Lecture 22 -- Graphs and Distributions

Prof Xiaowei Huang
https://cgi.csc.liv.ac.uk/~xiaowei/
(Attendance Code: **488245**)

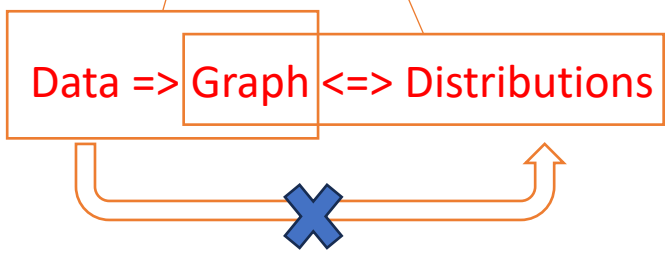
Up to now,

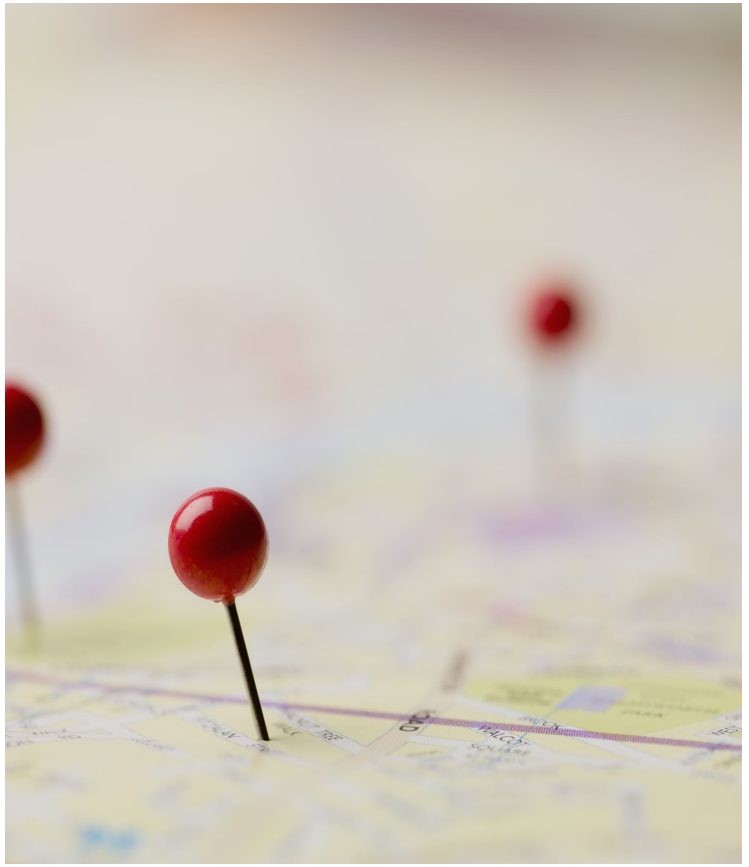
- Overview of Machine Learning
- Traditional Machine Learning Algorithms
- Adversarial Attack and Defence
- Deep learning
- Probabilistic Graphical Models
  - Introduction

Topics

- Markov Assumption and Definition of I-Maps
- From I-Map to Factorization, and back
- Perfect Map

- Learning of Graph Models
- Caution in Establishing a Connection between Two Variables
- Overview of Structure Learning Methods

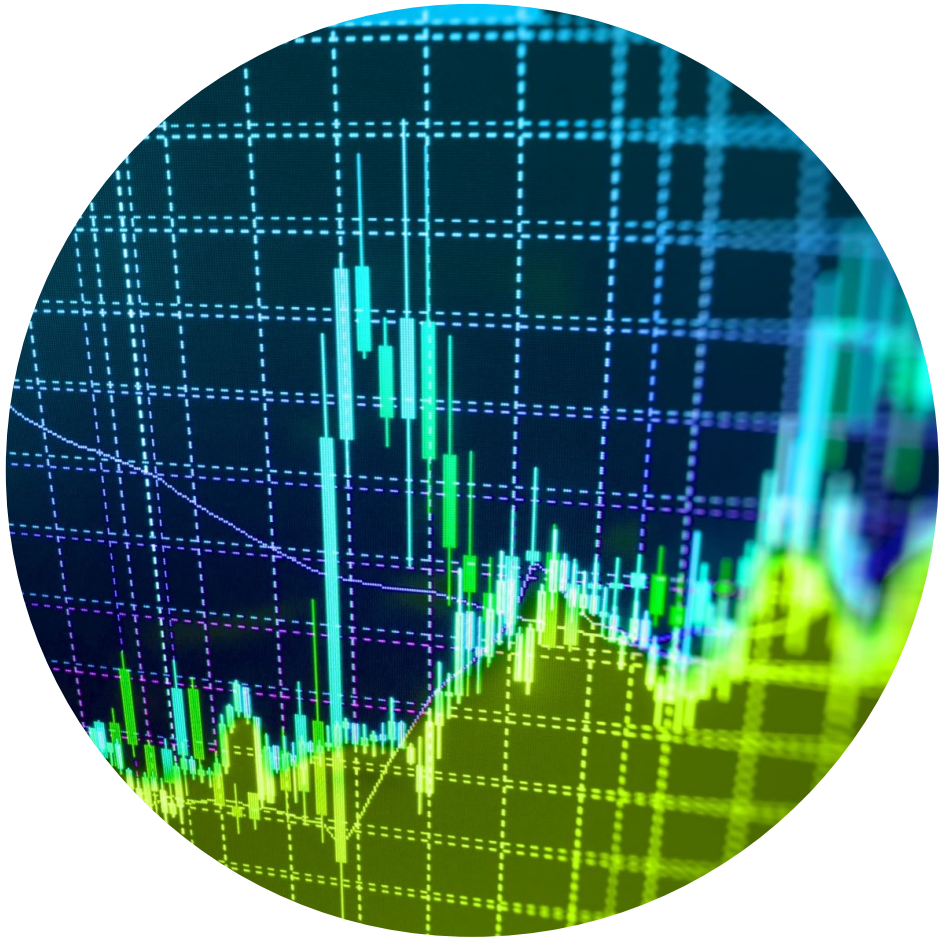




Graphs and Distributions

- Relating two concepts:
  - Conditional Independencies in distributions
  - Conditional Independencies in graphs
- I-Map is a relationship between the two

Graph <== I-Map ==> Distributions



Recap: Conditional Independence

- X, Y independent $X \perp Y$ or $X \perp Y | \emptyset$

  if and only if: $\forall x, y : P(x, y) = P(x)P(y)$

Now, given a joint distribution, we can determine the conditional independence according to the definitions.

