

## CS 2750 Machine Learning Lecture 16

### Bayesian belief networks

Milos Hauskrecht  
[milos@cs.pitt.edu](mailto:milos@cs.pitt.edu)  
5329 Sennott Square

---

CS 2750 Machine Learning

### Project proposals

**Due: Wednesday, March 23, 2010**

- **1 page long**

**Proposal**

- **Written proposal:**
  1. Outline of a learning problem, type of data you have available. Why is the problem important?
  2. Learning methods you plan to try and implement for the problem. References to previous work.
  3. How do you plan to test, compare learning approaches
  4. Schedule of work (approximate timeline of work)

---

CS 2750 Machine Learning

## Project proposals

### Where to find the data:

- From your research
- UC Irvine data repository
- Various text document repositories
- I have some bioinformatics data I can share but other data can be found on the NIH or various university web sites (e.g. microarray data, proteomic data)
- Synthetic data that are generated to demonstrate your algorithm works

## Project proposals

### Problems to address:

- Get the ideas for the project by browsing the web
- It is tempting to go with simple classification but definitely try to add some complexity to the investigations
- Multiple, not just one method, try some more advanced methods, say those that combine multiple classifiers to learn a model (ensemble methods) or try to modify the existing methods

## Project proposals

### Interesting problems to consider:

- Advanced methods for learning multi-class problems
- Learning the parameters and structure of Bayesian Belief networks
- Clustering of data – how to group examples
- Dimensionality reduction/feature selection – how to deal with a large number of inputs
- Learning how to act – Reinforcement learning
- Anomaly detection – how to identify outliers in data

---

CS 2750 Machine Learning

## Density estimation

**Data:**  $D = \{D_1, D_2, \dots, D_n\}$   
 $D_i = \mathbf{x}_i$  a vector of attribute values

### Attributes:

- modeled by random variables  $\mathbf{X} = \{X_1, X_2, \dots, X_d\}$  with:

- **Continuous values**

- **Discrete values**

E.g. *blood pressure* with numerical values

or *chest pain* with discrete values

[no-pain, mild, moderate, strong]

### Underlying true probability distribution:

$p(\mathbf{X})$

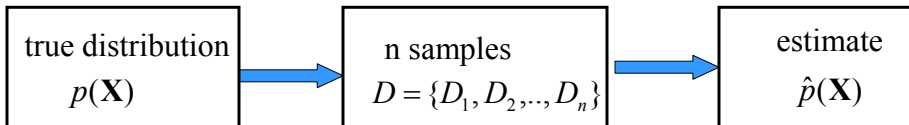
---

CS 2750 Machine Learning

## Density estimation

**Data:**  $D = \{D_1, D_2, \dots, D_n\}$   
 $D_i = \mathbf{x}_i$  a vector of attribute values

**Objective:** try to estimate the underlying true probability distribution over variables  $\mathbf{X}$ ,  $p(\mathbf{X})$ , using examples in  $D$



**Standard (iid) assumptions: Samples**

- are **independent** of each other
- come from the same (**identical**) **distribution** (fixed  $p(\mathbf{X})$ )

## Learning via parameter estimation

In this lecture we consider **parametric density estimation**

**Basic settings:**

- A set of random variables  $\mathbf{X} = \{X_1, X_2, \dots, X_d\}$
- **A model of the distribution** over variables in  $\mathbf{X}$  with parameters  $\Theta$  :

$$\hat{p}(\mathbf{X} | \Theta)$$

- **Data**  $D = \{D_1, D_2, \dots, D_n\}$

**Objective:** find the parameters  $\Theta$  that explain best the observed data

## Parameter estimation

- **Maximum likelihood (ML)**

maximize  $p(D | \Theta, \xi)$

- yields: one set of parameters  $\Theta_{ML}$
- the target distribution is approximated as:

$$\hat{p}(\mathbf{X}) = p(\mathbf{X} | \Theta_{ML})$$

- **Bayesian parameter estimation**

- uses the posterior distribution over possible parameters

$$p(\Theta | D, \xi) = \frac{p(D | \Theta, \xi)p(\Theta | \xi)}{p(D | \xi)}$$

- Yields: all possible settings of  $\Theta$  (and their “weights”)
- The target distribution is approximated as:

$$\hat{p}(\mathbf{X}) = p(\mathbf{X} | D) = \int_{\Theta} p(\mathbf{X} | \Theta)p(\Theta | D, \xi)d\Theta$$

