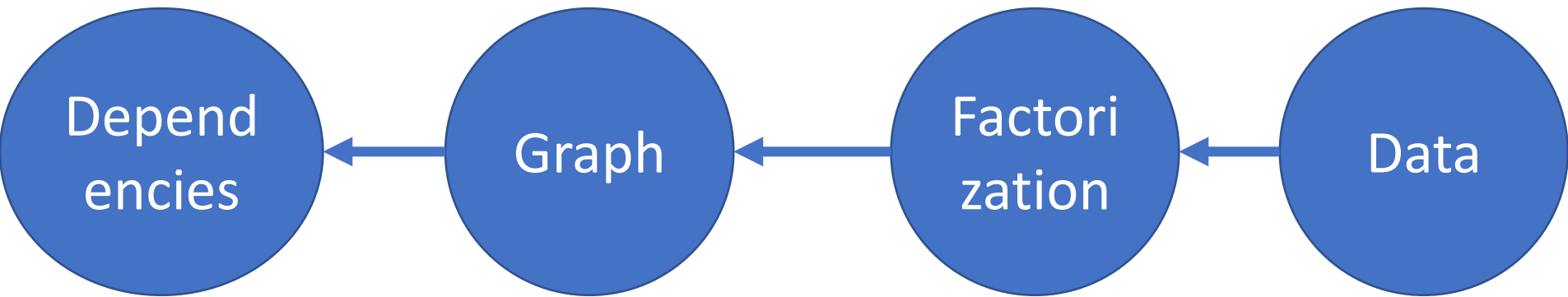
- Suppose graph  $G$  encodes all cond. indep. in your Gaussian distribution  $p$ .  $G$  contains **three edges, five nodes**. How many **non-zero elements** are there in **inverse covariance matrix** of  $p$ ?
- A.3
- B.8
- C.6
- D.10
- E.11



# Gaussian Markov Network

- If we do not know, cond. independence of  $p(\mathbf{x})$ , we can infer it from data!
- Given dataset  $D$ , we can fit a  $\hat{\Theta}$ .
  - Using MLE:  $\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log p(D; \Theta)$
  - The sparsity of  $\hat{\Theta}$  gives a graph corresponds to factorization of  $p(\mathbf{x})$ !
  - Such graph also reveals how R.V. are dependent on each other!

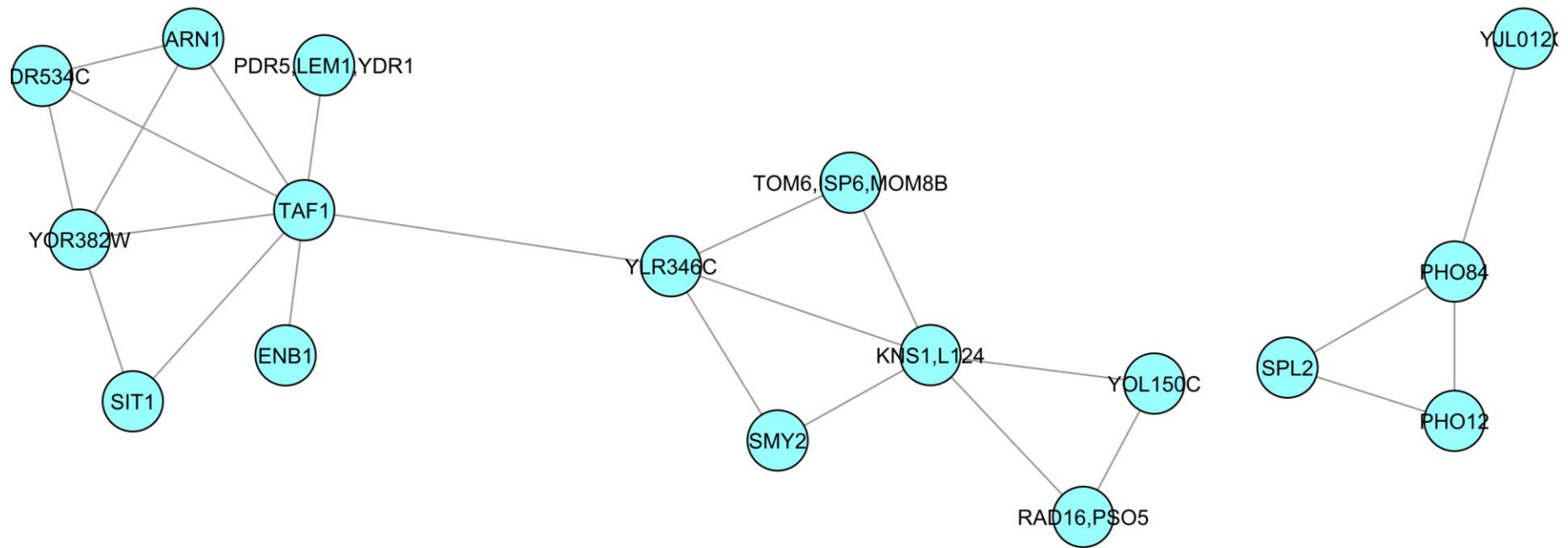
# From Factorization to Graph



# 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](#))



# Graphical Lasso ([Jerome Friedman et al., 2008](#))

- Given  $D = \{\mathbf{x}_i\}_{i=1}^n$ ,  $\mathbf{x} \in \mathbb{R}^d$ ,
- Construct a Gaussian likelihood:
  - $p(D|\Theta) = \prod_i N_{\mathbf{x}_i}(\mathbf{0}, \Theta^{-1})$
- $\hat{\Theta} := \operatorname{argmax}_{\Theta} \log p(D|\Theta) - \lambda \|\Theta\|_1$ 
  - $= \operatorname{argmax}_{\Theta} -\operatorname{tr}(\mathbf{S}\Theta) + \log \det \Theta - \lambda \|\Theta\|_1$
  - $\mathbf{S}$ : sample cov;  $\|\Theta\|_1 = \sum_{i,j} |\Theta^{(i,j)}|$
- Construct a graph using sparsity of  $\hat{\Theta}$

# 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
- $p(Y|X) = \frac{1}{N(X)} \prod_{c \in \mathcal{C}} g_c(V_c)$ ,
  - $\mathcal{C} := \{c \text{ is a clique in } G \mid V_c \not\subseteq X\}$
- $N(X) := \int \prod_{c \in \mathcal{C}} g_c(V_c) dY$
- Normalizing constant:
  - It normalizes the distribution to 1 over the domain of the random variable ( $Y$ ).