- X and Y are conditionally independent given Z: $X \perp Y | Z$

  if and only if:

$$\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$$

Independencies in a Distribution

- Let *P* be a distribution over *X*
- Define $I(P)$ to be the set of conditional independence assertions of the form $(X \perp Y | Z)$ that hold in *P*
- Example:

X	Y	P(X,Y)
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

X and Y are independent in P, e.g.,

$$P(x^1)=0.48+0.12=0.6$$
$$P(y^1)=0.32+0.48=0.8$$
$$P(x^1,y^1)=0.48=0.6\mathbf{x}0.8$$

The same for $P(x^0,y^1)$, $P(x^0,y^0)$, and $P(x^1,y^0)$

Thus $(X \perp Y | \phi) \in I(P)$

Independencies in a Distribution

- Let *P* be a distribution over *X*
- Define $I(P)$ to be the set of conditional independence assertions of the form $(X \perp Y | Z)$ that hold in *P*
- Example:

X	Y	P(X,Y)
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

X and Y are independent in P, e.g.,

$P(x^1)=0.48+0.12=0.6$
$P(y^1)=0.32+0.48=0.8$
$P(x^1,y^1)=0.48=0.6\mathbf{x}0.8$

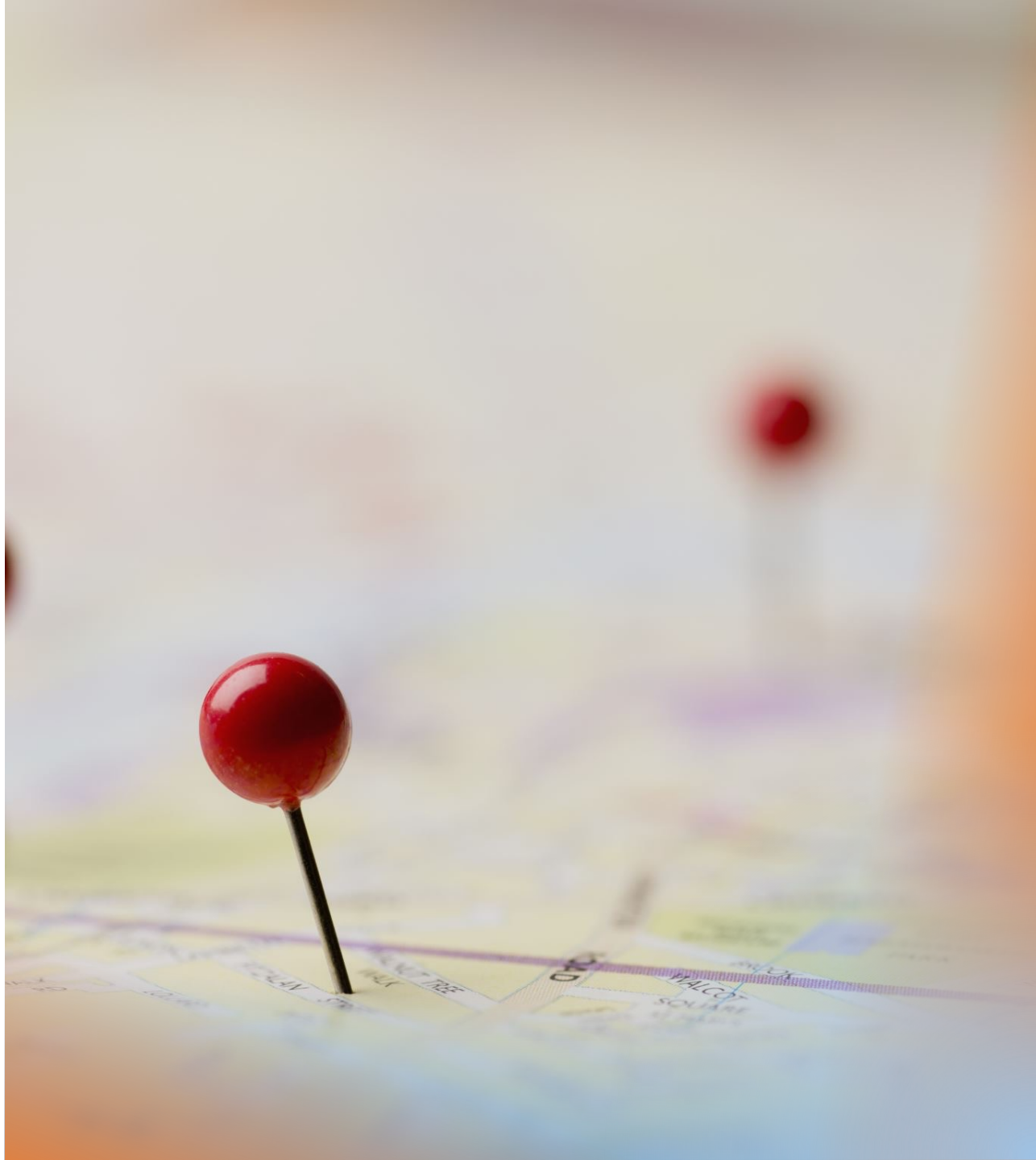
The same for $P(x^0,y^1)$, $P(x^0,y^0)$, and $P(x^1,y^0)$

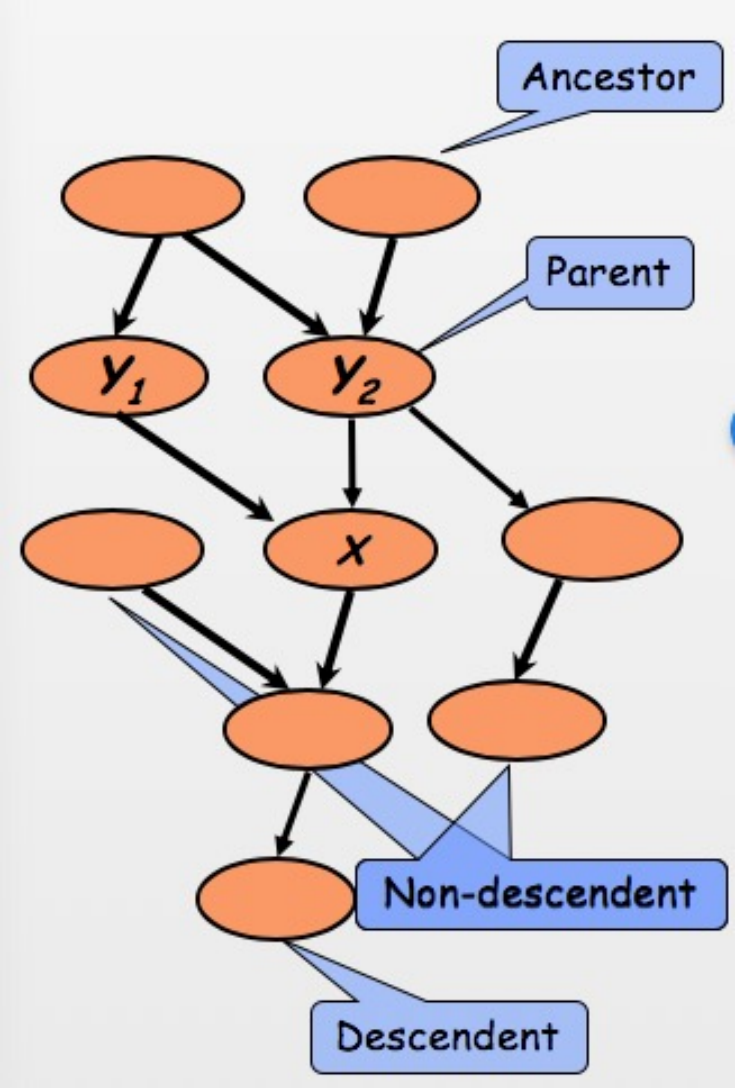
Thus $(X \perp Y|\phi) \in I(P)$

How about this distribution?

