# Inference in DAGs, Parametrized Conditional Distributions,Influence Decision Diagrams

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

February 3, 2018

# Contents

- Summary of Markov Properties of DGMs, I-Equivalence, I-Map, From I-Map to Factorization, From Factorization to I-Map, Directed Graphs as Distribution Filters, Perfect Map, Completeness of D-Separation

- Inference in DAGs

- Discrete and Gaussian Variables, Parametrized Conditional Distributions, Linear Gaussian Models, Plate Notation, Polynomial Regression, State Space Models, Context Specific Independence, Learning from Complete Data

- Influence Decision Diagrams, Posterior Expected Utility, Maximum Expected Utility, Value of Perfect Information, Partially Observed Markov Decision Process, Markov Decision Process

- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 10
- Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Chapter 3
- Chris Bishop, Pattern Recognition and Machine Learning, Chapter 8
- Jordan, M. I. (2007). An introduction to probabilistic graphical models. In preparation (Chapter 2) – Also review article entitled `Graphical Models'
- Video Lectures on Machine Learning, Z. Gahramani, C. Bishop and others.

# Summary of Markov Properties of DGMs

- We have now described three Markov properties for DAGs:

*G: Directed Global Markov Property* $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_E \iff$ A is d-separated from B given E

*L: Directed Local Markov Property* $t \perp nd(t) \setminus pa(t) \mid pa(t)$

*O: Ordered Markov Property* $t \perp pred(t) \setminus pa(t) \mid pa(t)$

- It is obvious that $G \Rightarrow L \Rightarrow O$.

- What is less obvious, but nevertheless true, is that $O \Rightarrow L \Rightarrow G$. Hence all *these properties are equivalent*.

- Furthermore, any distribution *p* that is Markov wrt *G* can be factorized as;

*F: Factorization Property*

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

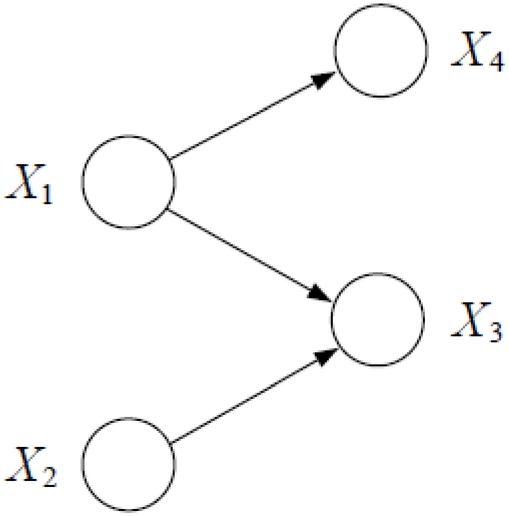
- Clearly $O \Rightarrow F$, but one can show that the converse also holds.
  - Koller, D. and N. Friedman (2009). Probabilistic Graphical Models: Principles and Techniques. MIT Press.

# Characterization of Directed Graphs

- Consider $\mathcal{D}_1$ being the joint distribution of the directed graph (for any values of the conditional Tables)

$$p(x_1, \ldots, x_n) \triangleq \prod_{i=1}^{n} p(x_i \mid x_{\pi_i})$$

- *There is a family* $\mathcal{D}_2$ of distributions associated with G that includes all $p(x_1, \ldots, x_n)$ that satisfy every CI relation associated with the graph.



$$X_1 \perp\!\!\!\perp X_2$$

$$X_2 \perp\!\!\!\perp X_4$$

$$X_2 \perp\!\!\!\perp X_4 \mid X_1$$

$$X_3 \perp\!\!\!\perp X_4 \mid X_1$$

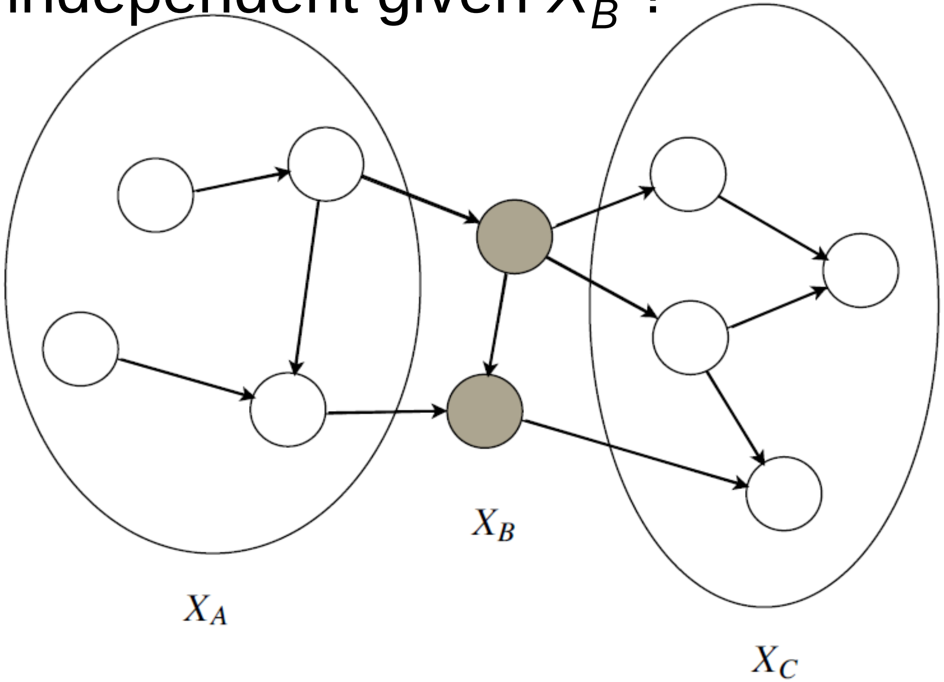
$$X_2 \perp\!\!\!\perp X_4 \mid \{X_1, X_3\}$$

$$\{X_2, X_3\} \perp\!\!\!\perp X_4 \mid X_1$$

- *Theorem: The two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical.*

# General Conditional Independence

- For any given (nonoverlapping) sets of nodes $X_A$, $X_B$, and $X_C$, and a given graph (factorization) $G$, are $X_A$ and $X_C$ conditionally independent given $X_B$ ?



- *What is the set **of all** CI assertions for graph G ?*

- *D-Separation* and the *Bayes Ball Algorithm* provide the answer.

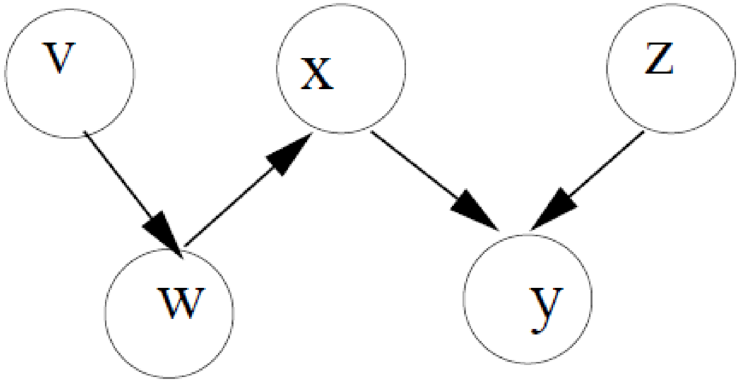
# I-Equivalence

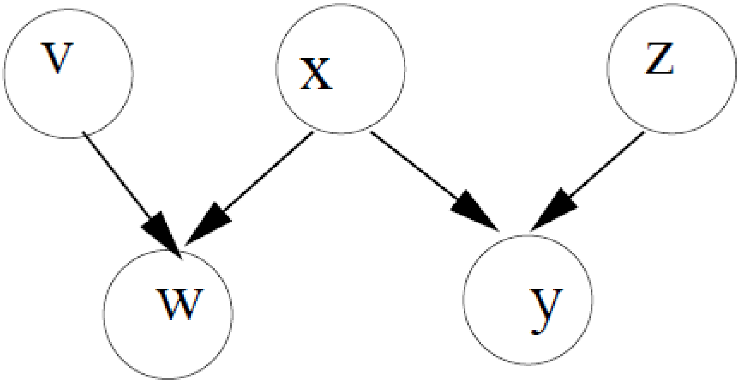
- We write $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_C$ if $A$ is independent of $B$ given $C$ in the graph $G$.

- Let *$I(G)$ be the set of all such CI statements encoded by the graph*.

- Definition: $G_1$ and $G_2$ are I-equivalent if $I(G_1) = I(G_2)$

- Theorem: *If $G_1$ and $G_2$ have the same* ***undirected*** *skeleton and the same set of v-structures, then they are I-equivalent.*

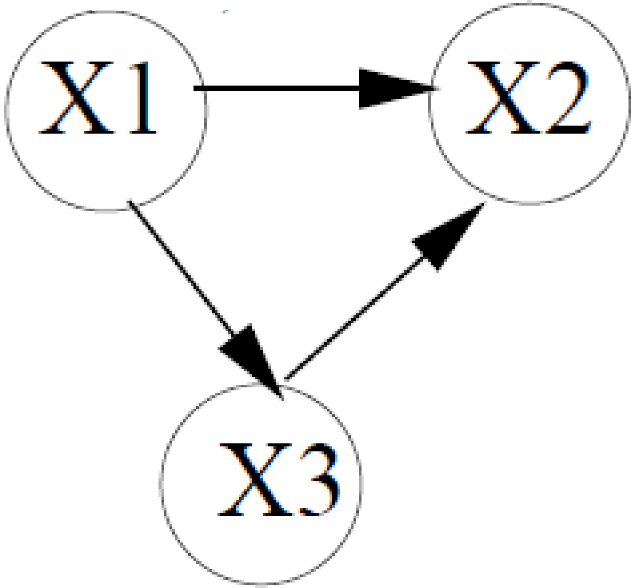


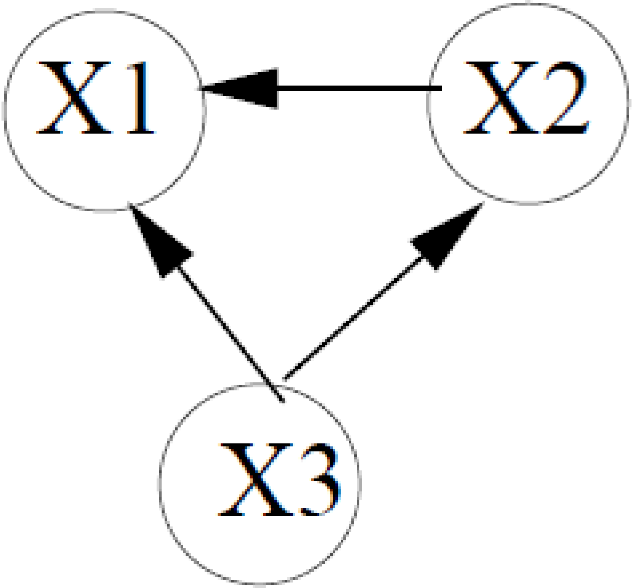


# I-Equivalence

- *If $G_1$ is I-equivalent to $G_2$, the two graphs do not necessarily have the same skeleton and v-structures.*

- *For example, the two graphs below have $I(G_1)=I(G_2)=0$.*





- *We can only identify graph structure up to I-equivalence, i.e., we cannot always tell the direction of all the arrows from observational data.*

- This is important in the context of graph structure learning and causality.