# Conditional Markov Network

- $p(Y|X)$  does not include factors defined on subsets of conditioning variable  $X$ !

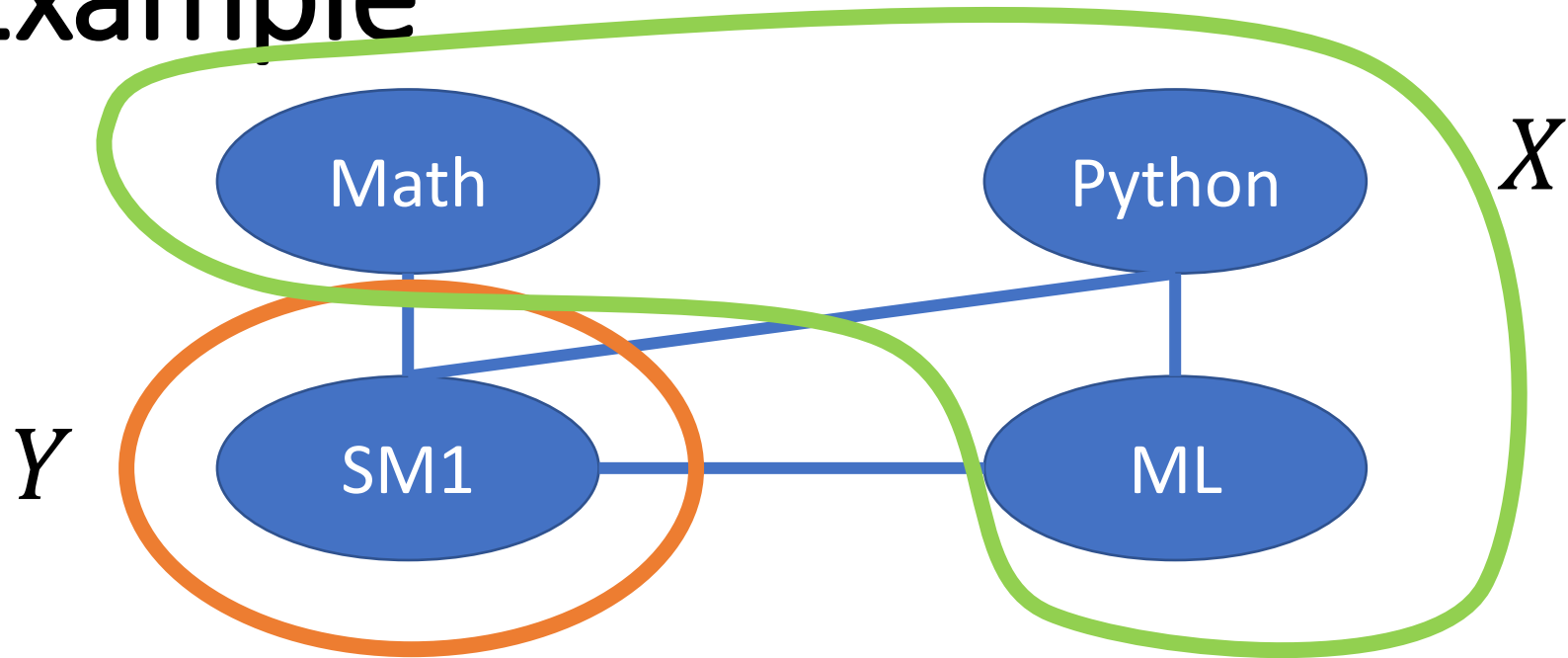
- $p(Y|X) = \frac{1}{N(X)} g_1(Y, X) g_2(X)$

- $N(X) = \int g_1(Y, X) g_2(X) dY = g_2(X) \int g_1(Y, X) dY$

- $p(Y|X) = \frac{g_1(Y, X) g_2(X)}{g_2(X) \int g_1(Y, X) dy} = \frac{g_1(Y, X)}{\int g_1(Y, X) dy}$