X	Y	P(X,Y)
x^0	y^0	0.10
x^0	y^1	0.16
x^1	y^0	0.64
x^1	y^1	0.10

Markov Assumption and Definition of I-Map



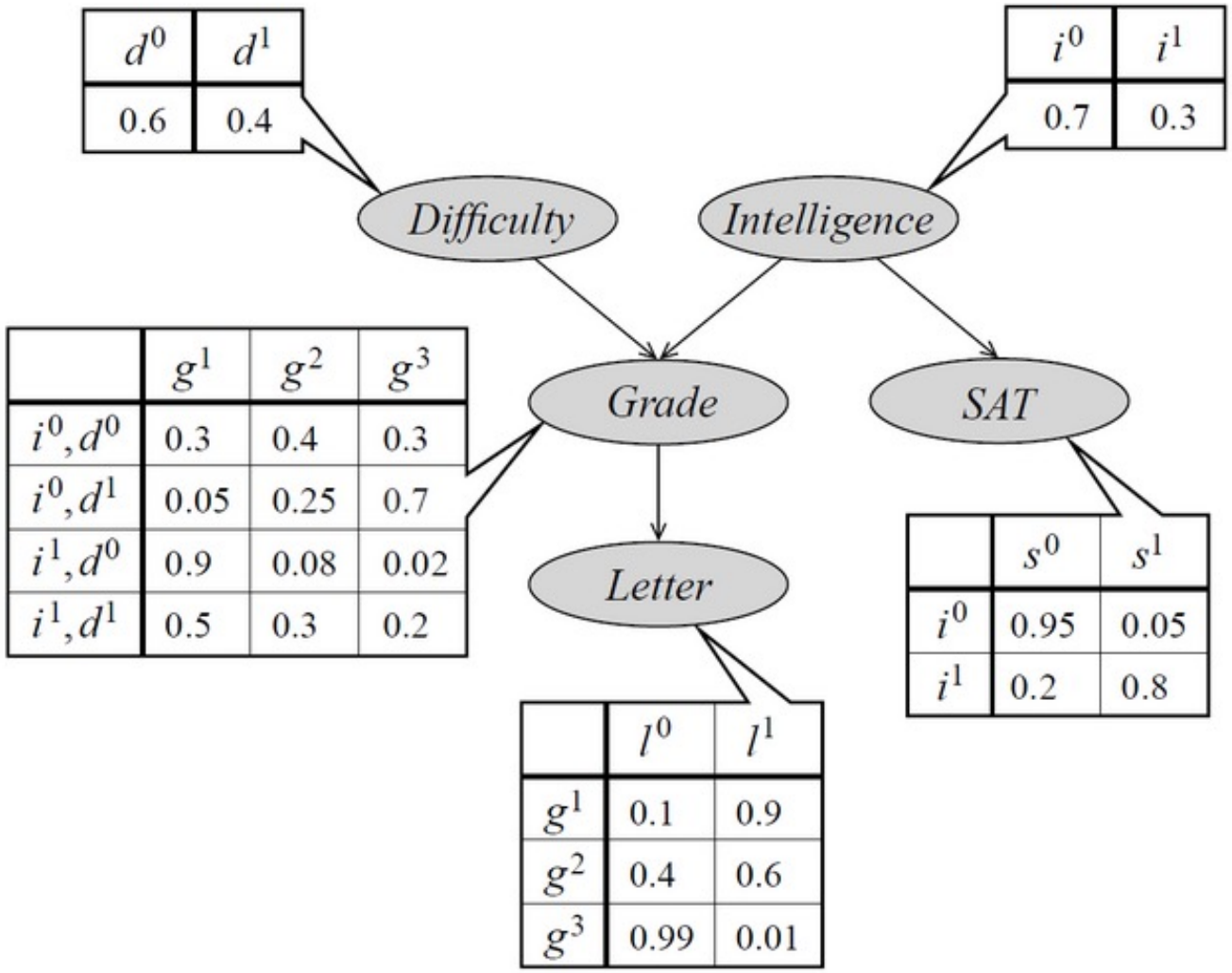


Markov Assumption

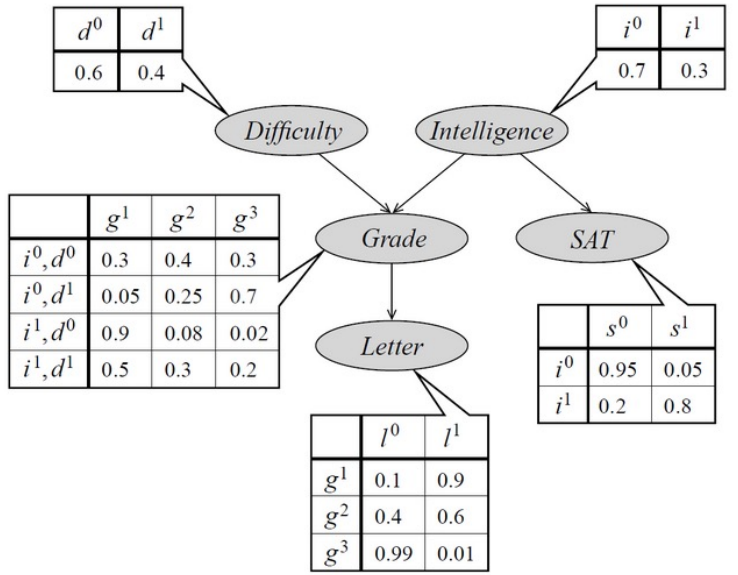
- We now make this independence assumption more precise for **directed acyclic graphs** (DAGs)
- Each random variable X, is independent of its non-descendents, given its parents Pa(X)
- Formally,

$$(X \perp NonDesc(X)|pa(X))$$

Can we read off the independencies from a graph?



Independencies in a Graph



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- Graph G with CPDs is equivalent to a set of independence assertions

$$P(D,I,G,S,L) = P(D)P(I)P(G \mid D,I)P(S \mid I)P(L \mid G)$$

- Local Conditional Independence Assertions (starting from leaf nodes):

$$I_{local}(G) = \{(L \perp I, D, S \mid G), \quad \text{L is conditionally independent of all other nodes given parent G}$$
$$(S \perp D, G, L \mid I), \quad S \text{ is conditionally independent of all other nodes given parent I}$$
$$(G \perp S \mid D, I), \quad \text{Even given parents, G is NOT independent of descendant L}$$
$$(I \perp D \mid \emptyset), \quad \text{Nodes with no parents are marginally independent}$$
$$(D \perp I, S \mid \emptyset)\} \quad \text{D is independent of non-descendants I and S}$$

- Parents of a variable shield it from probabilistic influence
  - Once value of parents known, no influence of ancestors
- Information about descendants can change beliefs about a node

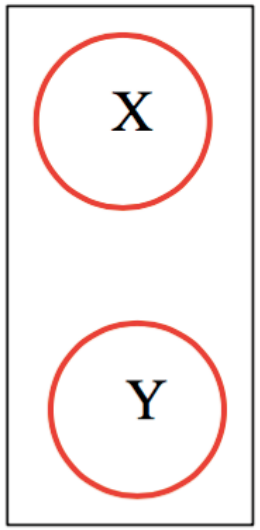
Definition of I-MAP

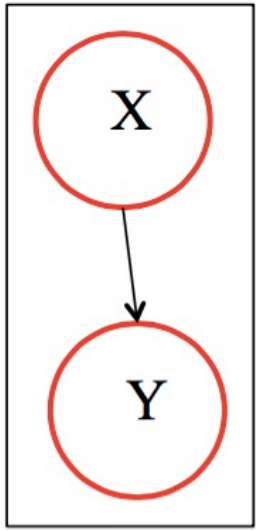
- Let G be a graph associated with a set of independencies I(G)
- Let P be a probability distribution with a set of independencies $I(P)$
- Then G is an **I-Map** of P if $I(G) \subseteq I(P)$
- From direction of inclusion
  - Distribution can have more independencies than the graph
  - Graph does not mislead in independencies existing in P
    - Any independence that G asserts must also hold in P