# Conditional Independence & I-map

- At the heart of any *directed* graphical model is a set of conditional independence (CI) assumptions

- We write $\boldsymbol{x}_A \perp_G \boldsymbol{x}_B | \boldsymbol{x}_C$ if $A$ is independent of $B$ given $C$ in the graph $G$.

- Let $I(G)$ be the set of all such CI statements encoded by the graph

- We say $I(G)$ is an I-map for $p$ if $I(G) \subseteq I(p)$, where $I(p)$ is the set of all conditional independent statements that hold for distribution $p$.

  - This allows us to use $G$ as a safe proxy for $p$.

# Independence Properties of Distributions

- Definition: let I(P) be the set of independence properties of the form $X \perp Y | Z$ that hold in distribution P.

- Consider a discrete distribution defined as follows:

| $X$ | $Y$ | $p(X,Y)$ |
|---|---|---|
| 0 | 0 | 0.08 |
| 0 | 1 | 0.32 |
| 1 | 0 | 0.12 |
| 1 | 1 | 0.48 |

- Note from this Table that:

$$P(X = 1) = 0.48 + 0.12 = 0.6, \qquad P(Y = 1) = 0.32 + 0.48 = 0.8$$

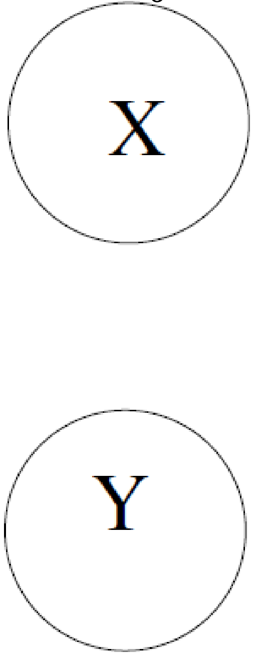
$$P(X = 1, Y = 1) = 0.48 = 0.6 \times 0.8 \Rightarrow P(X{=}x, Y{=}y) = P(X{=}x)P(Y{=}y) \ \forall x, y$$

$$\Rightarrow (X \perp Y) \in I(P) \quad \text{or } P \models (X \perp Y)$$

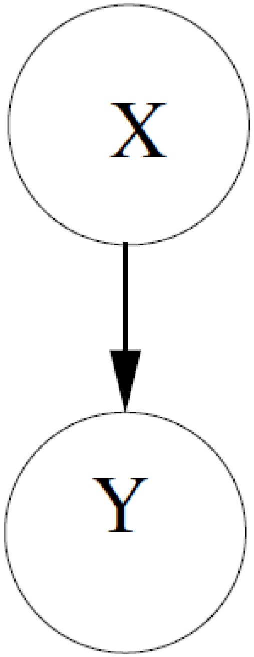
# Local Independence Properties $I_\ell(G)$ of G

- Let $I_l(G)$ *be the set of all CI statements encoded by the graphs shown*. By inspection, we can write:

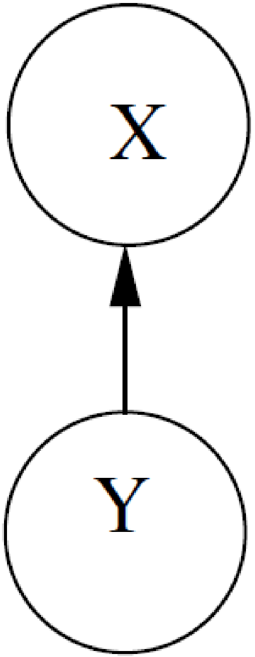
$G_0$



$G_{X \to Y}$



$G_{Y \to X}$



$$I_\ell\left(G_0\right) = \left\{\left(X \perp Y\right)\right\} \qquad I_\ell\left(G_{X \to Y}\right) = 0 \qquad I_\ell\left(G_{Y \to X}\right) = 0$$

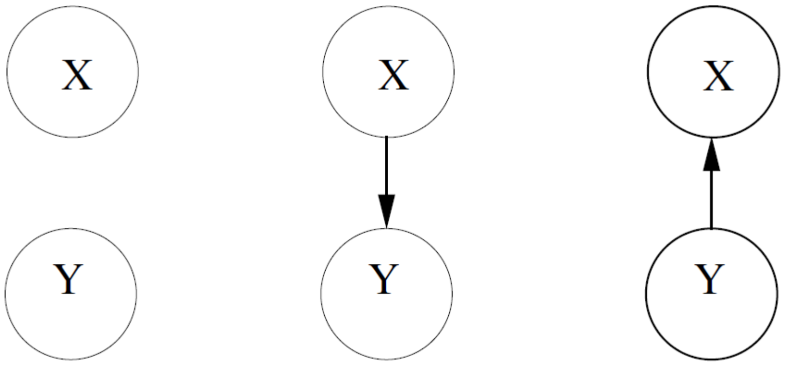
# I-Maps

- Definition: A DAG G is an I-map (independency-map) of P if

$$I_l(G) \subseteq I(P)$$

i.e. P satisfies all local independencies associated with G. However, P may have additional independencies not reflected by G.

- From previous example



$I_\ell(G_0)=\{(X \perp Y)\}$ $\quad I_\ell\left(G_{X\to Y}\right)=0$ $\quad I_\ell\left(G_{Y\to X}\right)=0$

| $X$ | $Y$ | $p(X,Y)$ |
|---|---|---|
| 0 | 0 | 0.08 |
| 0 | 1 | 0.32 |
| 1 | 0 | 0.12 |
| 1 | 1 | 0.48 |

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

- Hence all three graphs are I-maps of $P$.

# I-Maps

- Let *$I_l(G)$ be the set of all such CI statements encoded by the graph*.

- We say *that G is an* ***I-map*** *(independence map) for p, or* ***that p is Markov*** *wrt G, iff $I_l(G) \subseteq I(p)$,* where *I*(*p*) is the set of all CI statements that hold for distribution *p*.

- In other words, *the graph is an I-map if it does not make any assertions of CI that are not true of the distribution.*

- This allows us to ***use the graph as a safe proxy for p when reasoning about p's CI properties***.

- This is helpful for designing algorithms that work for large classes of distributions, regardless of their specific numerical parameters $\boldsymbol{\theta}$.

# From I-MAP to Factorization

- ❑ Definition: P factorizes according to G if P can be written as

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

- ❑ Theorem: If G is an I-map of P, then P factorizes according to G. The proof can be seen as follows. Assume a topological ordering

$$p(\boldsymbol{X}_{1:V} \mid G) = p(X_1)\,p\left(X_2 \mid X_1\right)p(X_3 \mid X_1, X_2)...$$

$$= \prod_{i=1}^{V} p(x_i \mid \boldsymbol{x}_{1:i-1})$$

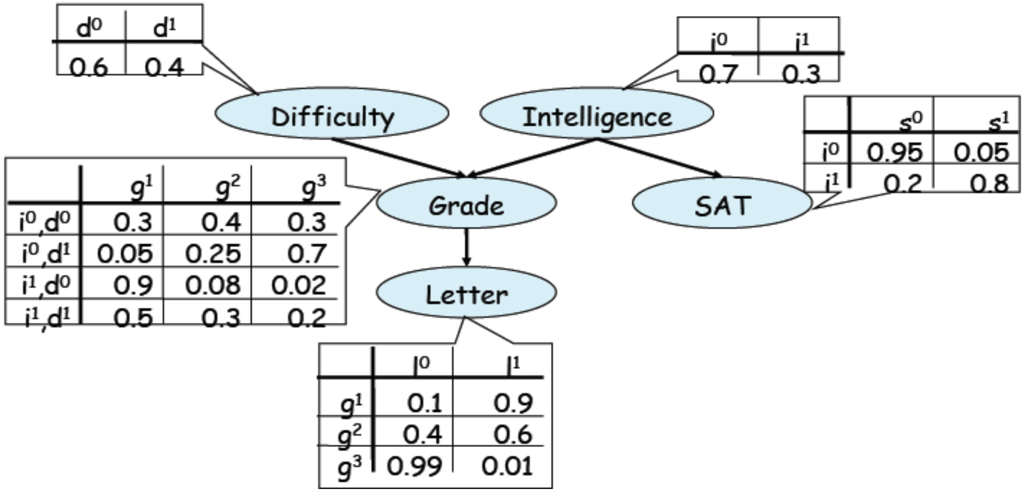
$$= \prod_{i=1}^{V} p\left(X_i \mid \boldsymbol{X}_{pa(i)}, \boldsymbol{Z}\right) = \prod_{i=1}^{V} p(X_i \mid \boldsymbol{X}_{pa(i)})\left(where : \boldsymbol{Z} \subseteq \boldsymbol{X}_{NonDescendants(i)}\right)$$

where the last step follows since G is an I-Map of p.

# Example

- In the graph shown below, the factorization of P is as follows:

$$p\left(D,I,G,S,L\right)=p\left(D\right)p\left(I\right)p\left(G\,|\,D,I\right)p\left(S\,|\,I\right)p(L\,|\,G)$$



| $d^0$ | $d^1$ |
|---|---|
| 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 |

$$L \perp I,D,S\,|\,G$$

$$S \perp D,G,L\,|\,I$$

$$G \perp S\,|\,I,D$$

$$I \perp D$$

$$D \perp I,S$$

- The number of independent parameters is now 1+1+8+2+3=15.

- Specification of the full joint requires 48 -1 = 47 parameters.

- In general, the number of parameters is less than V $2^k$, where k=# parents (exponentially smaller than the number of parameters needed in the joint $2^V$-1).