# Example

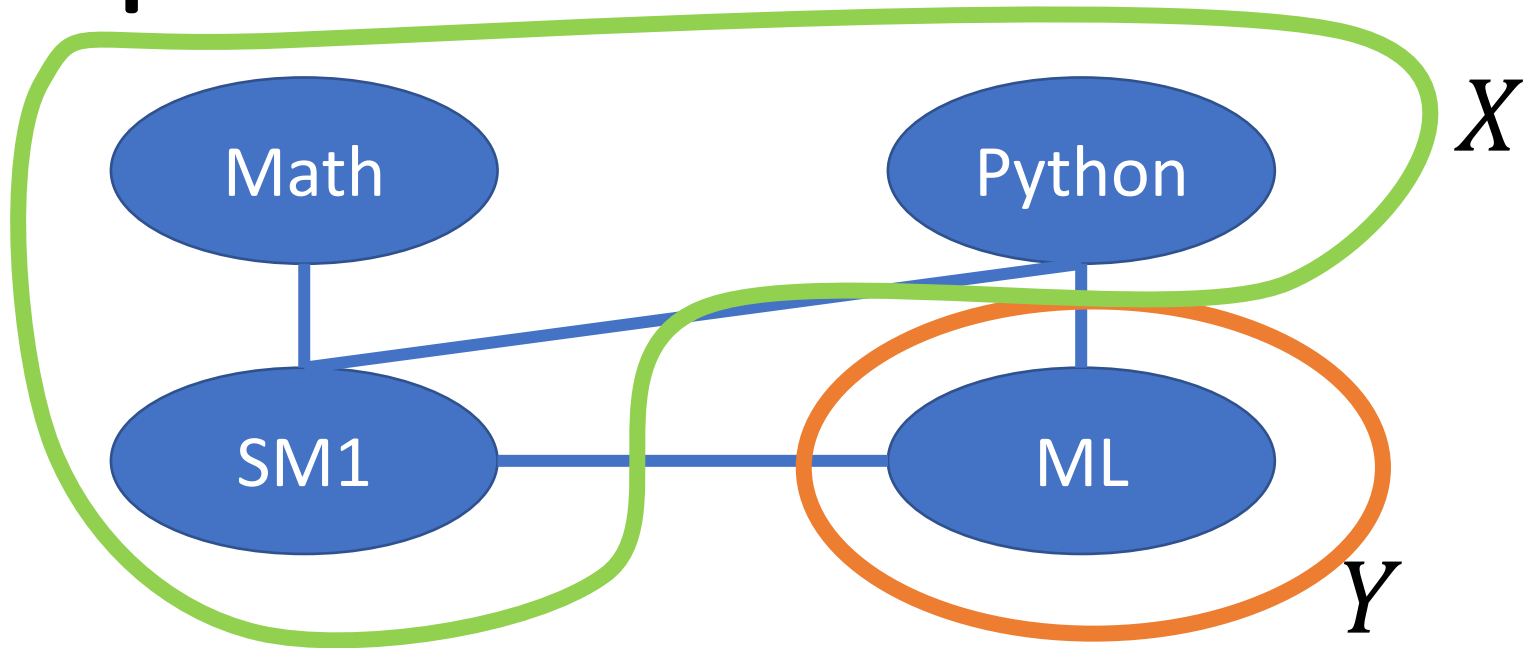


- $$p(SM1 | Ma, Py, ML)$$

$$= \frac{1}{N(Ma, Py, ML)} g_1(SM1, Py, ML) g_2(SM1, Ma)$$

- $$N(Ma, Py, ML) = \int g_1(SM1, Py, ML) g_2(SM1, Ma) dSM1$$

# Example



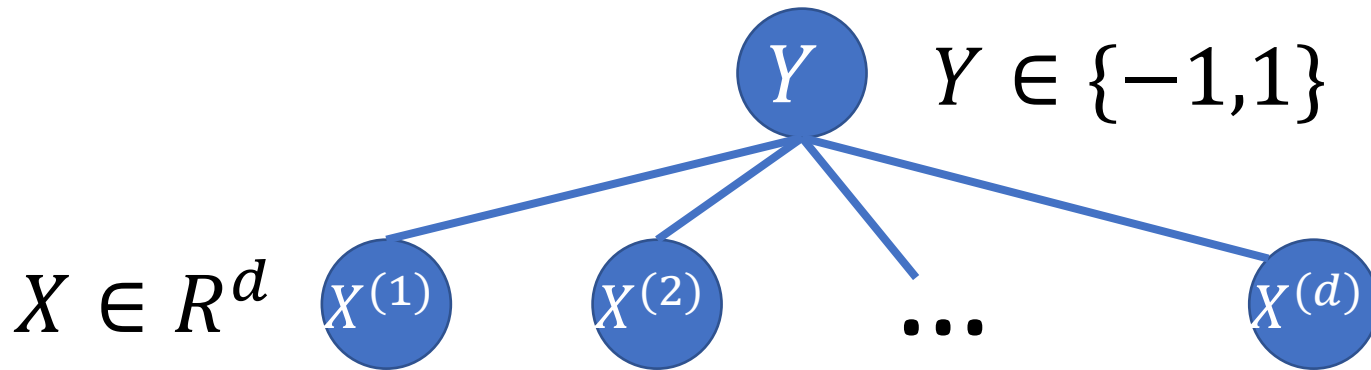
- $p_1(ML|Ma, Py, SM1)$   
$$= \frac{1}{N(Py, SM1)} g_1(SM1, Py, ML)$$

- $N(SM1, Py) = \int g_1(SM1, Py, ML) dML$

- $g_2$  is gone! Math is gone!

# Logistic Regression

- This way of constructing a conditional likelihood gives us: Logistic Regression.
- Consider a simple Markov Net

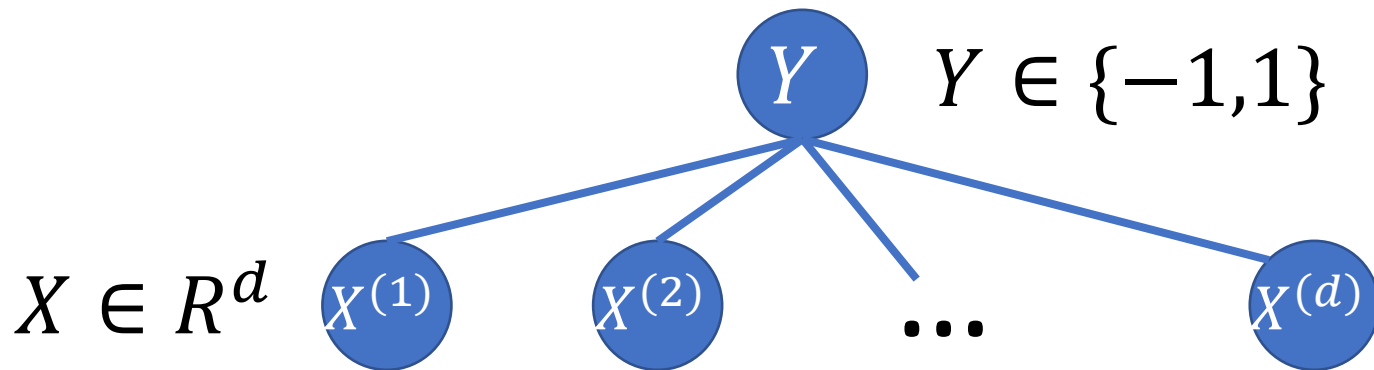


# Logistic Model

- Using the factorization rule above,

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

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



# Logistic Model

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

- By setting

$$g_i(Y = y, X_i = x^{(i)}; \beta_i, \beta_0) := \exp(y(\beta^{(i)} \cdot x^{(i)} + \beta_0))$$

- $$p(y|\mathbf{x}; \boldsymbol{\beta}, \beta_0) = \frac{1}{N(X)} \prod_i \exp(y(\beta^{(i)} \cdot x^{(i)} + \beta_0))$$
$$= \frac{1}{N(X)} \exp(y(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0)).$$

- $$N(X; \boldsymbol{\beta}, \beta_0) = \sum_{y \in \{1, -1\}} \exp(y(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0))$$

# Logistic Regression

- Logistic model:

- $p(y|\mathbf{x}; \boldsymbol{\beta}, \beta_0) = \frac{1}{N(\mathbf{x})} \exp(y(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0))$