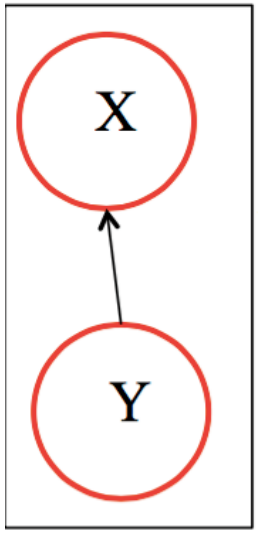
- I_{local}(G), which includes all local Markov assumptions, is a subset of I(G), which includes not only local Markov assumptions but also global conditional independence.
- $I_{local}(G) \subseteq I(G)$

to be introduced as d-separation

Example of I-MAP







G_0 encodes $X \perp Y$ or

$I(G_0) = \{X \perp Y\}$

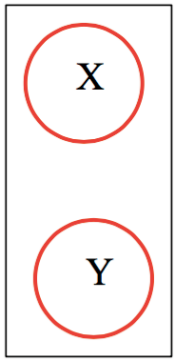
G_1 encodes no Independence, or

$I(G_1) = \emptyset$

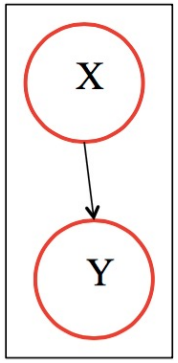
G_2 encodes no Independence, or

$I(G_2) = \emptyset$

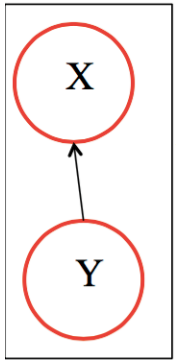
Example of I-MAP



G_0 encodes $X \perp Y$ or $I(G_0) = \{X \perp Y\}$



G_1 encodes no Independence, or $I(G_1) = \emptyset$



G_2 encodes no Independence, or $I(G_2) = \emptyset$

X	Y	P(X,Y)
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

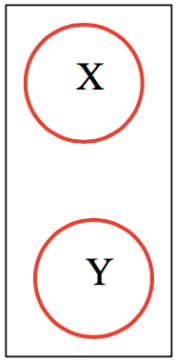
X and *Y* are independent in *P*, e.g.,

$I(P) = \{X \perp Y\}$
which suggests the following:

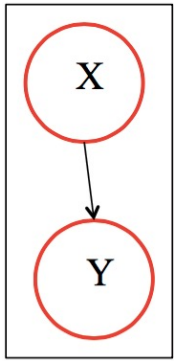
G_0 is an I-map of *P* — $I(G_0) \subseteq I(P)$
G_1 is an I-map of *P* — $I(G_1) \subseteq I(P)$
G_2 is an I-map of *P* — $I(G_2) \subseteq I(P)$

If G is an I-map of *P* then it captures some of the independences, not all

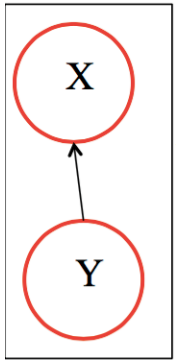
Example of I-MAP



G_0 encodes $X \perp Y$ or $I(G_0) = \{X \perp Y\}$



G_1 encodes no Independence, or $I(G_1) = \emptyset$



G_2 encodes no Independence, or $I(G_2) = \emptyset$

X	Y	P(X,Y)
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1

X and Y are not independent in P

Thus $\boxed{(X \perp Y) \notin I(P)}$

G_0 is not an I-map of P

G_1 is an I-map of P

G_2 is an I-map of P

$I(P) = \emptyset$

which suggests the following:

$I(G_0) \not\subseteq I(P)$

$I(G_1) \subseteq I(P)$

$I(G_2) \subseteq I(P)$

If G is an I-map of *P* then it captures some of the independences, not all

Exercise

- Please draw an I-Map for each of the following distributions:

x	y	P(x,y)
0	0	0.25
0	1	0.25
1	0	0.25
1	1	0.25

x	y	P(x,y)
0	0	0.2
0	1	0.3
1	0	0.4
1	1	0.1

From I-map to Factorization, and back

What is factorization?

- **factorization** or **factoring** consists of writing a number or another mathematical object as a product of several *factors*, usually smaller or simpler objects of the same kind

- In our context, for example:

$$P(D,I,G,S,L) = P(D)P(I)P(G \mid D,I)P(S \mid I)P(L \mid G)$$

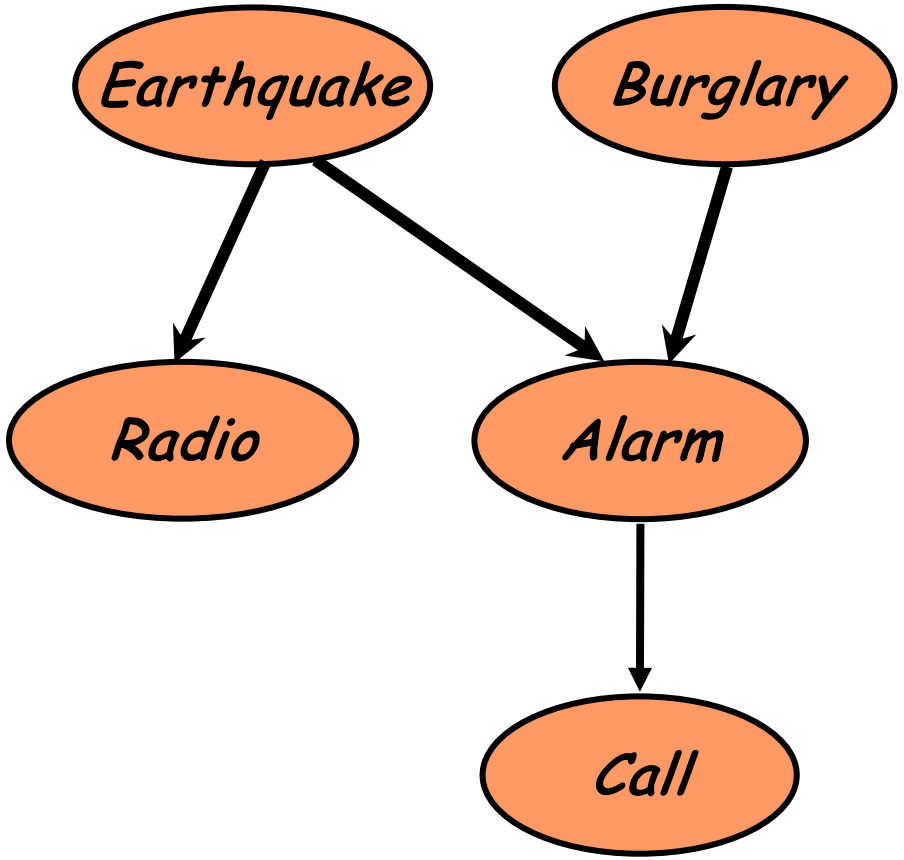
or

$$P(I,D,G,L,S) = P(I)P(D|I)P(G|I,D)P(L|I,D,G)P(S|I,D,G,L)$$

I-map to Factorization

- **Thm**: if G is an I-Map of P, then

$$P(X_1,\ldots,X_n) = \prod_i P(X_i \mid Pa(X_i))$$



Exercise

- Please give the factorization of the distribution P according to the I-Map shown in the figure.