# Compact Representation of the Joint Distribution

- Theorem: If G is an I-map of P, then P factorizes according to G.

$$p(\boldsymbol{x}_{1:V} \mid G) = \prod_{t=1}^{V} p(x_t \mid \boldsymbol{x}_{pa(t)})$$

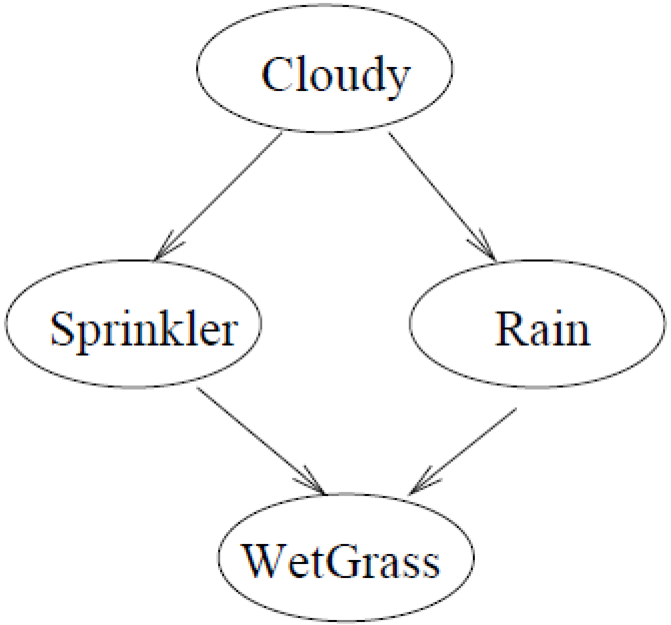
- Corollary: *If G is an I-map of P, then we can represent P using G and a set of conditional probability distributions (CPDs), $P(X_i|Pa(X_i))$, one per node.*

- Definition: *A Bayesian network (aka belief network) representing distribution P is an I-map of P and a set of CPDs.*

- For binary random variables, the Bayes net takes $\mathcal{O}(V2^K)$ parameters (K = max. num. parents), whereas full joint takes $\mathcal{O}(2^V)$ parameters.

- Factored representation is easier to understand, easier to learn and supports more efficient inference.

# Compact Representation of the Joint Distribution



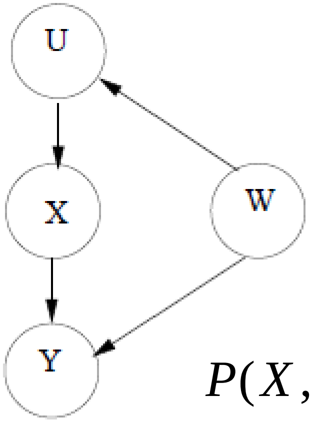
$$P(C, S,R,W) = P(C)P(S|C)P(R|C)P(W|S,R)$$

# From Factorization to I-Map

- *Theorem: If P factorizes according to G, then G is an I-Map of P.*

- We show the proof with an example. Consider a distribution P that factorizes according to G as:

$$P(X,W,U,Y) = p(W)\,p(U|W)\,p(Y|X,W)\,p(X|U)$$



- From this factorization we can derive

$$X \perp W | U$$

$$P(X,W|U) = \frac{\sum_Y P(X,W,U,Y)}{P(U)} = \frac{p(W)\,p(U|W)\,p(X|U)\sum_Y p(Y|X,W)}{P(U)}$$

$$= \frac{p(U,W)\,p(X|U)}{p(U)} \Rightarrow P(X,W|U) = P(X|U)P(W|U)$$

# Minimal I-Maps and Bayesian Nets

- Let G be a fully connected DAG. Then $I_l(G) = \emptyset \subseteq I(P)$ for any P.

- Hence the complete graph is an I-map for any distribution.

- The fully connected graph is an I-map of all distributions, since it makes no CI assertions at all (since it is not missing any edges).

- Definition: *A DAG G is a minimal I-map for P if it is an I-map for P, and if the removal of even a single edge from G renders it not an I-map.*

- We therefore say G is a **minimal I-map** of p if G is an I-map of p, and if there is no G ‘$\subseteq$ G which is an I-map of p.

# Minimal I-Maps and Bayesian Nets

- Construction: pick a node ordering, then let the parents of node $X_i$ be the minimal subset of $U \subseteq \{X_1, . . . ,X_{i-1}\}$ such that:

$$X_i \perp \{X_1, . . . ,X_{i-1}\} \setminus U | U.$$

- Updated Definition: *A Bayesian network (aka belief network) representing distribution P is a minimal I-map of P and a set of CPDs.*

# Conditional Independence Properties of DGMs

- A fully connected graph is an I-map of all distributions, since it makes no CI assertions at all

- We therefore say $G$ is a minimal I-map of $p$ is there is no $G' \subset G$ which is an I-map of $p$.

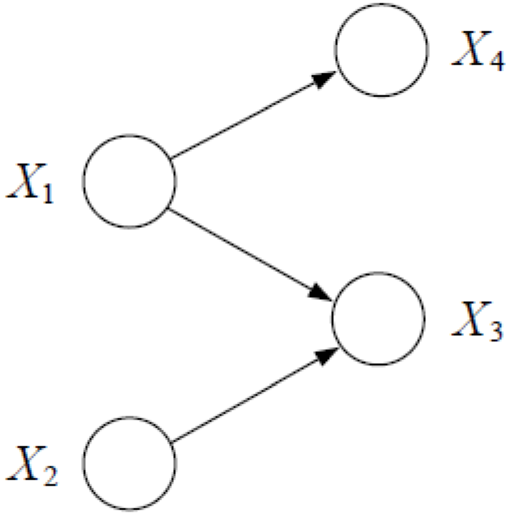
- Recall how to determine if $\boldsymbol{x}_A \perp_G \boldsymbol{x}_B | \boldsymbol{x}_C$

  - For undirected graph, determining unconditional independencies is straightforward based on simple graph separation.

  - However for directed graphical model, we need to take into account the directions of the edges as well (explaining away)

# Characterization of Directed Graphs

- ❑ Consider two distributions: One $\mathscr{D}_1$ being the joint distribution of the directed graph (for any values of the conditional Tables) and the other $\mathscr{D}_2$ defined from all the independent relations between the random variables in the graph.



$$p(x_1, \ldots, x_n) \triangleq \prod_{i=1}^{n} p(x_i \mid x_{\pi_i})$$

$$X_1 \perp\!\!\!\perp X_2$$

$$X_2 \perp\!\!\!\perp X_4$$

$$X_2 \perp\!\!\!\perp X_4 \mid X_1$$

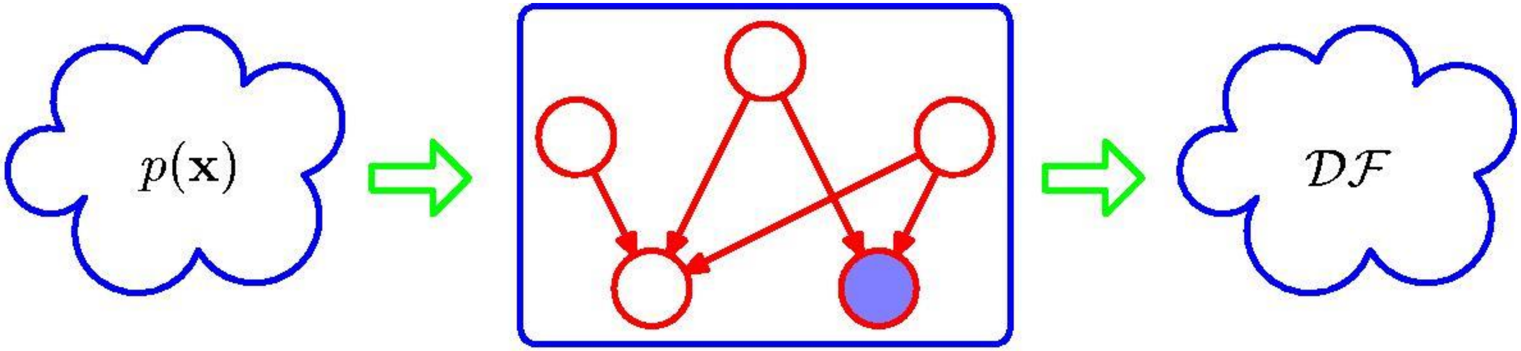
$$X_3 \perp\!\!\!\perp X_4 \mid X_1$$

$$X_2 \perp\!\!\!\perp X_4 \mid \{X_1, X_3\}$$

$$\{X_2, X_3\} \perp\!\!\!\perp X_4 \mid X_1$$

- ❑ *Theorem: The two distributions $\mathscr{D}_1$ and $\mathscr{D}_2$ are identical.*

# Directed Graphs as Distribution Filters

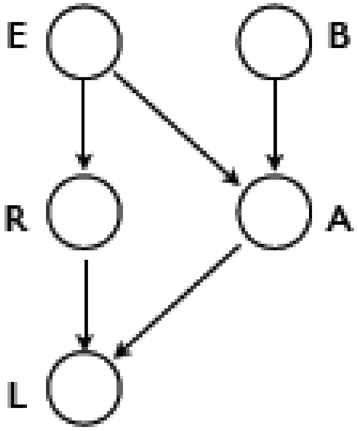


We can view a graphical model (in this case a directed graph) as a filter in which *a probability distribution p(x) is allowed through the filter if, and only if, it satisfies the directed factorization property.* The set of all possible probability distributions *p(x)* that pass through the filter is denoted $\mathcal{DF}$.

**Note** that for any given graph, *the set of distributions $\mathcal{DF}$ will include any distributions that have additional independence properties beyond those described by the graph.*

# Directed Graphs and Distributions

- The probability distribution associated with the graph needs to be consistent with all the independence relations encoded in the graph.



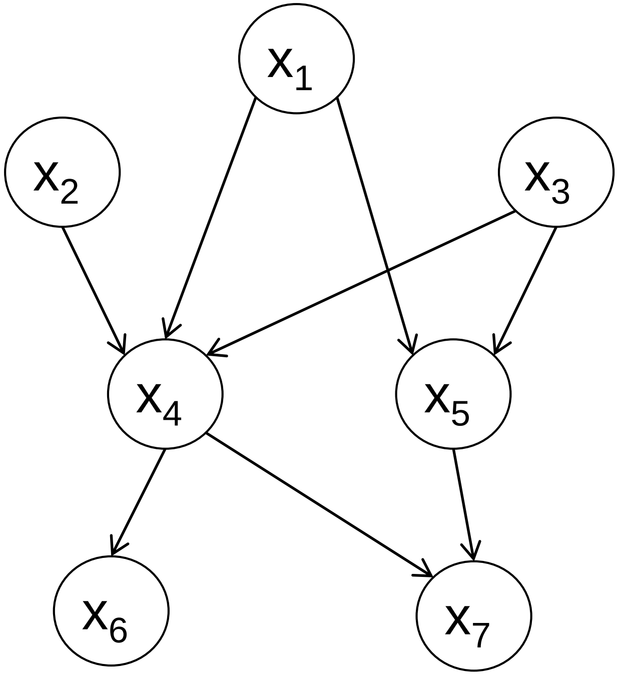
$$p(A,B,E,R,L) = p(E)p(B)p(R|E)p(A|B,E)p(L|A,R)$$

- However, a distribution that is consistent with the graph may satisfy additional independence properties not encoded in the graph, e.g.

$$p(A,B,E,R,L) = p(E)p(B)p(R|E)p(A|E)p(L|R)$$

$$p(A,B,E,R,L) = p(E)p(B)p(R|E)p(A)p(L)$$

# DAGs and Probability Distributions



$$p(x_1,\ldots,x_7) = p(x_1)\,p(x_2)\,p(x_3)\,p(x_4|x_1,x_2,x_3)\,p(x_5|x_1,x_3)\,p(x_6|x_4)\,p(x_7|x_4,x_5)$$