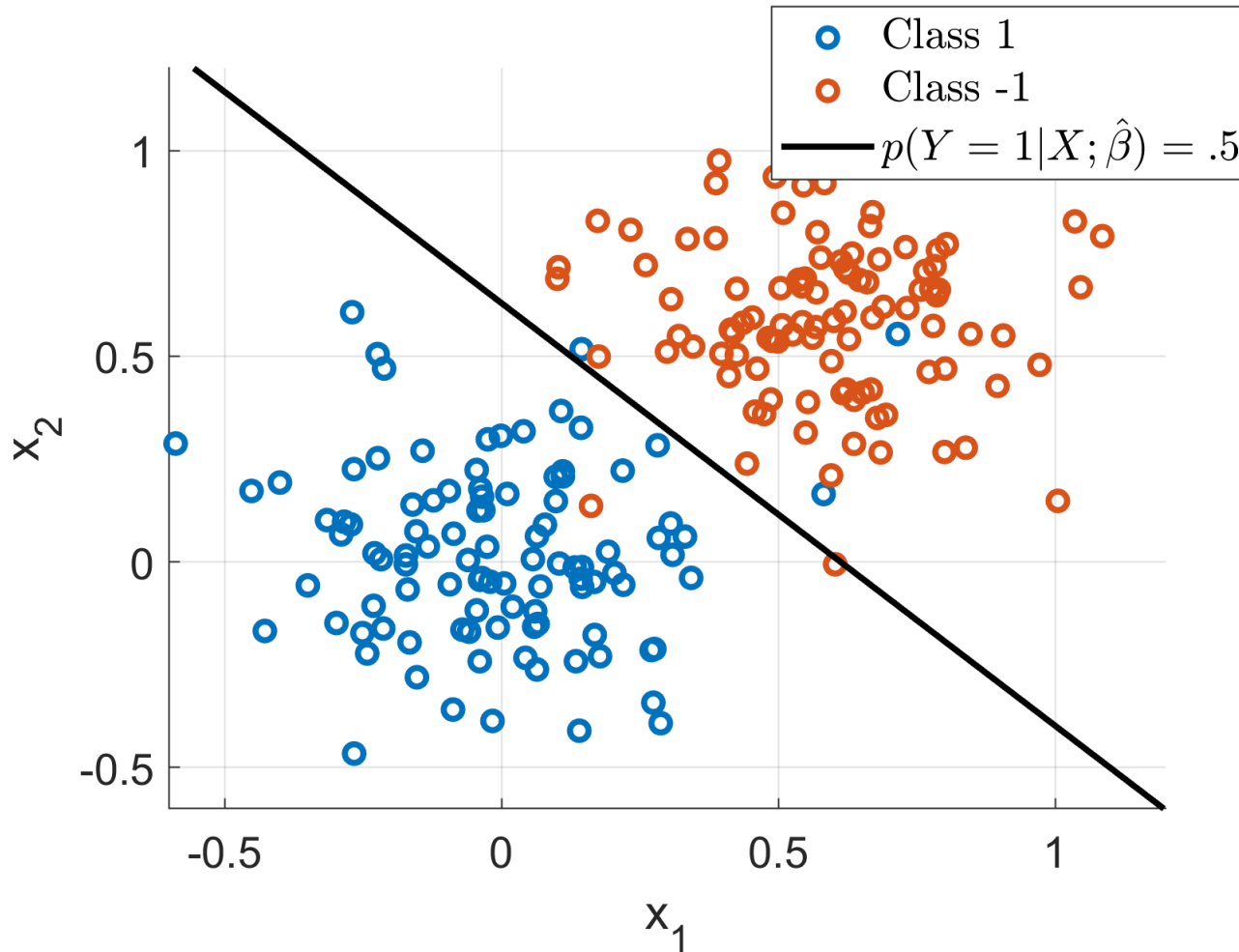
- $N(\mathbf{x}) = \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0) + \exp(-\langle \boldsymbol{\beta}, \mathbf{x} \rangle - d\beta_0)$

- $\boldsymbol{\beta}, \beta_0$  can be fitted using MLE.

- $\hat{\boldsymbol{\beta}}, \hat{\beta}_0 = \arg \max_{\boldsymbol{\beta}, \beta_0} \sum_{i=1}^n \log p(y_i|\mathbf{x}_i; \boldsymbol{\beta}, \beta_0)$

- Homework: Show this is the same Logistic Regression we talked about in Lec 10.