Factorization to I-map

Thm

$$P(X_1,\ldots,X_n) \square \prod_i P(X_i \mid Pa_i) \quad \Rightarrow \textbf{G} \text{ is an I-Map of } P$$

Perfect Map

Perfect Map

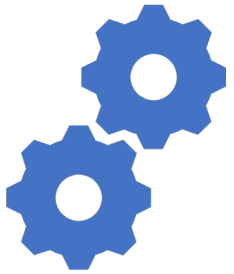
- I-map
  - All independencies in *I(*G*)* present in *I(P)*
  - Trivial case: all nodes interconnected
- D-Map
  - All independencies in *I(P)* present in *I(*G*)*
  - Trivial case: all nodes disconnected
- Perfect map
  - Both an I-map and a *D*-map
  - Interestingly not all distributions P over a given set of variables can be represented as a perfect map

Learning of Graph Models

Two approaches to task of acquiring a model

1. Knowledge Engineering

Construct a network by hand with expert's help



2. Machine Learning

Learn model from a set of instances

Knowledge Engineering vs ML

Knowledge Engineering Approach

- Pick variables, pick structure, pick probabilities
- Too much Effort
  - Simple ones require hours of effort, complex one: months
- Significant testing of model by evaluating results of typical queries yield plausible answers

Machine Learning Approach

- Instances available from distribution we wish to model
- Easier to get large data sets rather than human expertise

Difficulties with Manual Construction

- In some domains:
  - Amount of knowledge required too large
  - No experts who have sufficient understanding
  - Cost: expert time is valuable
- Properties of distribution change from one site to another
- Change over time
  - Expert cannot redesign every few weeks
- Modeling mistakes have serious impact on quality of answers

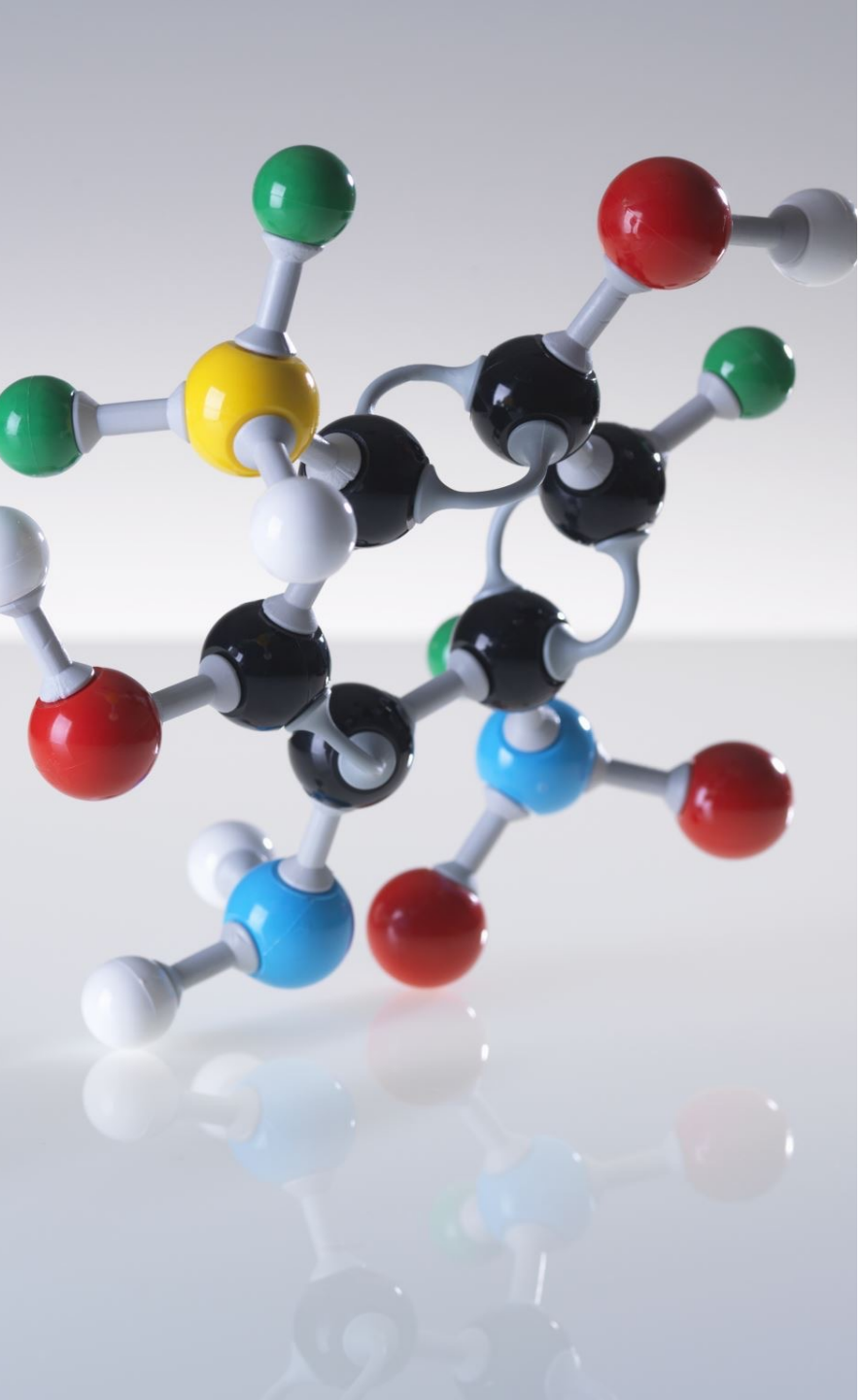
Advantage of ML approach

- We are in the Information Age
  - Easier to obtain even large amounts of data in electronic form than to obtain human expertise
- Example Data
  - Medical Diagnosis
    - Patient records
  - Pedigree Analysis (Genetic Inheritance)
    - Family trees for disease transmission
  - Image Segmentation
    - Set of images segmented by a person
- Machine Learning Algorithms