General Factorization

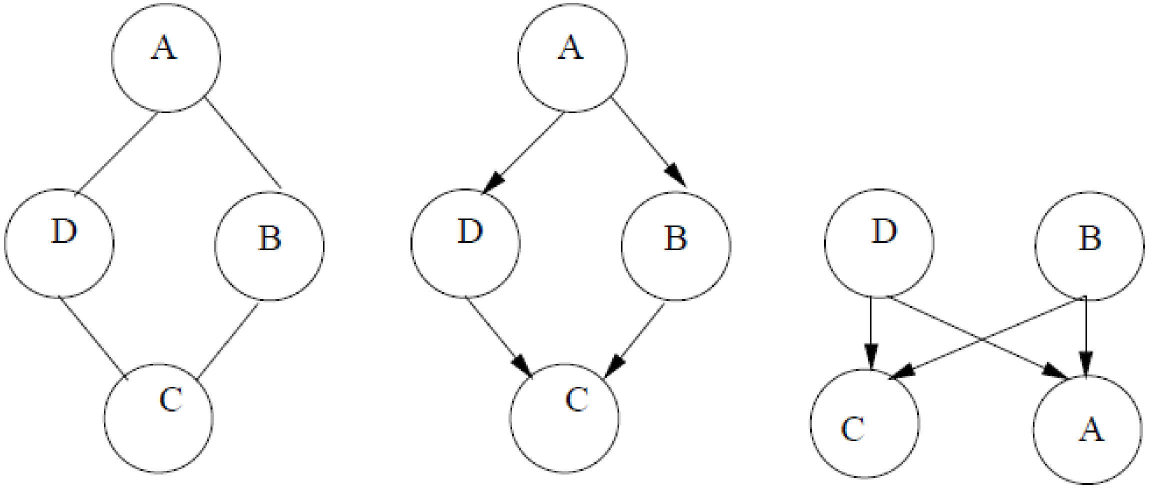
$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|x_{Pa_k})$$

$$x_{Pa_k} = \{x_j, j \in Pa_k\}$$

*Any distribution $p(x_1,x_2,..,x_K)$ consistent with all the independence statements implied by a DAG via D-separation can be written as a product of local conditional distributions of a variable given its parents*

# P-Map (Perfect-Map)

- ❑ Can we find a graph that captures all the independencies in an arbitrary distribution (and no more)?
- ❑ Defn: A DAG G is a perfect map (P-map) for a distribution P if $I(P) = I(G)$.
- ❑ Thm: *not every distribution has a perfect map.*
- ❑ Proof by counterexample. Suppose we have a model where A⊥C|{B,D}, and B ⊥ D|{A,C}. This cannot be represented by any Bayes net.
- ❑ In the example, BN1 wrongly says B ⊥ D|A, BN2 wrongly says B ⊥ D.



# Global Markov Properties of DAGS

- By chaining together local independencies, we can infer more global independencies.

- Defn: X is d-separated (directed-separated) from Y given Z if along every undirected path between X and Y there is a node w s.t. either

  - W has converging arrows (→ w ←) and neither W nor its descendants are in z; or

  - W does not have converging arrows and $W \in Z$.

- Definition: *I(G) = all independence properties that correspond to d-separation*

$$I(G) = \{(X \perp Y \,|Z) : d - sep_G(X; Y \,|Z)\}$$

# Soundness of d-Separation

❑ Theorem: If P factorizes according to G, then I(G) ⊆ I(P).

❑ This means that any independence claim made by the graph is satisfied by all distributions P that factorize according to G (no false claims of independence).

❑ The proof of the theorem is easier highlighted for undirected graphs where d-separation is a simple graph separation (see Koller and Friedman, Chapter 4).

# Completeness of d-Separation

- Theorem (Completeness) v1: For any distribution P that factorizes over G, if $(X \perp Y \mid Z) \in I(P)$, then $\text{dsep}_G(X; Y \mid Z)$.

- Contrapositive rule: $(A \Rightarrow B) \Longleftrightarrow (\neg B \Rightarrow \neg A)$.

- Theorem (Completeness, contrapositive form) v1. If X and Y are not d-separated given Z, then X and Y are dependent in all distributions P that factorize over G.

- This definition of completeness is too strong since P may have conditional independencies that are not evident from the graph.

- eg. Let G be the graph $X \rightarrow Y$ , where $P(Y \mid X)$ is

| $X$ | $Y=0$ | $Y=1$ |
|---|---|---|
| 0 | 0.4 | 0.6 |
| 1 | 0.4 | 0.6 |

- G is I-map of P since $I(G) = \emptyset \subseteq I(P) = \{(X \perp Y)\}$.

- But the CPD encodes $X \perp Y$ which is not evident in the graph.

# Completeness of d-Separation

- ❑ Theorem (Completeness) v2*: If ($X \perp Y \mid Z$) in **all** distributions P that factorize over G, then $\mathrm{dsep}_G(X; Y \mid Z)$.*

- ❑ Theorem (Completeness, contrapositive form) v2: *If X and Y are not d-separated given Z, then X and Y are dependent in some distribution P that factorizes over G.*

- ❑ Theorem: *d-separation is complete.*

- ❑ Proof: See Koller & Friedman, Theorem 3.5, p73.

- ❑ Hence d-separation captures as many of the independencies as possible (without reference to the particular CPDs) for all distributions that factorize over some DAG.

# D-Separation ⇔ Factorization

- □ Consider a DAG $G$ with nodes (variables) $X_1, \ldots, X_V$.

- □ Consider the set $\mathcal{U}$ of all (families of) joint distributions for the same variables

- □ A subset of distributions, $\mathcal{DI} \subseteq \mathcal{U}$, maintain the CI assertions implied by D-separation in $G$

- □ Another subset of distributions, $\mathcal{DF} \subseteq \mathcal{U}$, can be factored according to $G$

- □ It turns out that $\mathcal{DI} = \mathcal{DF}$

# Inference in DAGs

- DAGs provide a compact way to define joint probability distributions. We can use such a joint distribution to perform **probabilistic inference**.

- This refers to the task of estimating unknown quantities from known quantities.

- *For example, in HMMs estimate the hidden states (e.g., words) from the observations (e.g., speech signal).*

- *In the genetic linkage analysis one of the goals was to estimate the likelihood of the data under various DAGs, corresponding to different hypotheses about the location of the disease-causing gene.*

- For posing the inference problem let us suppose we have a set of correlated random variables with joint distribution $p(\mathbf{x}_{1:V} | \boldsymbol{\theta})$. (Assume at this point that $\boldsymbol{\theta}$ are known).

# Inference in DAGs

- We partition $\mathbf{x}_{1:V}$ into the *visible variables $x_v$*, which are observed, and *the hidden variables, $x_h$,* which are unobserved.

- Inference refers to *computing the posterior distribution of the unknowns given the knowns:*

$$p(\boldsymbol{x}_h \mid \boldsymbol{x}_v, \theta) = \frac{p(\boldsymbol{x}_h, \boldsymbol{x}_v \mid \theta)}{p(\boldsymbol{x}_v \mid \theta)} = \frac{p(\boldsymbol{x}_h, \boldsymbol{x}_v \mid \theta)}{\sum_{\boldsymbol{x}_h^{'}} p(\boldsymbol{x}_h^{'}, \boldsymbol{x}_v \mid \theta)}$$

- We are conditioning on the data by clamping the visible variables to their observed values and then normalizing going from $p(\mathbf{x}_h, \mathbf{x}_v)$ to $p(\mathbf{x}_h|\mathbf{x}_v)$.

- $p(\mathbf{x}_v|\boldsymbol{\theta})$ (likelihood of the data) is the *probability of the evidence.*

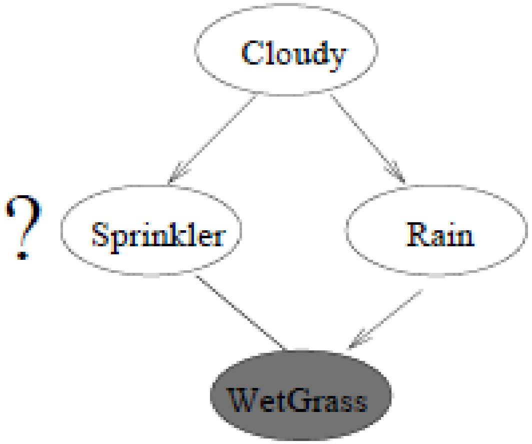
- *Examples:*
  - *Medical diagnosis:* $\mathbf{x}_v$=symptoms, $\mathbf{x}_h$=diseases
  - *Speech recognition:* $\mathbf{x}_v$=acoustic wave form, $\mathbf{x}_h$=spoken words
  - *Genetic Pedigree analysis:* $\mathbf{x}_v$=phenotype, $\mathbf{x}_h$=genotype

# Naïve Inference

- In the example shown, we can represent the joint probability distribution P(C,S,R,W) as a 4D table of $2^4$ = 16 numbers.

- We observe the grass is wet and want to know how likely it was that the sprinkler caused this event.

$$p(s=1|w=1)=\frac{\sum_{r,c=0}^{1} p\left(C=c,s=1,R=r,w=1\right)}{\sum_{s,c,r=0}^{1} p\left(C=c,S=s,R=r,w=1\right)}$$