## Parameter estimation

### Other possible criteria:

- **Maximum a posteriori probability (MAP)**

maximize  $p(\Theta | D, \xi)$  (mode of the posterior)

- Yields: one set of parameters  $\Theta_{MAP}$
- Approximation:

$$\hat{p}(\mathbf{X}) = p(\mathbf{X} | \Theta_{MAP})$$

- **Expected value of the parameter**

$\hat{\Theta} = E(\Theta)$  (mean of the posterior)

- Expectation taken with regard to posterior  $p(\Theta | D, \xi)$
- Yields: one set of parameters
- Approximation:

$$\hat{p}(\mathbf{X}) = p(\mathbf{X} | \hat{\Theta})$$

## Density estimation

- **So far we have covered density estimation for “simple” distribution models:**
  - Bernoulli
  - Binomial
  - Multinomial
  - Gaussian
  - Poisson

### **But what if:**

- The dimension of  $\mathbf{X} = \{X_1, X_2, \dots, X_d\}$  is large
  - Example: patient data
- Compact parametric distributions do not seem to fit the data
  - E.g.: multivariate Gaussian may not fit
- We have only a “small” number of examples to do accurate parameter estimates

---

CS 2750 Machine Learning

## How to learn complex distributions

How to learn complex multivariate distributions  $\hat{p}(\mathbf{X})$  with large number of variables?

### **One solution:**

- **Decompose the distribution using conditional independence relations**
- **Decompose the parameter estimation problem to a set of smaller parameter estimation tasks**

Decomposition of distributions under conditional independence assumption is the main idea behind **Bayesian belief networks**

---

CS 2750 Machine Learning

## Example

### Problem description:

- **Disease:** pneumonia
- **Patient symptoms (findings, lab tests):**
  - Fever, Cough, Paleness, WBC (white blood cells) count, Chest pain, etc.

### Representation of a patient case:

- Symptoms and disease are represented as random variables

### Our objectives:

- **Describe a multivariate distribution representing the relations between symptoms and disease**
- **Design of inference and learning procedures for the multivariate model**

CS 2750 Machine Learning

## Joint probability distribution

### Joint probability distribution (for a set variables)

- Defines probabilities for all possible assignments to values of variables in the set

$P(\text{pneumonia}, \text{WBCcount})$   $2 \times 3$  table

		$\text{WBCcount}$			$P(\text{Pneumonia})$
		<i>high</i>	<i>normal</i>	<i>low</i>	
$\text{Pneumonia}$	<i>True</i>	0.0008	0.0001	0.0001	0.001
	<i>False</i>	0.0042	0.9929	0.0019	0.999
		0.005	0.993	0.002	

$P(\text{WBCcount})$

**Marginalization** (summing of rows, or columns)  
- summing out variables

CS 2750 Machine Learning

## Variable independence

- The joint distribution over a subset of variables can be always computed from the joint distribution through marginalization
- Not the other way around !!!
  - Only exception: when variables are independent

$$P(A, B) = P(A)P(B)$$

$P(\text{pneumonia}, \text{WBCcount})$	$\text{WBCcount}$	$P(\text{Pneumonia})$																
	<table border="1" style="border-collapse: collapse;"> <tr> <td></td> <td style="text-align: center;"><i>high</i></td> <td style="text-align: center;"><i>normal</i></td> <td style="text-align: center;"><i>low</i></td> </tr> <tr> <td style="text-align: center;"><i>True</i></td> <td style="text-align: center;">0.0008</td> <td style="text-align: center;">0.0001</td> <td style="text-align: center;">0.0001</td> </tr> <tr> <td style="text-align: center;"><i>False</i></td> <td style="text-align: center;">0.0042</td> <td style="text-align: center;">0.9929</td> <td style="text-align: center;">0.0019</td> </tr> <tr> <td></td> <td style="text-align: center;">0.005</td> <td style="text-align: center;">0.993</td> <td style="text-align: center;">0.002</td> </tr> </table>		<i>high</i>	<i>normal</i>	<i>low</i>	<i>True</i>	0.0008	0.0001	0.0001	<i>False</i>	0.0042	0.9929	0.0019		0.005	0.993	0.002	<div style="border: 1px solid red; padding: 2px; display: inline-block;">                 0.001 0.999             </div>
	<i>high</i>	<i>normal</i>	<i>low</i>															
<i>True</i>	0.0008	0.0001	0.0001															
<i>False</i>	0.0042	0.9929	0.0019															
	0.005	0.993	0.002															
$P(\text{WBCcount})$																		