to work out the (conditional) probability tables

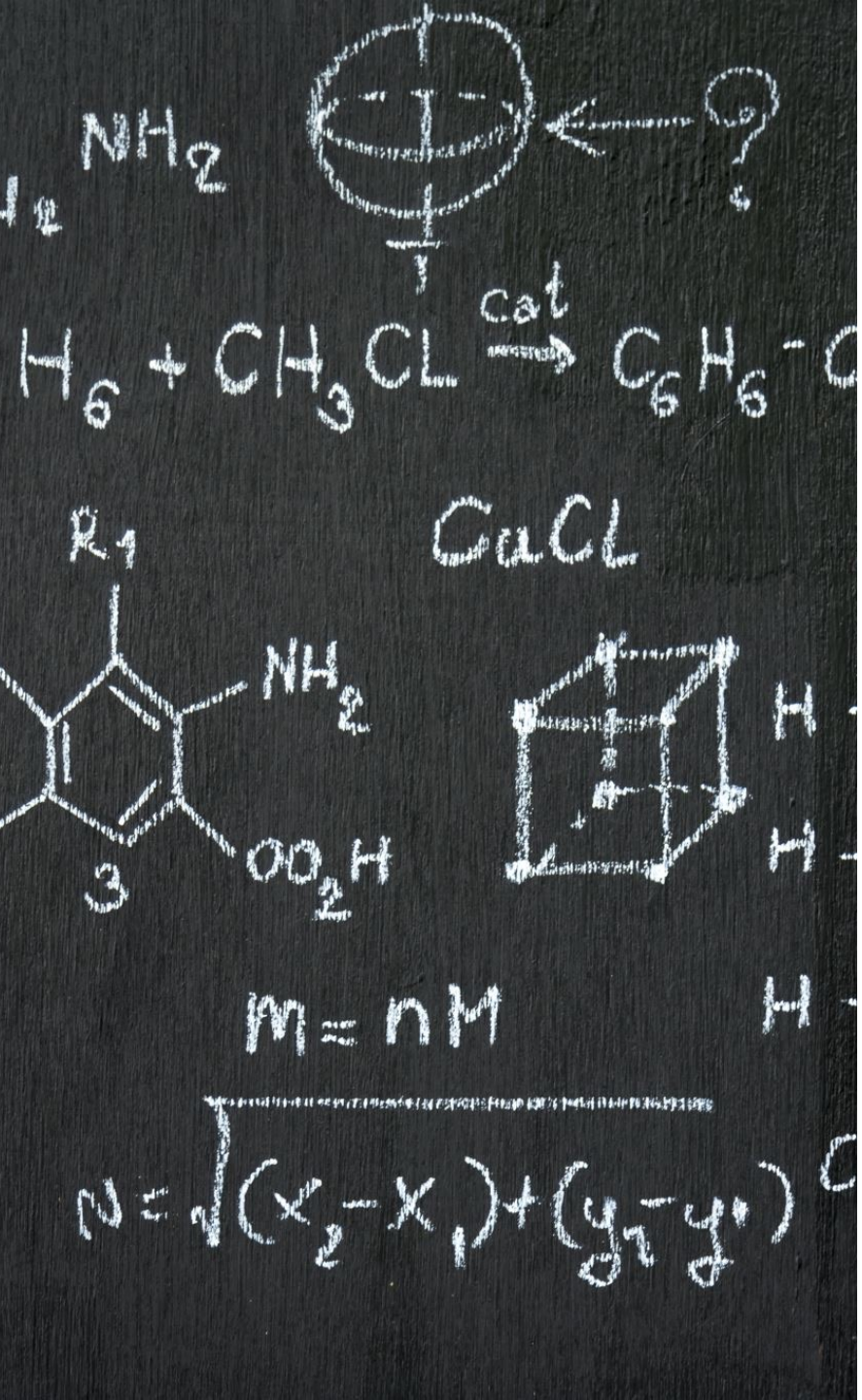
Example: Medical Diagnosis Task

- Collection of patient records
  - History:
    - Age, sex, history, medical complications
  - Symptoms
  - Results of tests
  - Diagnosis
  - Treatment
  - Outcome
- Task: Use data to model distribution of patients
    - Pathologist diagnoses disease of lymph nodes (Pathfinder 1992)



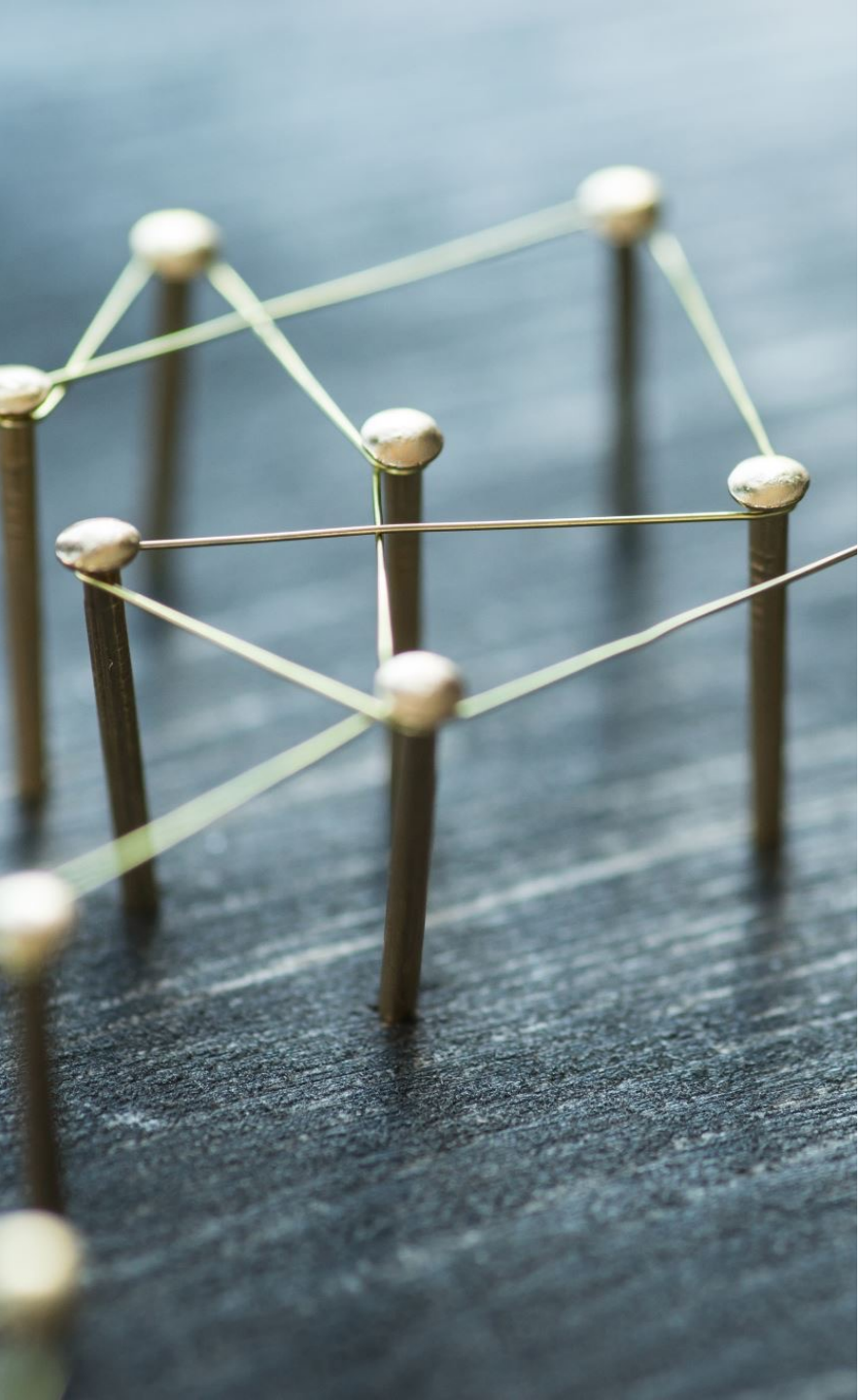
Goal of Structure Learning: Knowledge Discovery

- A tool for discovering knowledge about *P**
  - What are the direct/indirect independencies?
  - Nature of dependencies
    - E.g., positive or negative correlation
  - Example: in medical domain, which factors lead to a disease
- Bayesian network reveals much finer structure
  - Distinguish between direct and indirect independencies, both of which lead to correlations