- Query/hidden variables = {S}

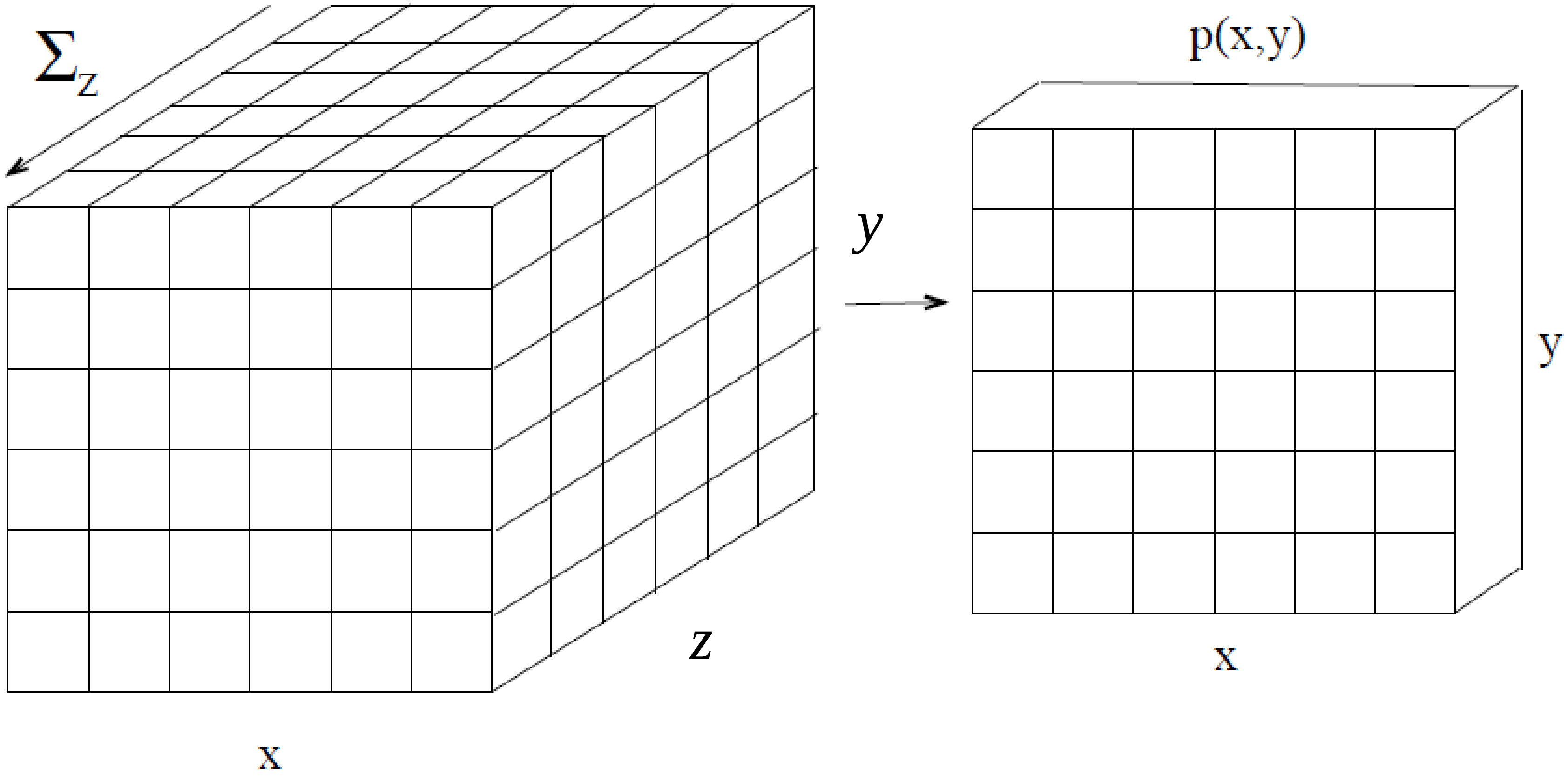
- Visible variables = {W}

- Nuisance variables = {C,R} (need to be integrated out)

# Marginalization for CPTs

- It is easy to marginalize a joint probability distribution when it is represented as a table

$$p(X,Y)=\sum_{Z} p(X,Y,Z)$$



# Representation of CPD as CPTs

- There are several problems in representing the joint CPD as a big table (CPT)

  - *Representation: big table of numbers is hard to understand.*
  - *Inference: computing a marginal $p(X_i)$ takes $\mathcal{O}(2^N)$ time.*
  - *Learning: there are $\mathcal{O}(2^N)$ free parameters to estimate.*

- Graphical models solve all of the above problems by providing a structured representation for the joint probability distribution.

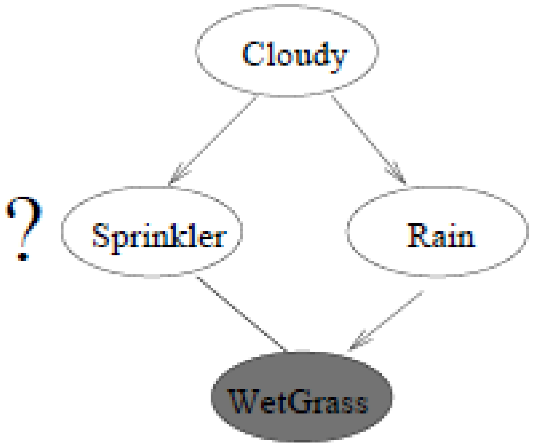
- Graphs encode CI properties and represent families of probability distributions that satisfy these properties.

# Inference by Marginalizing the Joint

- We can answer an query by marginalization of the joint distribution using the factorization implied by the graph. For example:

$$p(s=1|w=1)=\frac{\sum_{c,r} p\left(C=c,s=1,R=r,w=1\right)}{\sum_{s,c,r} p\left(C=c,S=s,R=r,w=1\right)}$$

$$\frac{\sum_{c,r} p\left(C=c\right)p\left(S=1|C=c\right)p\left(R=r|C=c\right)p\left(W=1|S=s,R=r\right)}{\sum_{s,c,r} p\left(C=c,S=s,R=r,w=1\right)}$$



$$p\left(C,S,R,W\right)=p\left(C\right)p\left(S\,|\,C\right)p\left(R\,|\,C\right)p\left(W\,|\,S,R\right)$$

# Discrete & Gaussian Variables

- The exponential family forms useful building blocks for constructing complex probability distributions, and the framework of graphical models is very useful in expressing the way in which these building blocks are linked together.

- Consider the case when the parent and child node each correspond to **discrete variables** or **Gaussian variables**. In these two cases the relationship can be extended hierarchically to construct arbitrarily complex directed acyclic graphs.

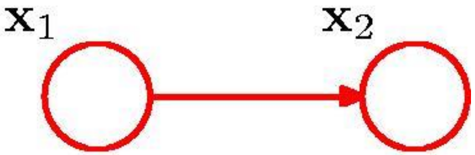
# Discrete Variables

- Suppose two discrete variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, each of which has $K$ possible states, the joint probability distribution is given by

$$p\left(\boldsymbol{x}_1,\boldsymbol{x}_2|\mu\right)=\prod_{i=1}^{K}\prod_{j=1}^{K}\mu_i^{x_{1i}}\mu_j^{x_{2j}}$$

where $\mu_i$ is the probability that state i occurred, $x = (x_1,x_2,...,x_K)$ only has one non-zero value.

- Dependent joint distribution: $K^2 - 1$ parameters



$$p\left(\boldsymbol{x}_1,\boldsymbol{x}_2|\mu\right)=\prod_{i=1}^{K}\prod_{j=1}^{K}\mu_{ij}^{x_{1i}x_{2j}}$$

- Independent joint distribution: $2(K-1)$ parameters

$\mathbf{x}_1$
$\mathbf{x}_2$

$$p\left(\boldsymbol{x}_1,\boldsymbol{x}_2|\mu\right)=\prod_{i=1}^{K}\mu_i^{x_{1i}}\prod_{j=1}^{K}\mu_j^{x_{2j}}$$

# Discrete Variables

❑ *General joint distribution over M variables: $K^M$ -1* parameters grow exponentially with *M*

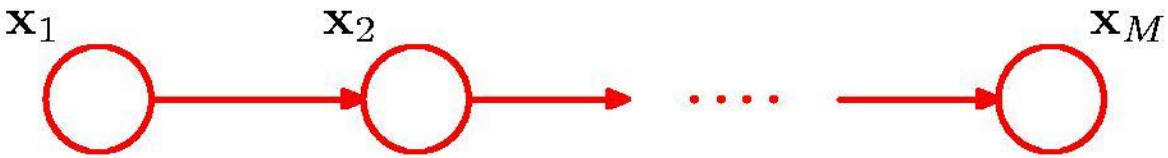
❑ If all of *M* variables are independent: *M(K-1)* parameters

❑ *M-node Markov chain: K-1 + (M-1)K(K-1) parameters*

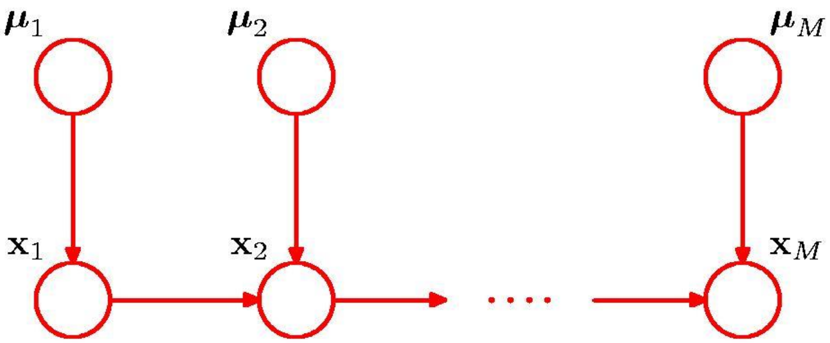
the marginal distribution $p(x_1)$ requires *K-1* parameters

each of the M-1 conditional distributions $p(x_i|x_{i-1})$, for $i=2,\ldots,M$, requires *K(K-1)* parameters

If all conditional distributions share parameters: $K^2$-1 total number of parameters



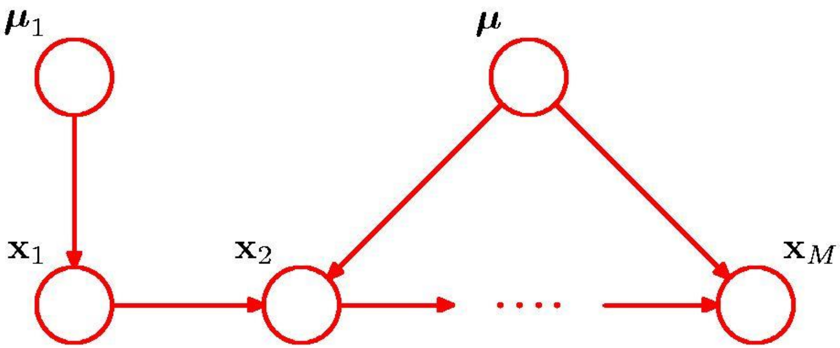
# Discrete Variables: Bayesian Parameters



$$p\left(\{x_m,\mu_m\}\right)=$$

$$p\left(x_1|\mu_1\right)p\left(\mu_1\right)\prod_{m=2}^{M}p\left(x_m|x_{m-1},\mu_m\right)p\left(\mu_m\right)$$

$$p\left(\mu_m\right)=\mathcal{Dir}\left(\mu_m|\alpha_m\right)$$



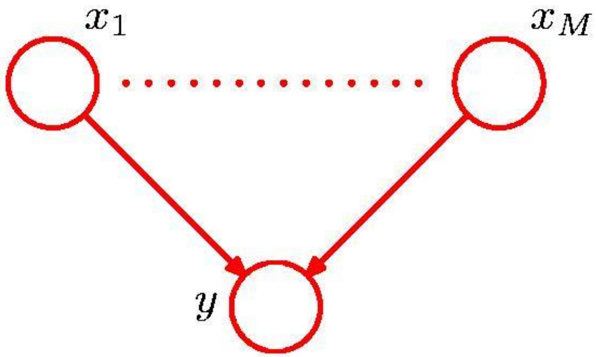
$$p\left(\{x_m,\mu_m\}\right)=$$

$$p\left(x_1|\mu_1\right)p\left(\mu_1\right)\prod_{m=2}^{M}p\left(x_m|x_{m-1},\mu\right)p\left(\mu\right)$$

Tying of parameters

# Parameterized Conditional Distributions

- Use parameterized models to control the exponential growth in the number of parameters in models of discrete variables.



If $x_1,\dots,x_M$ are discrete, *K*-state variables, $p\left(y=1 \middle| x_1,\dots,x_M\right)$ in general has $\mathcal{O}(K^M)$ parameters.