CS 2750 Machine Learning

## Conditional probability

### Conditional probability :

- Probability of A given B

$$P(A | B) = \frac{P(A, B)}{P(B)}$$

- Conditional probability is defined in terms of joint probabilities
- Joint probabilities can be expressed in terms of conditional probabilities

$$P(A, B) = P(A | B)P(B) \quad \text{(product rule)}$$

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \quad \text{(chain rule)}$$

- Conditional probability – is useful for **various probabilistic inferences**

$$P(\text{Pneumonia} = \text{True} | \text{Fever} = \text{True}, \text{WBCcount} = \text{high}, \text{Cough} = \text{True})$$

CS 2750 Machine Learning



## Modeling uncertainty with probabilities

- **Full joint distribution:**
  - joint distribution over all random variables that define the domain
  - it is sufficient to do any type of probabilistic inferences

## Inference

**Any query can be computed from the full joint distribution !!!**

- **Joint over a subset of variables** is obtained through marginalization

$$P(A = a, C = c) = \sum_i \sum_j P(A = a, B = b_i, C = c, D = d_j)$$

- **Conditional probability over a set of variables**, given other variables' values is obtained through marginalization and definition of conditionals

$$\begin{aligned} P(D = d \mid A = a, C = c) &= \frac{P(A = a, C = c, D = d)}{P(A = a, C = c)} \\ &= \frac{\sum_j P(A = a, B = b_j, C = c, D = d)}{\sum_i \sum_j P(A = a, B = b_i, C = c, D = d_j)} \end{aligned}$$

## Inference

Any query can be computed from the full joint distribution !!!

- Any joint probability can be expressed as a product of conditionals via the **chain rule**.

$$\begin{aligned}P(X_1, X_2, \dots, X_n) &= P(X_n | X_1, \dots, X_{n-1})P(X_1, \dots, X_{n-1}) \\ &= P(X_n | X_1, \dots, X_{n-1})P(X_{n-1} | X_1, \dots, X_{n-2})P(X_1, \dots, X_{n-2}) \\ &= \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1})\end{aligned}$$

- It is often easier to define the distribution in terms of conditional probabilities:
  - E.g.  $\mathbf{P}(\text{Fever} | \text{Pneumonia} = T)$   
 $\mathbf{P}(\text{Fever} | \text{Pneumonia} = F)$

## Modeling uncertainty with probabilities

- **Full joint distribution:** joint distribution over all random variables defining the domain
  - it is sufficient to represent the complete domain and to do any type of probabilistic inferences

### Problems:

- **Space complexity.** To store full joint distribution requires to remember  $O(d^n)$  numbers.  
 $n$  – number of random variables,  $d$  – number of values
- **Inference complexity.** To compute some queries requires  $O(d^n)$  steps.
- **Acquisition problem.** Who is going to define all of the probability entries?

## Pneumonia example. Complexities.

- **Space complexity.**

- Pneumonia (2 values: T,F), Fever (2: T,F), Cough (2: T,F), WBCcount (3: high, normal, low), paleness (2: T,F)
- Number of assignments:  $2*2*2*3*2=48$
- We need to define at least 47 probabilities.

- **Time complexity.**

- Assume we need to compute the probability of Pneumonia=T from the full joint

$$P(\text{Pneumonia} = T) = \sum_{i \in T, F} \sum_{j \in T, F} \sum_{k=h, n, l} \sum_{u \in T, F} P(\text{Fever} = i, \text{Cough} = j, \text{WBCcount} = k, \text{Pale} = u)$$