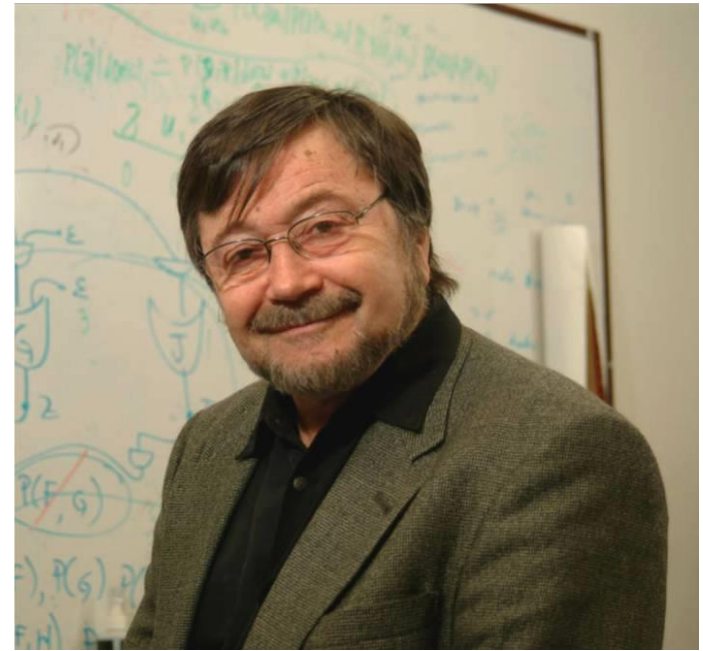
# Example



# 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 prob. dist. factorizes over a classification network.

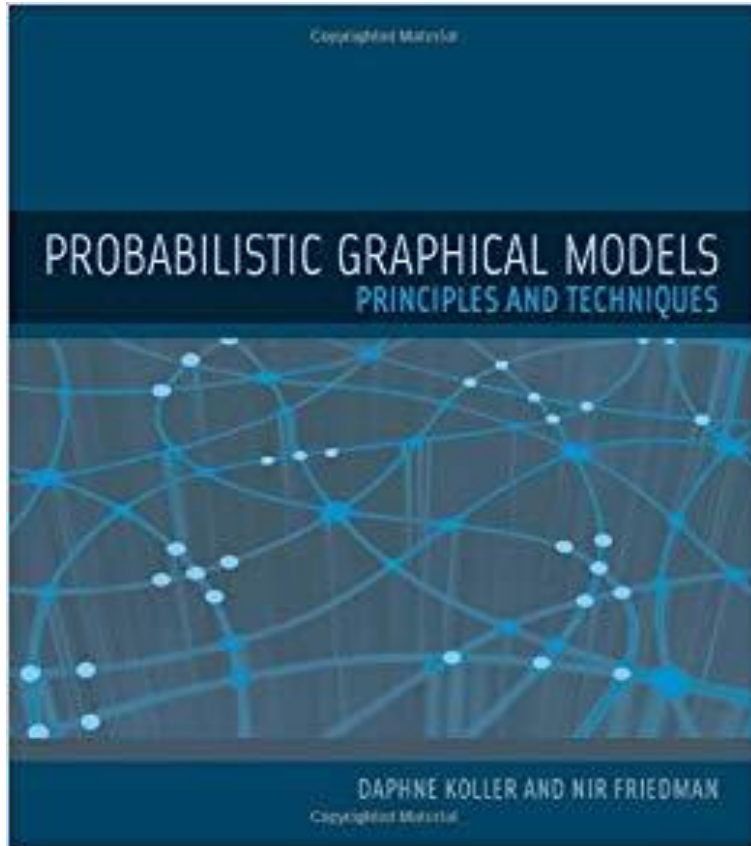




Judea Pearl

# Bayesian Network

# References



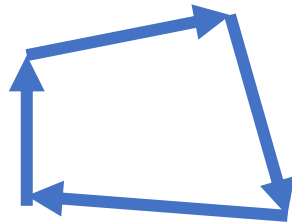
- Today's class roughly follows Chapter 3 in Probabilistic Graphical Models by Koller and Friedman.

# A Directed Graphical Model

- Markov network is an **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 dependencies are better addressed using a directed model.

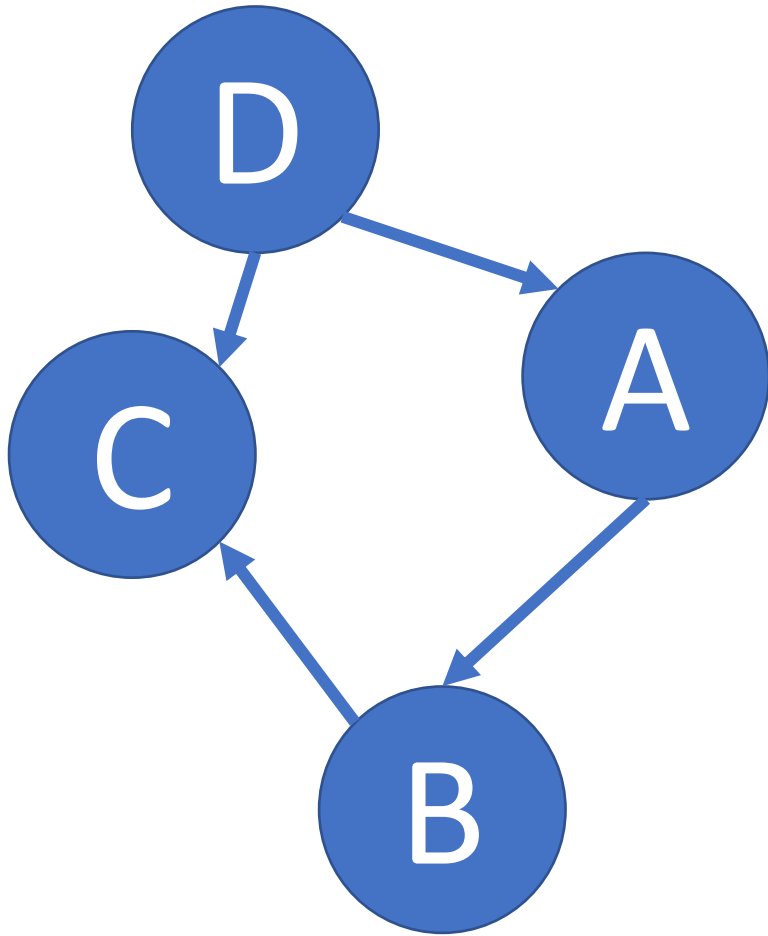
# Directed Acyclic Graph

- 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 a  
**directed edge**  $A \rightarrow B$ :  
 $A$  is the parent of  $B$  and  
 $B$  is the child of  $A$ .