The parameterized form using the sigmoid function

$$p\left(y=1 \middle| x_1,\dots,x_M\right)=\sigma\left(w_0+\sum_{i=1}^{M} w_i x_i\right)=\sigma\left(w^T x\right), \sigma\left(a\right)=\frac{1}{1+e^{-a}}$$

requires only *M* + *1* parameters!

# Linear-Gaussian Models

- Consider an arbitrary directed acyclic graph over *D* variables in which node *i* represents a Gaussian variable $x_i$. The mean is a linear combination of its parent nodes $pa_i$

$$p\left(x_i \middle| \mathrm{Pa}_i\right) = \mathcal{N}\left(x_i; \sum_{j\in \mathrm{Pa}_i} w_{ij}x_j + b_i, \upsilon_i\right), i.e.\ x_i = \sum_{j\in \mathrm{Pa}_i} w_{ij}x_j + b_i + \sqrt{\upsilon_i}\varepsilon_i,\ \varepsilon_i \sim \mathcal{N}(0,1)$$

$$\mathbb{E}\left[\varepsilon_i\right] = 0,\ \mathbb{E}\left[\varepsilon_i\varepsilon_j\right] = I_{ij}$$

- Note that the joint distribution $p(\boldsymbol{x})$ is Gaussian:

$$\ln p(\boldsymbol{x}) = \sum_{i=1}^{D} \ln p\left(x_i \middle| \mathrm{Pa}_i\right) = \sum_{i=1}^{D} \frac{1}{2\upsilon_i}\left(x_i - \sum_{j\in \mathrm{Pa}_i} w_{ij}x_j - b_i\right)^2 + const.$$

- We can *compute the mean and covariance of $p(\boldsymbol{x})$ recursively.*

$$\mathbb{E}\left[x_i\right] = \sum_{j\in \mathrm{Pa}_i} w_{ij}\mathbb{E}\left[x_j\right] + b_i \Rightarrow \qquad \mathbb{E}\left[\boldsymbol{x}\right] = \left(\mathbb{E}\left[x_1\right],\ldots,\mathbb{E}\left[x_D\right]\right)^T$$

$$\mathrm{cov}\left[x_i, x_j\right] = \sum_{k\in \mathrm{Pa}_j} w_{jk}\,\mathrm{cov}\left[x_i, x_k\right] + I_{ij}\sqrt{\upsilon_i\upsilon_j}$$

- Roweis, S. and Z. Ghahramani (1999). A unifying review of linear Gaussian models. *Neural Computation* **11**(2), 305–345.

# Linear-Gaussian Models

- Note the covariance matrix for $x_i$ and $x_j$ (*assume i≤j*)

$$\operatorname{cov}\left[x_i, x_j\right] = \mathbb{E}\left[\left(x_i - \mathbb{E}[x_i]\right)\left(x_j - \mathbb{E}\left[x_j\right]\right)\right]$$

$$= \mathbb{E}\left[\left(x_i - \mathbb{E}[x_i]\right)\left\{\sum_{k \in \mathrm{Pa}_j} w_{jk}\left(x_k - \mathbb{E}[x_k]\right) + \sqrt{\upsilon_j}\,\varepsilon_j\right\}\right] = \sum_{k \in \mathrm{Pa}_j} w_{jk} \operatorname{cov}\left[x_i, x_k\right] + I_{ij}\upsilon_j$$

- To show the 2nd term in the Eq. above, use recursively (until you reach the root node)

$$x_i - \mathbb{E}[x_i] = \sum_{m \in \mathrm{Pa}_i} w_{im}\left(x_m - \mathbb{E}[x_m]\right) + \sqrt{\upsilon_i}\,\varepsilon_i = \sum_{m \in \mathrm{Pa}_i} w_{im}\left(\sum_{n \in \mathrm{Pa}_m} w_{mn}\left(x_n - \mathbb{E}[x_n]\right) + \sqrt{\upsilon_m}\,\varepsilon_m\right) + \sqrt{\upsilon_i}\,\varepsilon_i$$

and apply $\mathbb{E}\left[\varepsilon_i \varepsilon_j\right] = I_{ij}, \mathbb{E}\left[\varepsilon_m \varepsilon_j\right] = 0 \ \textit{for } m < i$

- The covariance can be computed recursively starting from the lowest number node.

  - Roweis, S. and Z. Ghahramani (1999). A unifying review of linear Gaussian models. *Neural Computation* **11**(2), 305–345.

# Linear-Gaussian Models

- *For D-isolated (no links) nodes*, the number of independent parameters in the model are (no $w_{ij}$ in this model):

$$\underset{parameters\ b_i}{D} + \underset{parameters\ \upsilon_i}{D} = 2D$$

The mean of p(x) is $(b_1,\ldots,b_D)^T$ & the variance diag$(\nu_1,\ldots,\nu_D)$.

- *For a fully connected graph* in which each node has *all lower numbered nodes as parents*, the number of parameters is:

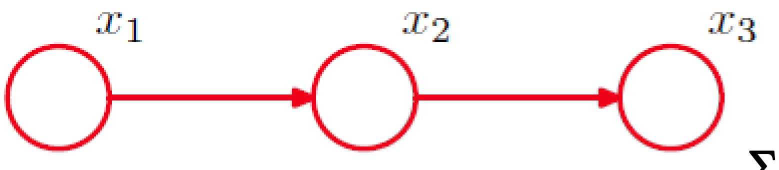
$$\underbrace{\frac{D^2 - D}{2}}_{\substack{parameters\ w_{ij} \\ (lower\ triangle\ matrix)}} + \underset{parameters\ \upsilon_i}{D} = \frac{D(D+1)}{2}$$

- Note that this is the same number of parameters as for a symmetric covariance matrix!

  - Roweis, S. and Z. Ghahramani (1999). A unifying review of linear Gaussian models. *Neural Computation* **11**(2), 305–345.

# Linear-Gaussian Models

- Intermediate levels of complexity can be obtained by making simplifications on the graph. For example, in the graph shown (missing link between nodes $x_1$ and $x_3$):



$$\mathbb{E}[\boldsymbol{x}] = \left(b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1\right)^T$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \upsilon_1 & w_{21}\upsilon_1 & w_{32}w_{21}\upsilon_1 \\ w_{21}\upsilon_1 & \upsilon_2 + w_{21}^2\upsilon_1 & w_{32}\left(\upsilon_2 + w_{21}^2\upsilon_1\right) \\ w_{32}w_{21}\upsilon_1 & w_{32}\left(\upsilon_2 + w_{21}^2\upsilon_1\right) & \upsilon_3 + w_{32}^2\left(\upsilon_2 + w_{21}^2\upsilon_1\right) \end{pmatrix}$$

$$\begin{aligned} \operatorname{cov}[x_1, x_3] &= \sum_{k\in\{2\}} w_{jk} \operatorname{cov}[x_1, x_k] + I_{13}\upsilon_3 \\ &= w_{32}\operatorname{cov}[x_1, x_2] = w_{32}\left(w_{21}\operatorname{cov}[x_1, x_1] + I_{12}\upsilon_2\right) = w_{32}w_{21}\upsilon_1 \end{aligned}$$

- Roweis, S. and Z. Ghahramani (1999). A unifying review of linear Gaussian models. *Neural Computation* **11**(2), 305–345.

# Linear-Gaussian Models

- One can extend these calculations to vector-valued Gaussian nodes.

$$p\left(\boldsymbol{x}_i \middle| \mathrm{Pa}_i\right) = \mathcal{N}\left( \boldsymbol{x}_i ; \sum_{j \in \mathrm{Pa}_i} \boldsymbol{W}_{ij} \boldsymbol{x}_j + \boldsymbol{b}_i , \boldsymbol{\Sigma}_i \right)$$

- The joint distribution is again a Gaussian.

- Roweis, S. and Z. Ghahramani (1999). A unifying review of linear Gaussian models. *Neural Computation* **11**(2), 305–345.

# Directed Gaussian Graphical Models

- Return to the DGM with real-valued variables & CPDs of the form:

$$p(x_t \mid \boldsymbol{x}_{pa(t)}) = \mathcal{N}(x_t \mid b_t + \boldsymbol{w}_t^T \boldsymbol{x}_{pa(t)}, \sigma_t^2)$$

- This is a **linear Gaussian** CPD. Multiplying all these CPDs together results in a joint Gaussian $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$. This is a directed GGM.

- Let us derive $\boldsymbol{\Sigma}$ from the CPD parameters with an easier approach. Subtract the means $\mu_t = \mathbb{E}[x_t] = \sum_{j \in Pa_i} w_{tj}\mathbb{E}[x_j] + b_t$, and write:

$$x_t = \mu_t + \sum_{s \in pa(t)} w_{ts}(x_s - \mu_s) + \sigma_t z_t \ \ where \ z_t \sim \mathcal{N}(0,1),$$

- $\sigma_t$ is the conditional standard deviation of $x_t$ given its parents, $w_{ts}$ is the strength of the $s \rightarrow t$ edge, and $\mu_t$ is the mean.
- The global mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)$ is derived from the resursice relation shown earlier. We now derive the global covariance, $\boldsymbol{\Sigma}$.
- Let $\boldsymbol{S}=diag(\boldsymbol{\sigma})$ be a diagonal matrix containing the standard deviations. We rewrite the Eq. above in matrix-vector form as:

$$\boldsymbol{x} - \boldsymbol{\mu} = \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{\mu}) + \boldsymbol{Sz}$$

- Shachter, R. and C. R. Kenley (1989). Gaussian influence diagrams. *Management Science 35*(5), 527–550 (App. B)

# Directed Gaussian Graphical Models

- Now let **e** be the vector of noise terms: **e** = **Sz,** where $\mathbf{S}=\mathrm{diag}(\boldsymbol{\sigma})$

- We can rearrange $\mathbf{x}-\boldsymbol{\mu}=\mathbf{W}(\mathbf{x}-\boldsymbol{\mu})+\mathbf{Sz}$ to get $\mathbf{e}=(\mathbf{I}-\mathbf{W})(\mathbf{x}-\boldsymbol{\mu})$

- Since **W** is lower triangular ($w_{ts}=0$ if $t>s$ in the topological ordering), **I −W** is lower triangular with 1's on the diagonal. Hence

$$\begin{pmatrix} e_1 \\ e_2 \\ . \\ . \\ e_d \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -w_{21} & 1 & & & \\ -w_{32} & -w_{31} & 1 & & \\ . & . & . & . & . \\ -w_{d1} & -w_{d2} & & -w_{d,d-1} & 1 \end{pmatrix} \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \\ . \\ . \\ x_d-\mu_d \end{pmatrix}$$