- Sum over  $2*2*3*2=24$  combinations

---

CS 2750 Machine Learning

## Bayesian belief networks (BBNs)

### Bayesian belief networks.

- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables

- **A and B are independent**

$$P(A, B) = P(A)P(B)$$

- **A and B are conditionally independent given C**

$$P(A, B | C) = P(A | C)P(B | C)$$

$$P(A | C, B) = P(A | C)$$

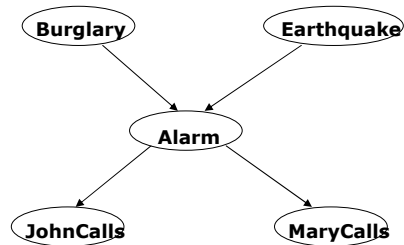
---

CS 2750 Machine Learning

## Alarm system example

- Assume your house has an **alarm system** against **burglary**. You live in the seismically active area and the alarm system can get occasionally set off by an **earthquake**. You have two neighbors, **Mary** and **John**, who do not know each other. If they hear the alarm they call you, but this is not guaranteed.
- We want to represent the probability distribution of events:
  - Burglary, Earthquake, Alarm, Mary calls and John calls

### Causal relations



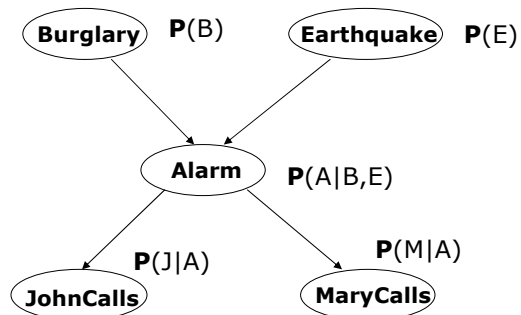
CS 2750 Machine Learning

## Bayesian belief network

### 1. Directed acyclic graph

- Nodes** = random variables  
Burglary, Earthquake, Alarm, Mary calls and John calls
- Links** = direct (causal) dependencies between variables.

The chance of Alarm being is influenced by Earthquake,  
The chance of John calling is affected by the Alarm

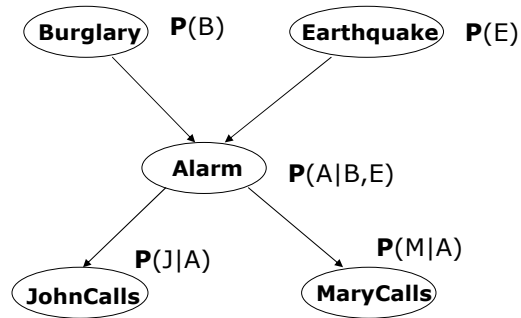


CS 2750 Machine Learning

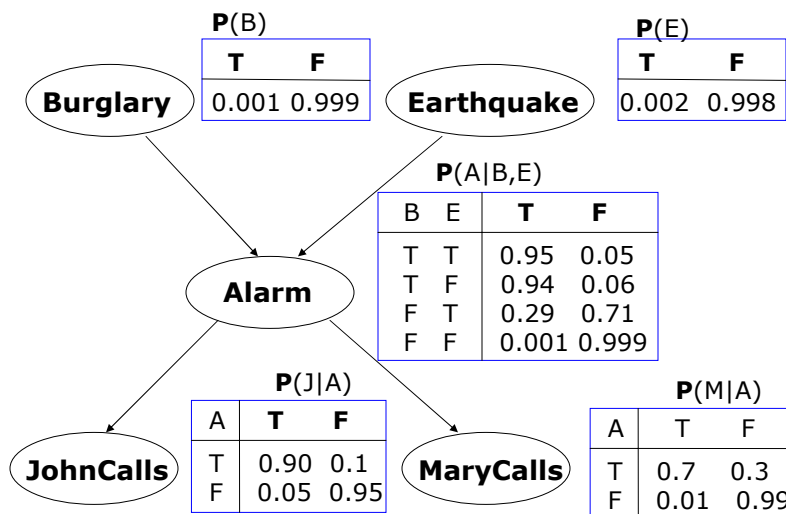
# Bayesian belief network

## 2. Local conditional distributions

- relate variables and their parents



# Bayesian belief network



## Full joint distribution in BBNs

**Full joint distribution** is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

### Example:

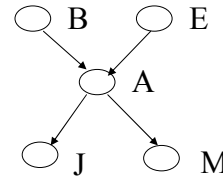
Assume the following assignment of values to random variables

$$B = T, E = T, A = T, J = T, M = F$$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

$$P(B = T)P(E = T)P(A = T \mid B = T, E = T)P(J = T \mid A = T)P(M = F \mid A = T)$$