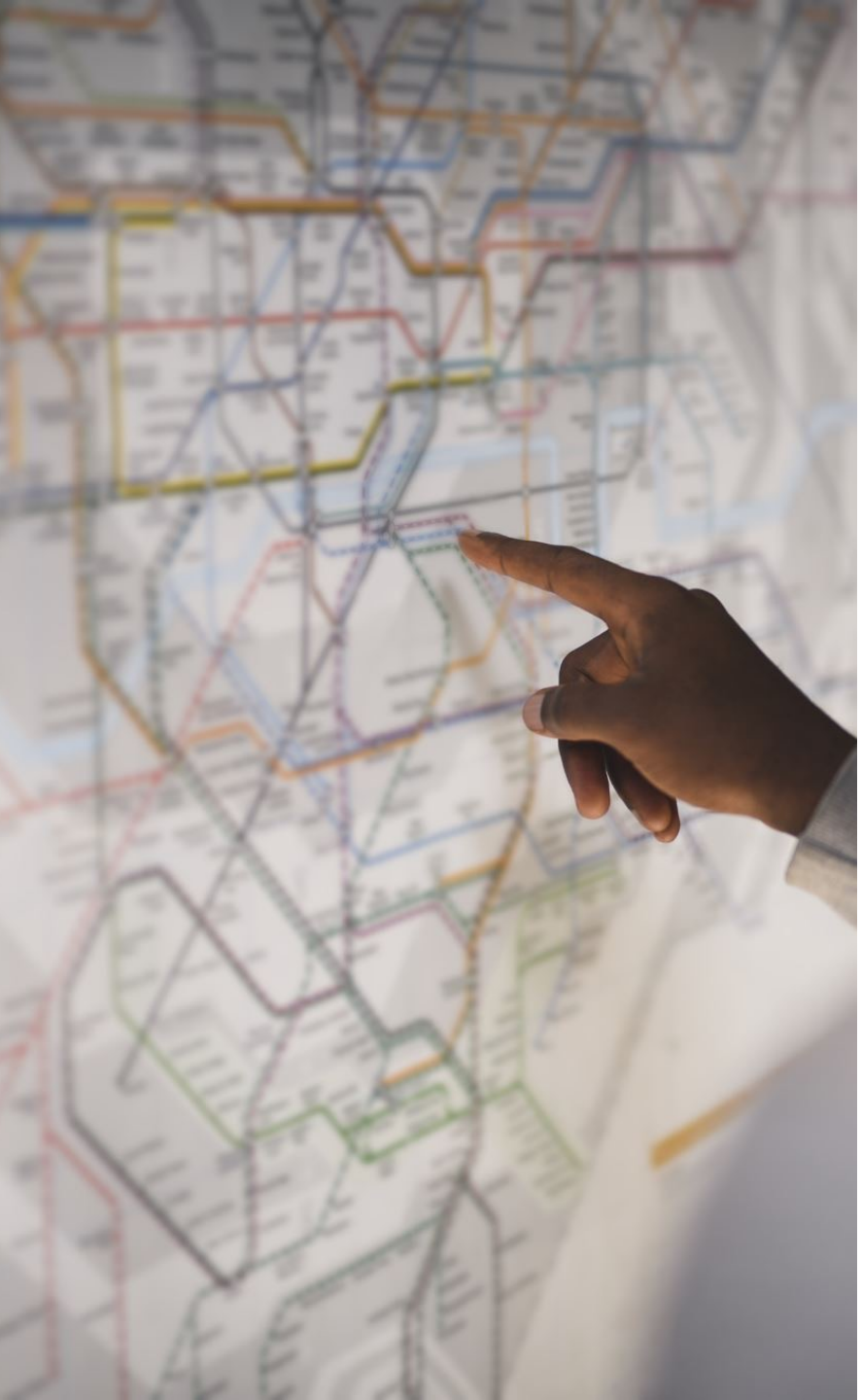


Problem Assumptions

- We do not know the structure
- Dataset is fully observed
  - A strong assumption
- Assume data *D* is generated i.i.d. from distribution *P*(X)*
- Assume that *P** is induced by BN *G**

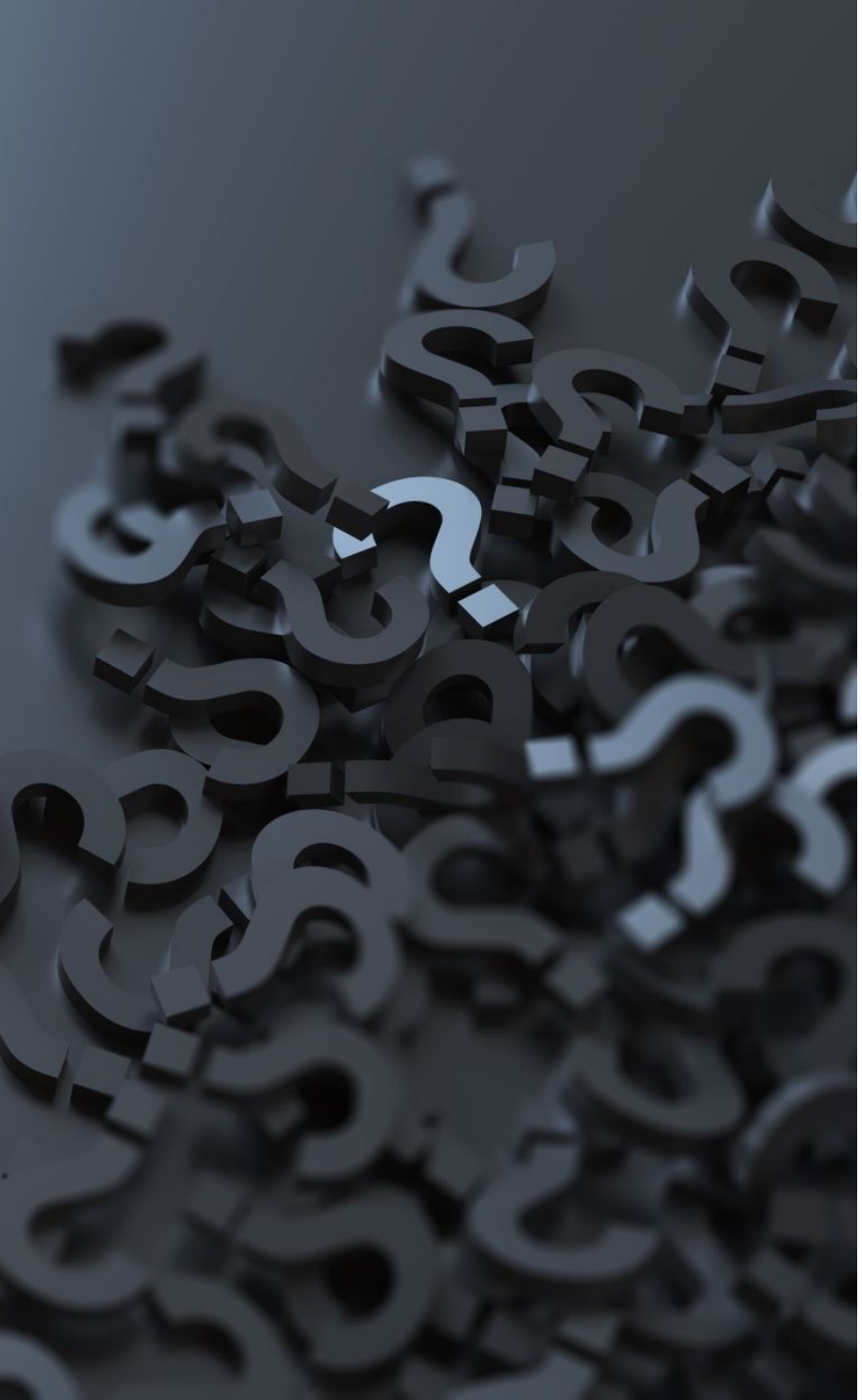


Caution in Establishing a Connection between Two Variables



Knowledge Discovery Goal

- Goal: recover *G**
- Since there are many I-maps for P* we cannot distinguish them from *D*
- Thus *G** is not *identifiable*
- Best we can do is recover *G**s equivalence class



Too few or too many edges in *G**

- Even learning equivalence class of networks is hard
- Data sampled is noisy
- Need to make decisions about including edges we are less sure about
  - Too few edges means missing out on dependencies
  - Too many edges means spurious dependencies

To what extent do independencies in *G** manifest in *D*?

- Two coins X and Y tossed independently
- We are given data set of 100 instances
- Learn a model for this scenario
- Typical data set:
  - 27 head/head
  - 22 head/tail
  - 25 tail/head
  - 26 tail/tail
- Are the coins independent?