- Since **I −W** is always invertible, we can write $\mathbf{x}-\boldsymbol{\mu}=(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}=\mathbf{Ue}=\mathbf{USz}$ where we defined $\mathbf{U}=(\mathbf{I}-\mathbf{W})^{-1}$. Thus the regression weights (via **U**) are connected to the Cholesky decomposition of $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}=\mathrm{cov}[\mathbf{x}]=\mathrm{cov}[\mathbf{x}-\boldsymbol{\mu}]=\mathrm{cov}[\mathbf{USz}]=\mathbf{US}\ \mathrm{cov}\ [\mathbf{z}]\ \mathbf{SU}^T=\mathbf{US}^2\mathbf{U}^T$$

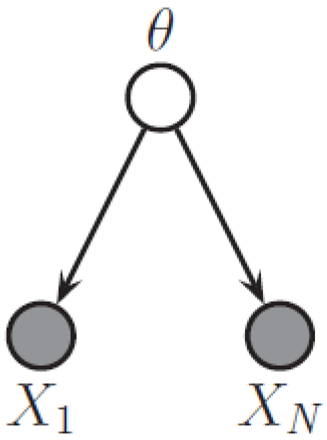
# Plate Notation

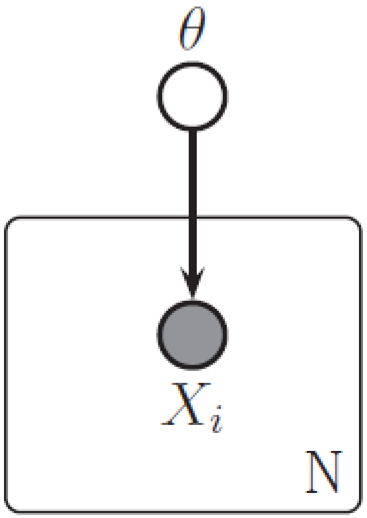
- When inferring $\boldsymbol{\theta}$ from data, we assume the data is iid (but generated from the same distribution). This is represented with the GM shown here.

- *The data cases are independent conditional on the parameters* $\boldsymbol{\theta}$; however, *marginally the data cases are dependent.*

- *The data is exchangeable*: the order in which the data arrive makes no difference to our beliefs about $\boldsymbol{\theta}$: all data orderings result in the same sufficient statistics.

- It is common to use a plate notation as shown:
  - draw a box around the repeated variables
  - nodes within the box get repeated when the model is **unrolled**.
  - write the number of copies in the bottom right corner.

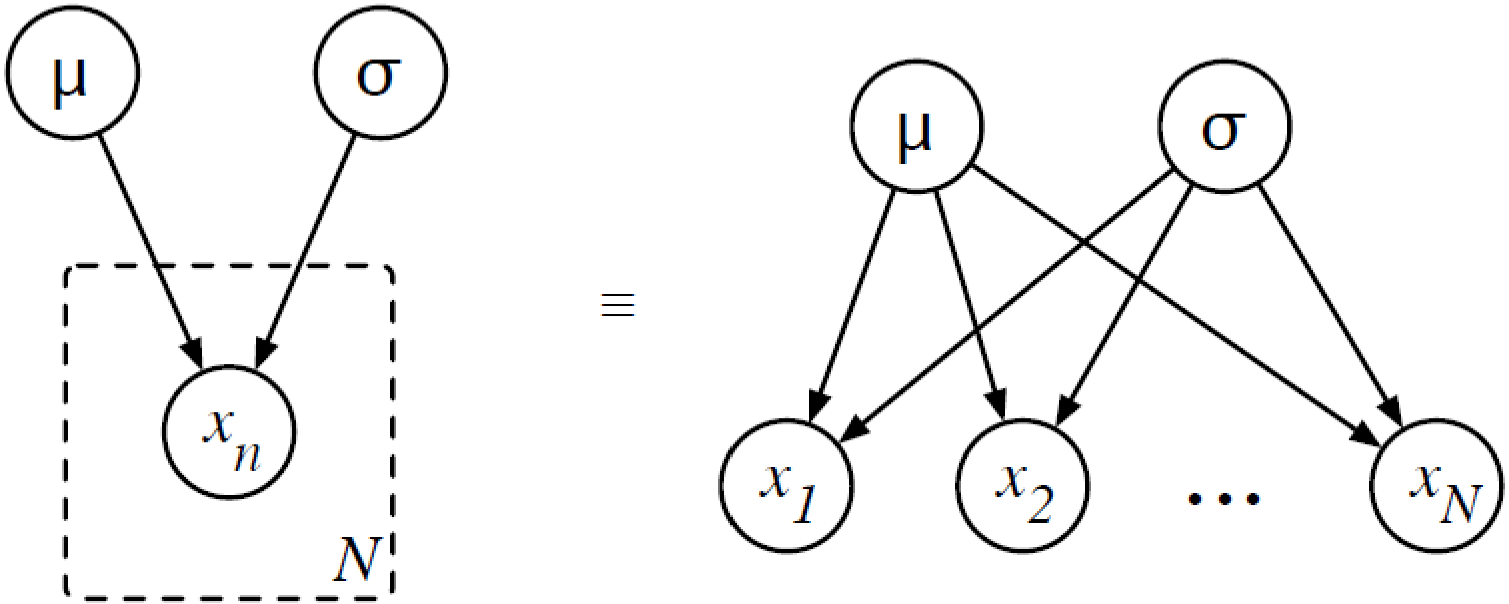




$$p(\boldsymbol{\theta},\mathcal{D}) = p(\boldsymbol{\theta})\prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})$$

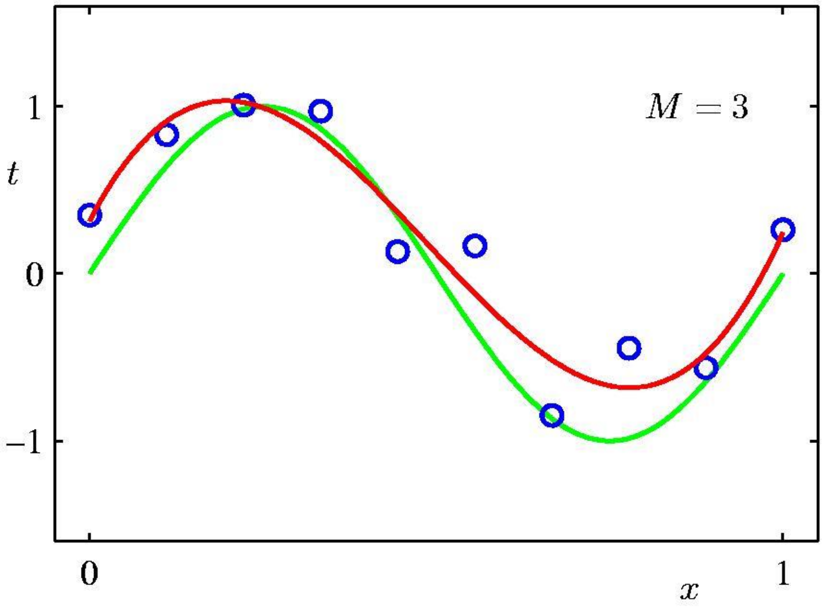
# Plate Notation

- Representation of a *N* points generated from a Gaussian



$$p\left(x_1, x_2, \ldots, x_N, \mu, \sigma\right) = p\left(\mu\right) p\left(\sigma\right) \prod_{n=1}^{N} p\left(x_n \middle| \mu, \sigma\right)$$

# Example: Polynomial Regression



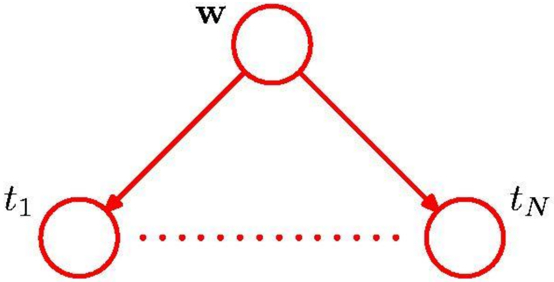
Polynomial regression

$$t = y(x, w) = \sum_{i=0}^{M} w_i x^i$$

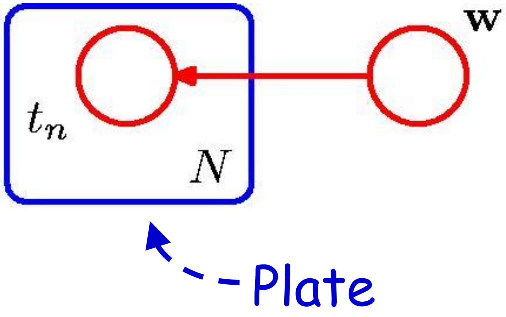
$$p(t, w) = p(w) \prod_{n=1}^{N} p(t_n | w)$$

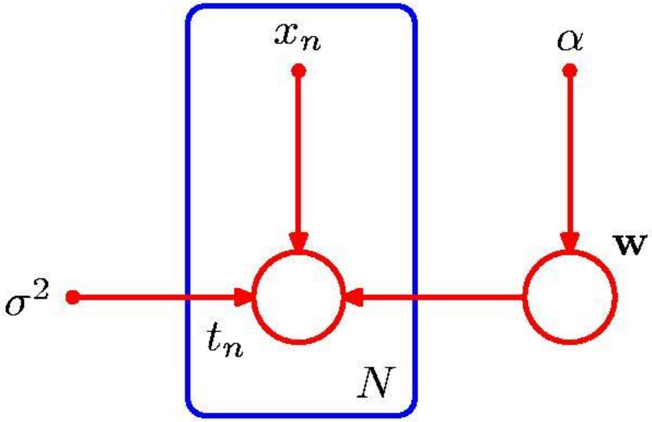
# Example: Polynomial Regression

- Graphical model for the polynomial regression (left). A more compact graphical representation using plate for the polynomial regression (right)



$$p(\mathbf{t},\mathbf{w}) = p(\mathbf{w})\prod_{n=1}^{N} p(t_n|\mathbf{w})$$





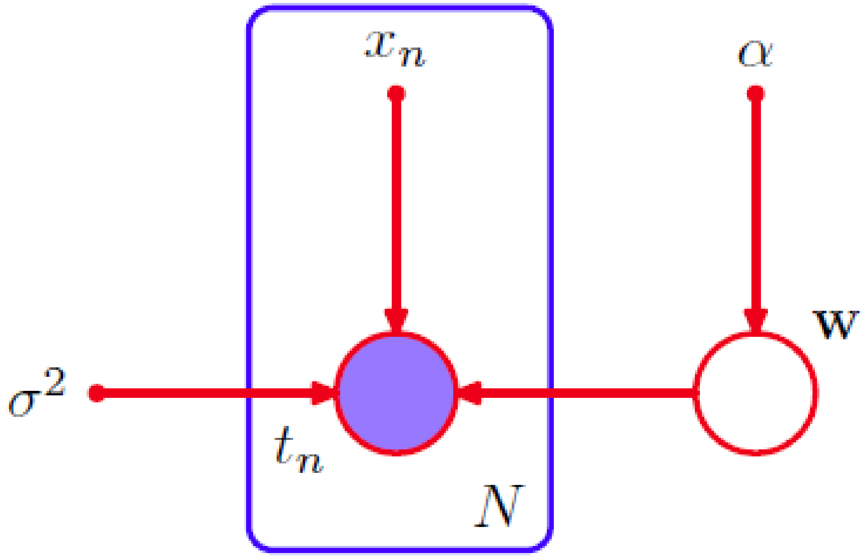
Express the model in an explicit form:

$$p(\mathbf{t},\mathbf{w}|\mathbf{x},\alpha,\sigma^2) = p(\mathbf{w}|\alpha)\prod_{n=1}^{N} p(t_n|\mathbf{w},x_n,\sigma^2)$$