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

Parent(A): D

Children(A): B

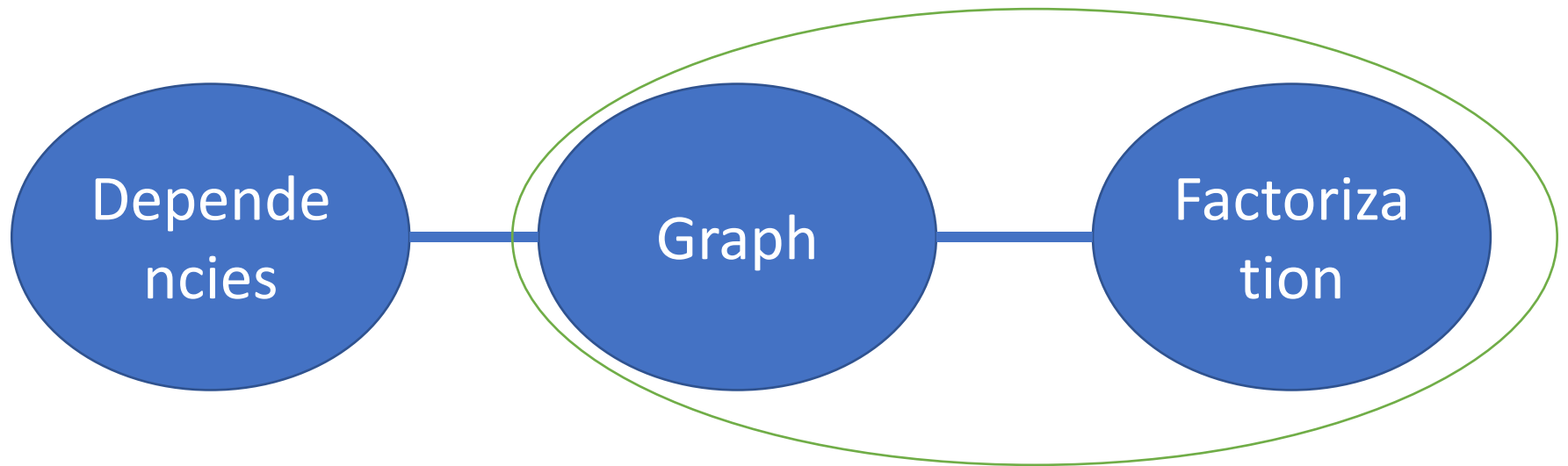
Descendants(A): B, C

# Example



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

# Graph and Factorization



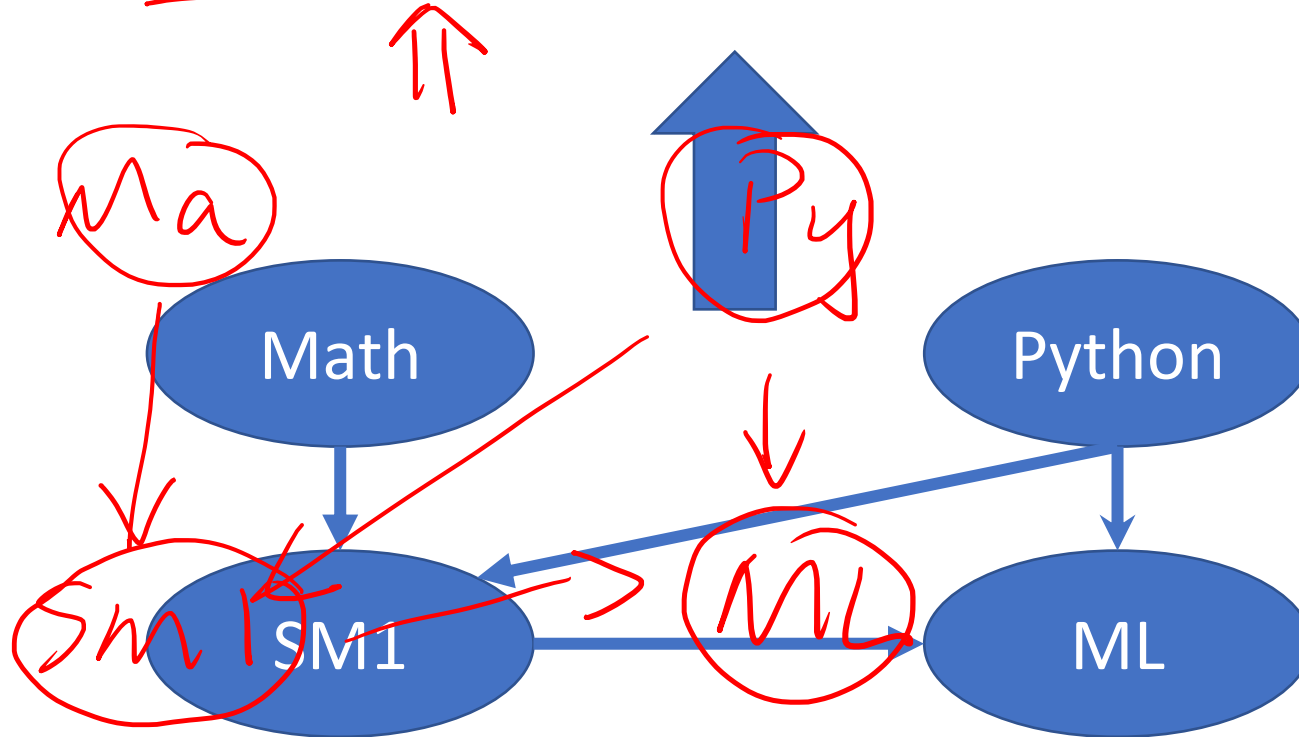
# Representing Factorization using DAG

- 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
- $p(X) = \prod_{v \in V} p(X_v | X_{\text{parent}(X_v)})$

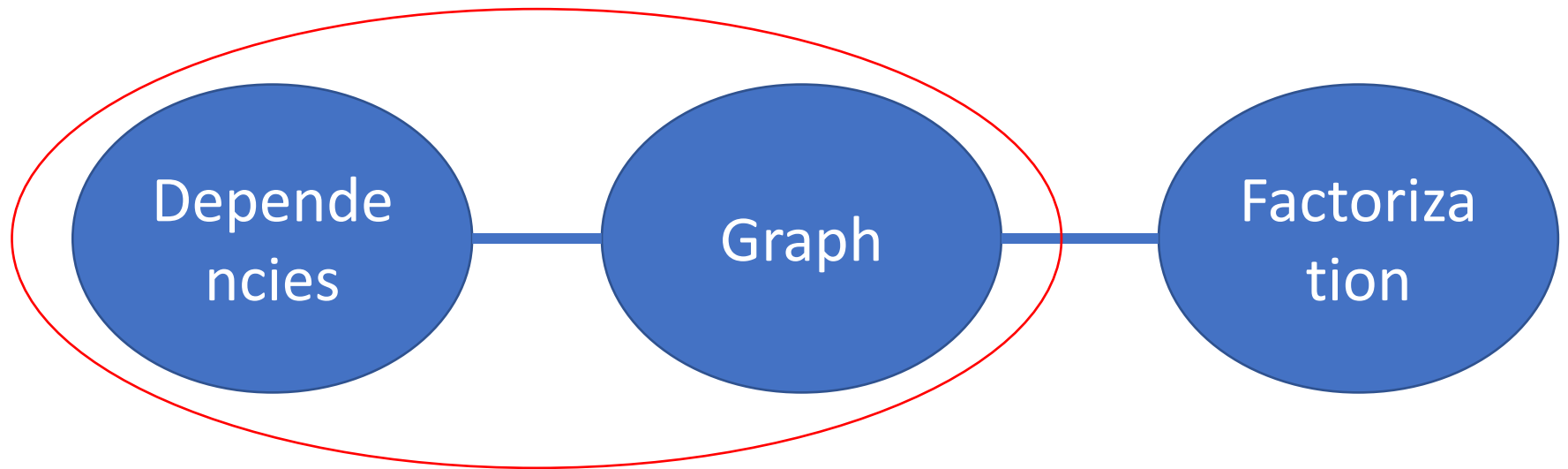


# Example

$$\bullet p(Ma, Py, SM1, ML) = \underbrace{p(\underline{Ma})}_{\uparrow} \underbrace{p(\underline{Py})}_{\uparrow} \underbrace{p(\underline{SM1} | \underline{Ma}, \underline{Py})}_{\uparrow} \underbrace{p(\underline{ML} | \underline{SM1}, \underline{Py})}_{\uparrow}$$