## Bayesian belief networks (BBNs)

### Bayesian belief networks

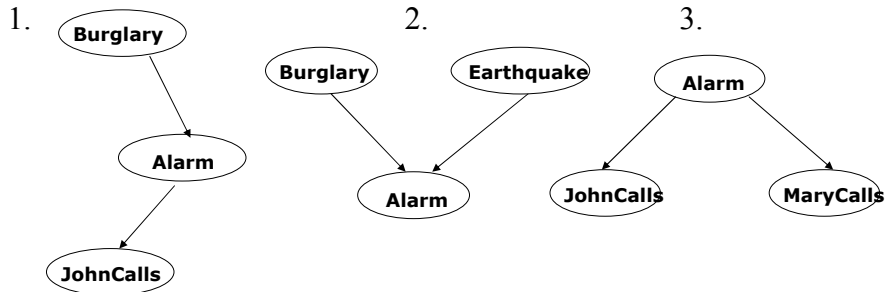
- Represent the full joint distribution over the variables more compactly using the product of local conditionals.
- **But how did we get to local parameterizations?**

### Answer:

- **Graphical structure** encodes **conditional and marginal independences** among random variables
- **A and B are independent**  $P(A, B) = P(A)P(B)$
- **A and B are conditionally independent given C**  
$$P(A \mid C, B) = P(A \mid C)$$
$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$
- **The graph structure implies the decomposition !!!**

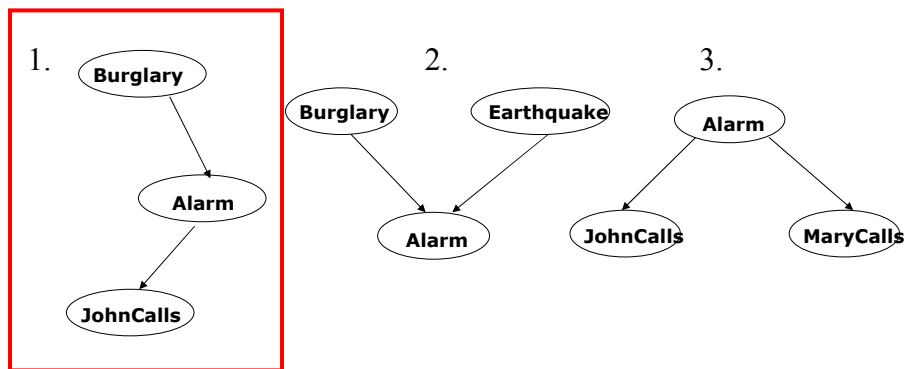
## Independences in BBNs

3 basic independence structures:



CS 2750 Machine Learning

## Independences in BBNs



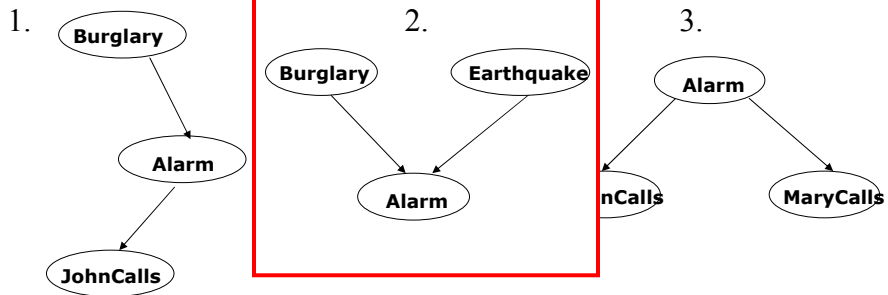
1. JohnCalls is **independent** of Burglary given Alarm

$$P(J | A, B) = P(J | A)$$

$$P(J, B | A) = P(J | A)P(B | A)$$

CS 2750 Machine Learning

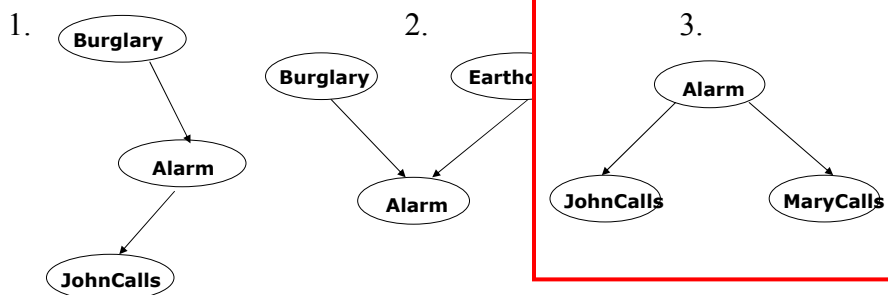
## Independences in BBNs



2. Burglary is **independent** of Earthquake (not knowing Alarm)  
 Burglary and Earthquake **become dependent** given Alarm !!