X

Y

Coin Tossing Probabilities

- Marginal Probabilities
  - *P(X=head)=.49, P(X=tail)=0.51, P(Y=head)=.52, P(Y=tail)=.48*
- Products of marginals*:*
  - *P(X=head)* x *P(Y=head) =.49 x .52 =.25*
  - *P(X=head)* x *P(Y=tail) =.49 x .48 =.24*
  - *P(X=tail)* x *P(Y=head) =.51 x .52 =.27*
  - *P(X=tail)* x *P(Y=tail) =.51 x .48 =.24*
- Joint Probabilities
  - *P(XY=head-head)=.27*
  - *P(XY=head-tail)=22*
  - *P(XY=tail-head)=.25*
  - *P(XY=tail-tail)=.26*

27 head/head
22 head/tail
25 tail/head
26 tail/tail

According to empirical distribution: not independent

X

Y

- But we suspect independence
  - since probability of getting exactly 25 in each category is small (approx. 1 in 1,000)

Rain-Soccer Probabilities

- Scan sports pages for 100 days
- Select an article at random to see
  - If there is mention of rain and soccer
- Marginal Probabilities
  - *P(X=rain)=.49, P(X=no rain)=.51, P(Y=soccer)=.48, P(Y=no soccer)=. 52*
- Joint Probabilities
  - *P(XY=rain-soccer)=.27*
  - *P(XY=rain-no soccer)=.22*
  - *P(XY=no rain-soccer)=.25*
  - *P(XY=no rain-no soccer)=.26*

Rain washes away girls soccer district openers

Area teams wait until today for district tourneys

12:57 AM, Jan. 14, 2014

The No. 2-seeded Navarre Raiders will have to wait another day for their shot at Crestview as rain and lightning postponed the start of the District 1-4A girls soccer tournament at the Santa Rosa Soccer

According to empirical distribution: not independent

We suspect there is a weak connection (not independent)

It is hard to be sure whether the true underlying model has an edge between X and Y

Data Fragmentation with spurious edges

- In a table CPD number of bins grows exponentially with number of parents
- Cost of adding a parent can be very large
- Cost of adding a parent grows with no of parents already there
- It is better to obtain a sparser structure
- We can sometimes learn a better model by learning a model with fewer edges even if it does not represent the true distribution.

Overview of Structure Learning methods

Structure Learning Algorithms

- Constraint-based
  - Find structure that best explains determined dependencies
  - Sensitive to errors in testing individual dependencies
- Score-based
  - Search the space of networks to find high-scoring structure
  - Since space is super-exponential, need heuristics

Finds a Bayesian network structure whose implied independence constraints "match" those found in the data.

Find the Bayesian network structure that can represent distributions that "match" the data (i.e. could have generated the data).

Elements of BN Structure Learning

Local: Independence Tests

- Measures of *Deviance*-from-independence between variables
- Rule for accepting/rejecting hypothesis of independence

Global: Structure Scoring

- Goodness of Network

Independence Tests

Idea: measure the independence between two variables with a value, and then determine whether to establish the connection based on the score.

- For variables x_i, x_j in data set D of M samples
  - Pearson's Chi-squared (χ^2) statistic

$$d_{\chi^2}(\mathcal{D}) = \sum_{x_i, x_j} \frac{\left(M[x_i, x_j] - M \cdot \hat{P}(x_i) \cdot \hat{P}(x_j)\right)^2}{M \cdot \hat{P}(x_i) \cdot \hat{P}(x_j)}$$

    - Independence $d_\chi(D)=0$, larger value when Joint $M[x,y]$ and expected counts (under independence assumption) differ
  - Mutual Information (K-L divergence) between joint and product of marginals

$$d_I(\mathcal{D}) = \frac{1}{M} \sum_{x_i, x_j} M[x_i, x_j] \log \frac{M[x_i, x_j]}{M[x_i] M[x_j]}$$

    - Independence $d_I(D)=0$, otherwise a positive value
- Decision rule
  - False rejection probability due to choice of t is its p-value

$$R_{d,t}(\mathcal{D}) = \begin{cases} \text{Accept } d(\mathcal{D}) \le t \\ \text{Reject } d(\mathcal{D}) > t \end{cases}$$

Structure Scoring

Idea: measure each graph with a score, and select the graph with best score as G

- Log-likelihood Score for *G* with *n* variables

$$score_L(\mathcal{G} : \mathcal{D}) = \sum_{\mathcal{D}} \sum_{i=1}^{n} \log \hat{P}(x_i \mid pax_i)$$ Sum over all data and variables x_i

- 2. Bayesian Score

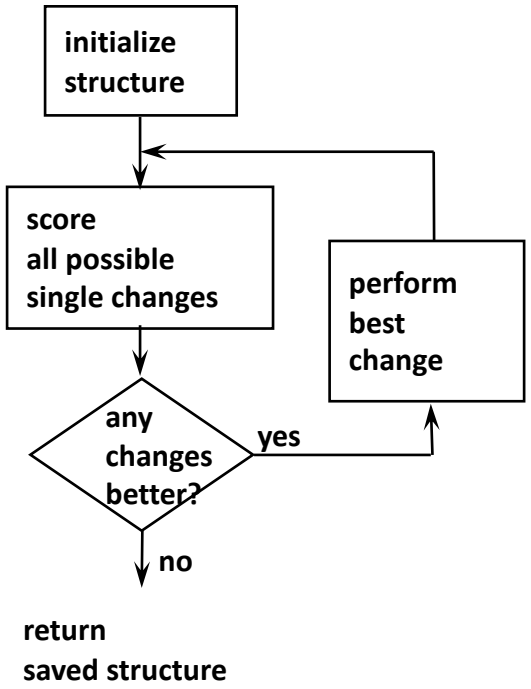
$$score_B(\mathcal{G} : \mathcal{D}) = \log p(\mathcal{D} \mid \mathcal{G}) + \log p(\mathcal{G})$$

- 3. Bayes Information Criterion
  - With Dirichlet prior over graphs

$$score_{BIC}(\mathcal{G} : D) = l(\hat{\theta}_G : D) - \frac{\log M}{2} Dim(\mathcal{G})$$

Model search

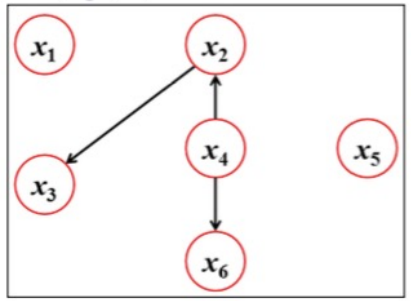
- Finding the BN structure with the highest score among those structures with at most *k* parents is NP hard for *k*>1 (Chickering, 1995)
- Heuristic methods
  - Greedy
  - Greedy with restarts
  - MCMC methods



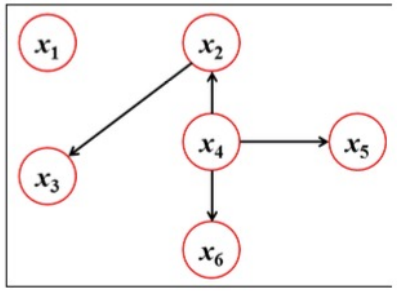
Heuristic for BN Structure Learning

- Consider pairs of variables ordered by χ^2 value
- Add next edge if score is increased

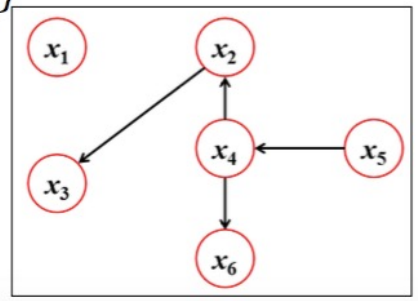
$G^*=\{V,E^*,\theta^*\}$
Start
Score s_{k-1}



$G_{c1}=\{V,E_{c1},\theta_{c1}\}$
Candidate
$x_4 \rightarrow x_5$
Score s_{c1}



$G_{c1}=\{V,E_{c1},\theta_{c1}\}$
Candidate
$x_5 \rightarrow x_4$
Score s_{c2}



Choose G_{c1} or G_{c2}
depending on which one
increases the score $s(\mathcal{D},\mathcal{G})$
evaluate using cross validation
on validation set