# Graph and Dependency

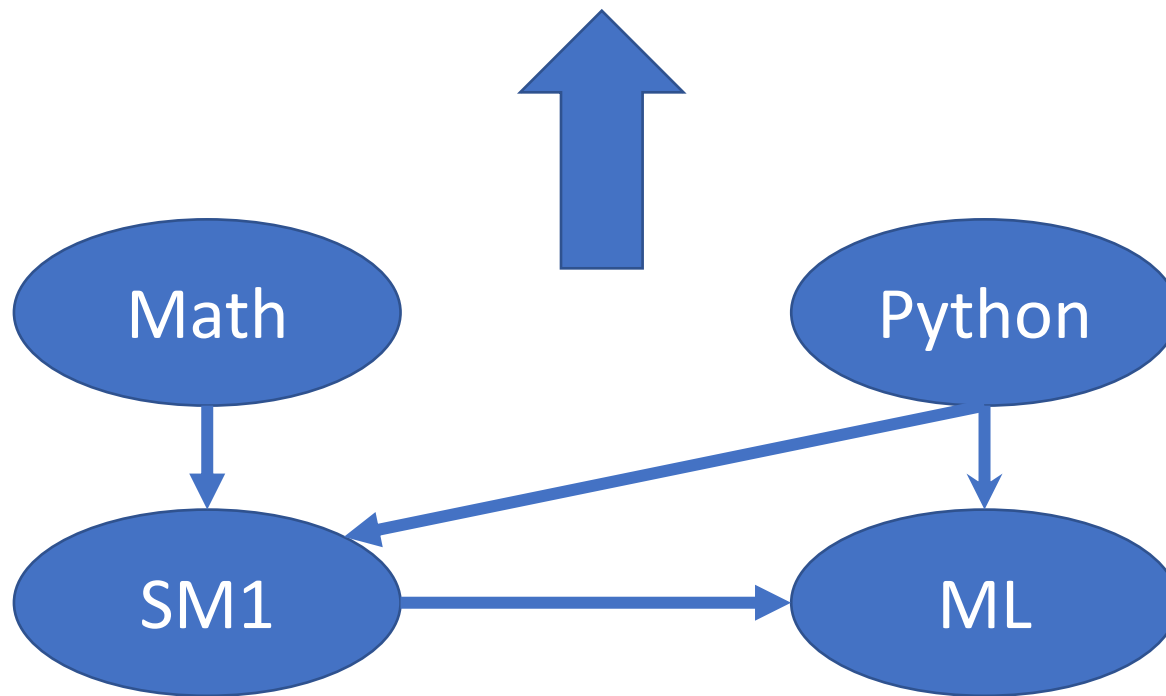


# Represent Cond. Indep. using DAG

- 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  $X_v$ .

# Example

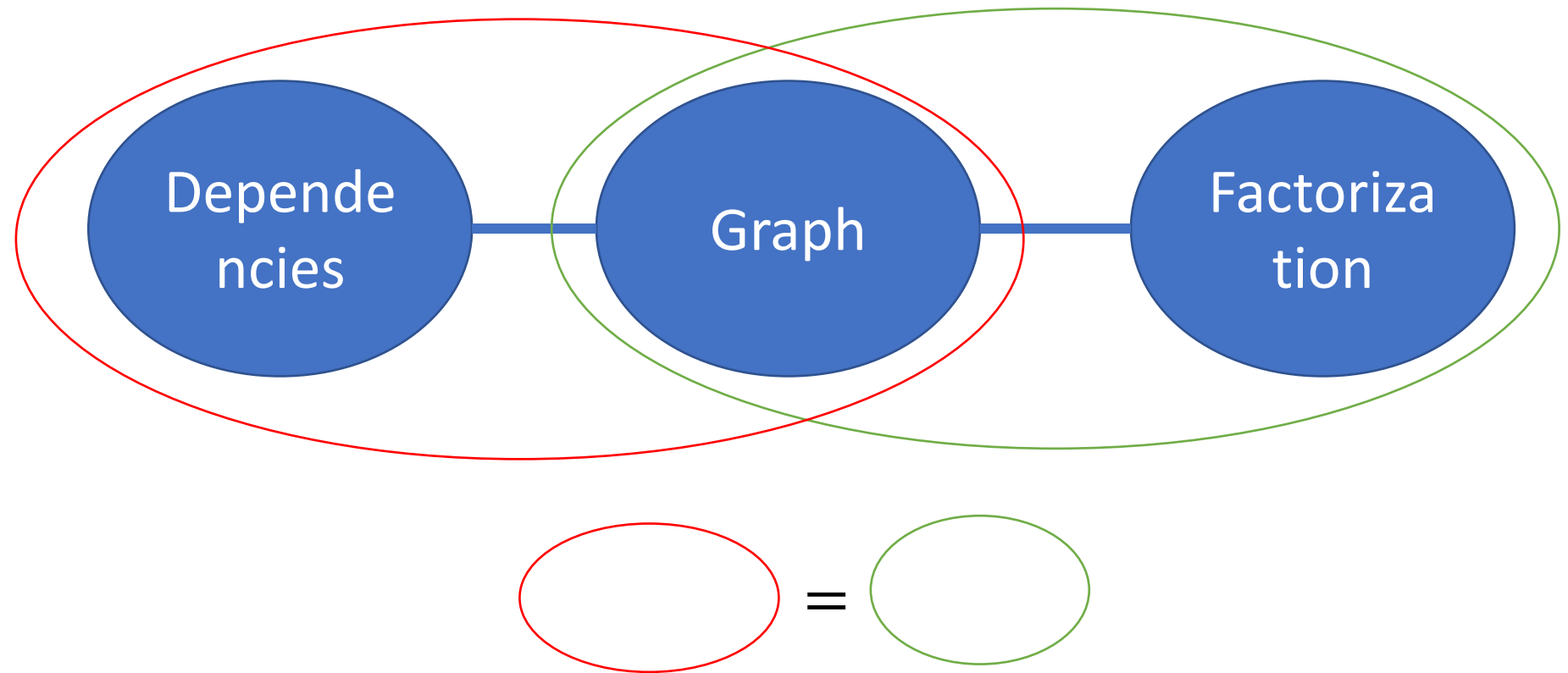
- $ML \perp Math \mid SM1, Python$
- $Math \perp Python$