$$P(B, E) = P(B)P(E)$$

## Independences in BBNs



3. MaryCalls is **independent** of JohnCalls given Alarm

$$P(J | A, M) = P(J | A)$$

$$P(J, M | A) = P(J | A)P(M | A)$$



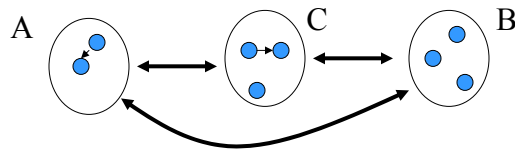
## Independences in BBN

- BBN distribution models many conditional independence relations relating distant variables and sets
- These are defined in terms of the graphical criterion called d-separation
- **D-separation in the graph**
  - Let  $X, Y$  and  $Z$  be three sets of nodes
  - If  $X$  and  $Y$  are d-separated by  $Z$  then  $X$  and  $Y$  are conditionally independent given  $Z$
- **D-separation :**
  - **A is d-separated from B given C** if every undirected path between them is **blocked**
- **Path blocking**
  - 3 cases that expand on three basic independence structures

CS 2750 Machine Learning

## Undirected path blocking

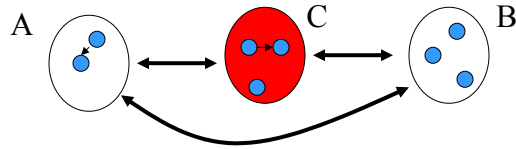
A is d-separated from B given C if every undirected path between them is **blocked**



CS 2750 Machine Learning

## Undirected path blocking

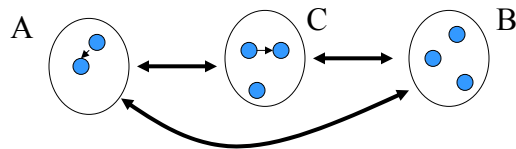
A is d-separated from B given C if every undirected path between them is **blocked**



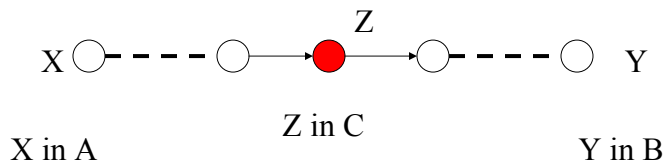
CS 2750 Machine Learning

## Undirected path blocking

A is d-separated from B given C if every undirected path between them is **blocked**



- 1. Path blocking with a linear substructure

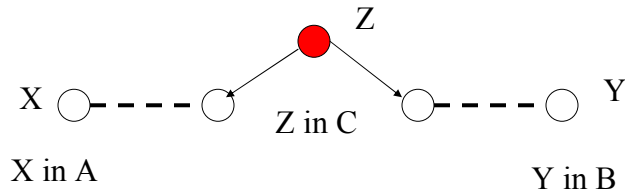


CS 2750 Machine Learning

## Undirected path blocking

A is d-separated from B given C if every undirected path between them is **blocked**

- 2. Path blocking with the wedge substructure

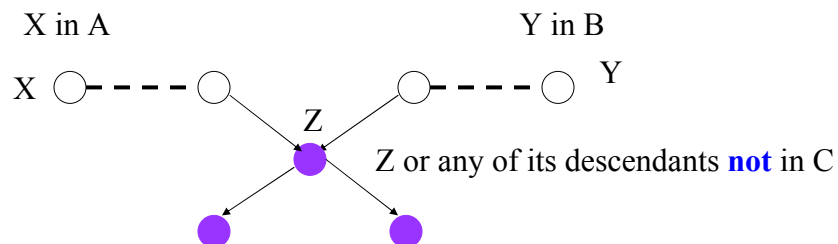


CS 2750 Machine Learning

## Undirected path blocking

A is d-separated from B given C if every undirected path between them is **blocked**

- 3. Path blocking with the vee substructure



CS 2750 Machine Learning