# 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  $G$ .

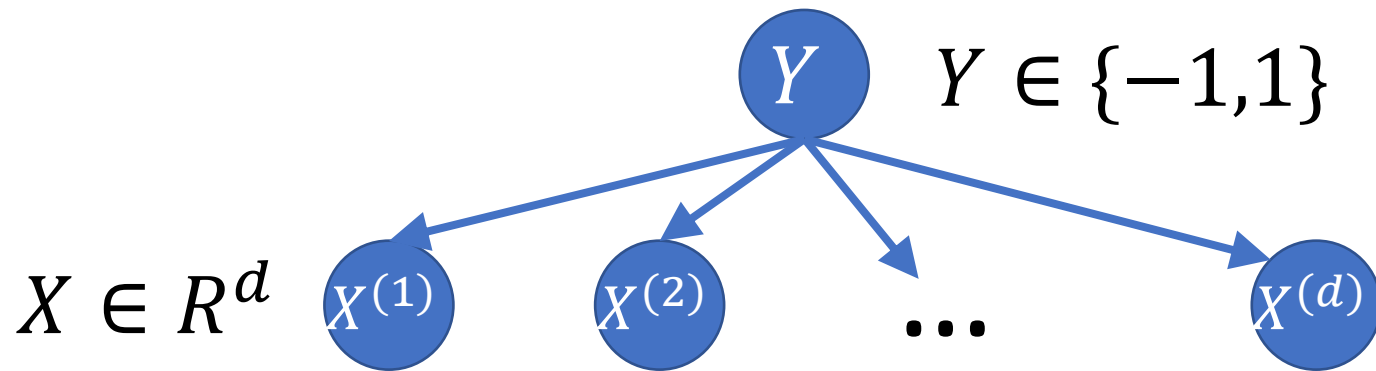
# Dependencies, Graph, Factorization



# Bayesian Network

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

# Bayesian Network for Classification



- Looks familiar?



# Bayesian Network for Classification

- Write down the conditional probability  $P(Y|X)$ .

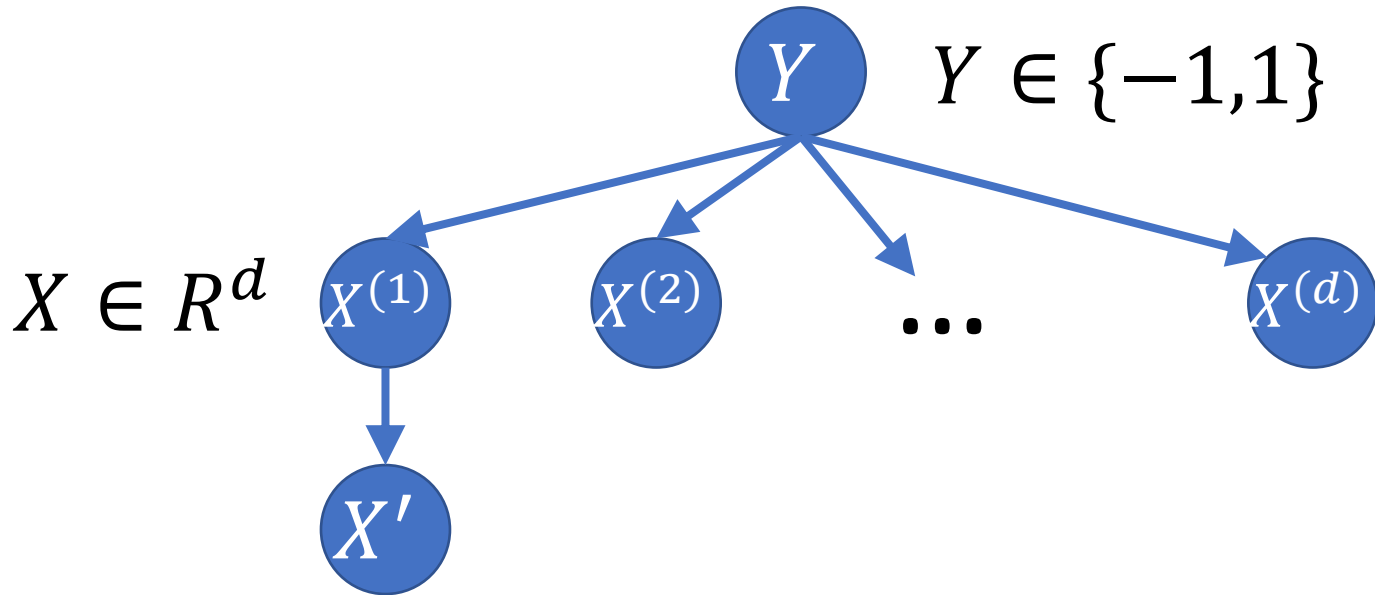
- $$P(Y|X) = \frac{\prod_i P(X^{(i)}|Y)P(Y)}{P(X)}$$

- This is how Naïve Bayes is derived!

# Bayesian Network for Classification

- Compare NB and Logistic regression from the following perspectives:
  - The graphical structure
    - Same structure
    - Directed vs. Undirected
  - The factorization
    - Pairwise factors between  $Y$  and  $X_i$ .
    - **Factor on cliques** vs. **Conditional Prob.**
  - The probabilistic model
    - Both use  $p(Y|X)$  to make prediction
    - NB **does not** give you  $p(Y|X)$ , only up to a constant
  - The training/fitting of a classifier
    - Estimation of  $p(Y|X)$  vs.  $P(X|Y)$
  - Prediction rule
    - Both  $\hat{y} := \operatorname{argmax}_y p(Y|X)$

# Question



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

# Conclusion

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