# Example: Polynomial Regression

- Evidence nodes are shaded (observed values). For our regression problem this is shown as follows:

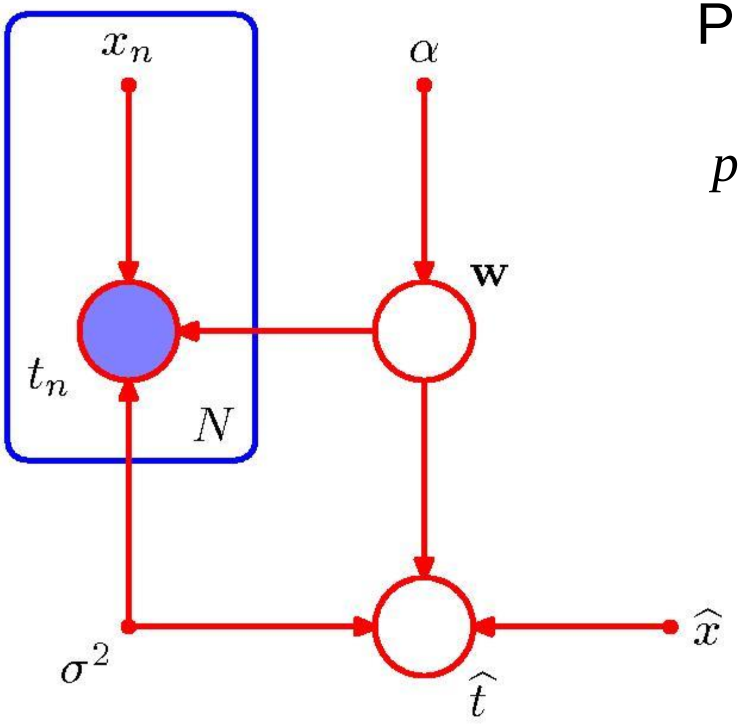
- ***w*** is an example of a latent variable (unobserved)



$$p\left(\mathbf{t},\mathbf{w}\middle|\mathbf{x},\alpha,\sigma^2\right)=p\left(\mathbf{w}\middle|\alpha\right)\prod_{n=1}^{N}p\left(t_n\middle|\mathbf{w},x_n,\sigma^2\right)$$

# Example: Polynomial Regression

- In Bayesian regression, our goal is to make predictions for new input values.



Predictive distribution:

$$p\left(\hat{t}\middle|x,\mathbf{x},\mathbf{t},\alpha,\sigma^2\right)\propto\int p\left(\hat{t},\mathbf{t},\mathbf{w}\middle|x,\mathbf{x},\alpha,\sigma^2\right)d\mathbf{w}$$

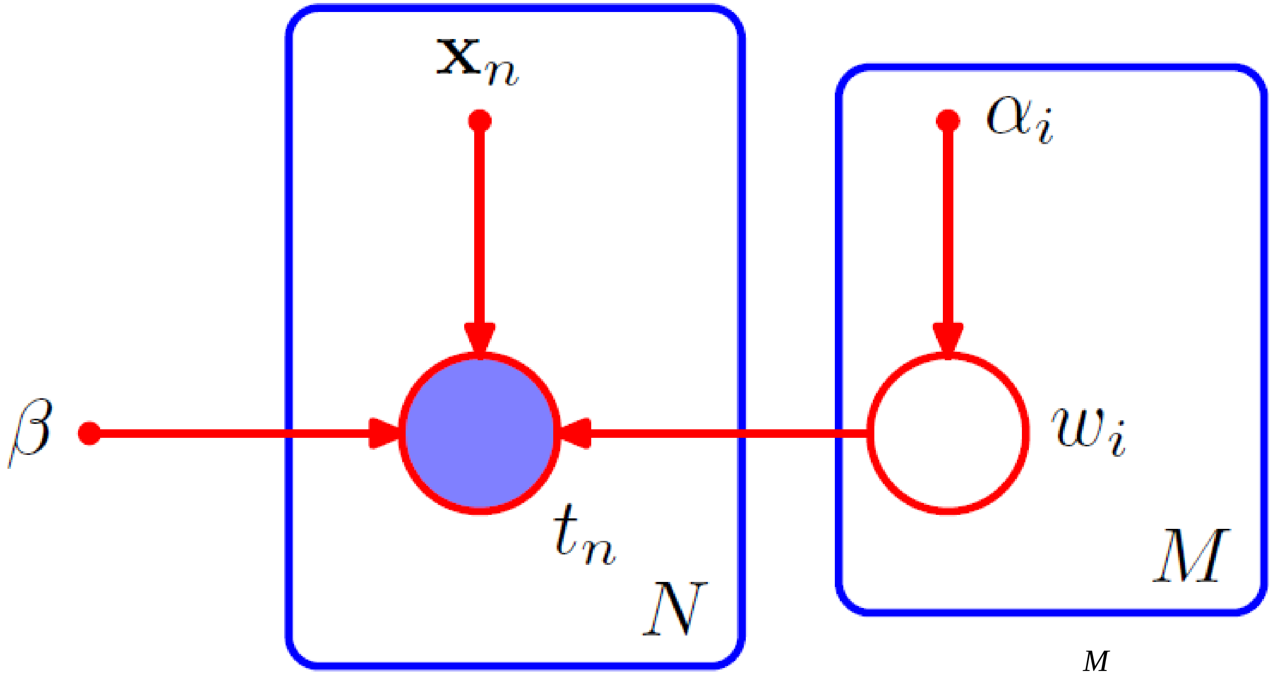
where

$$p\left(\hat{t},\mathbf{t},\mathbf{w}\middle|x,\mathbf{x},\alpha,\sigma^2\right)=$$

$$\left[\prod_{n=1}^{N}p\left(t_n\middle|\mathbf{w},x_n,\sigma^2\right)\right]p\left(\mathbf{w}\middle|\alpha\right)p\left(\hat{t}\middle|x,\mathbf{w},\sigma^2\right)$$

# Example: Relevance Vector Machine

- Recall the RVM (relevance vector machine) framework for regression.
- The graphical model is shown below.

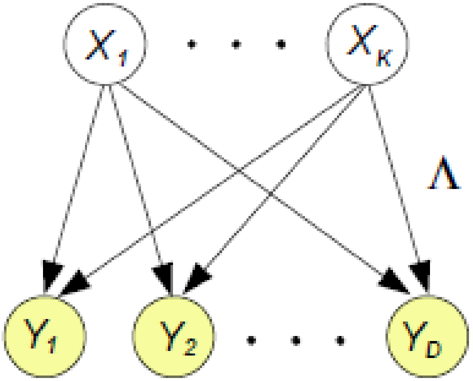


$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} p\left(w_i | 0, \alpha_i^{-1}\right)$$

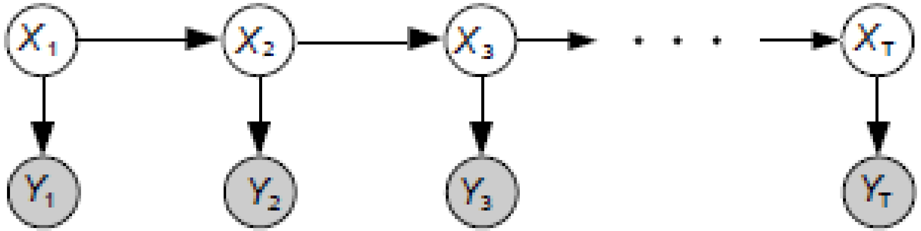
$$p(\mathbf{t}|\boldsymbol{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} p(t_n \,|\, \boldsymbol{x}_n, \mathbf{w}, \beta)$$

# State Space Models

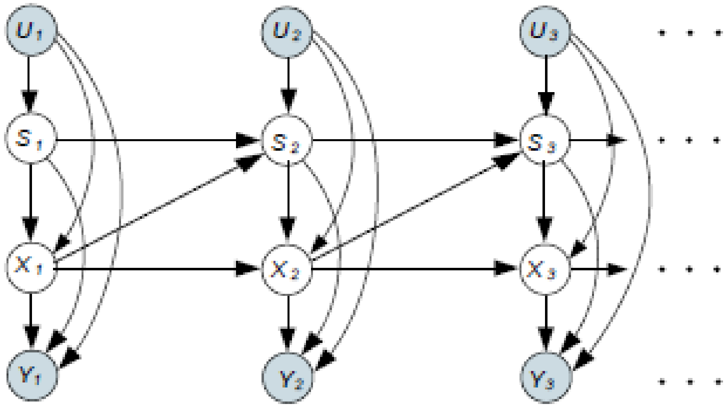
- Probabilistic Principal Component Analysis (PCA)



- Hidden Markov Models

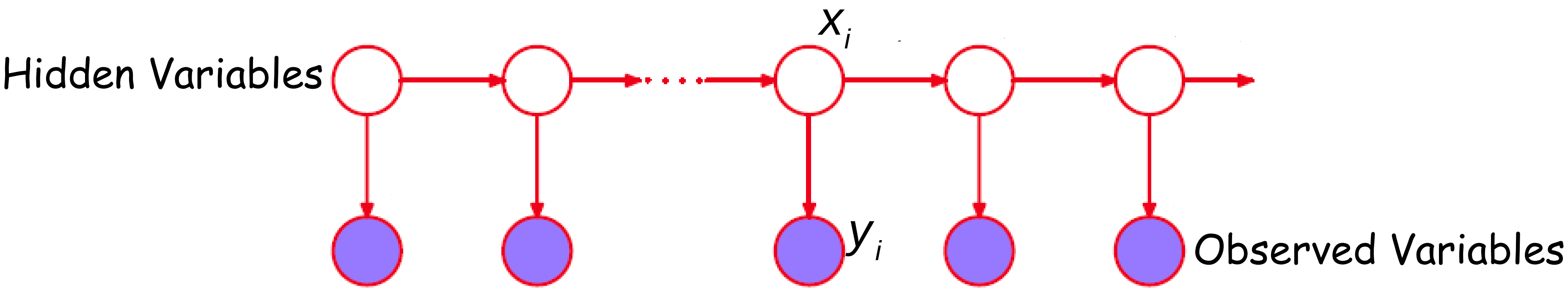


- Switching State-Space Models



# Example in State Space Models

- Consider a Hidden Markov Model (discrete states) or a Gaussian Filter (linear Gaussian Model)

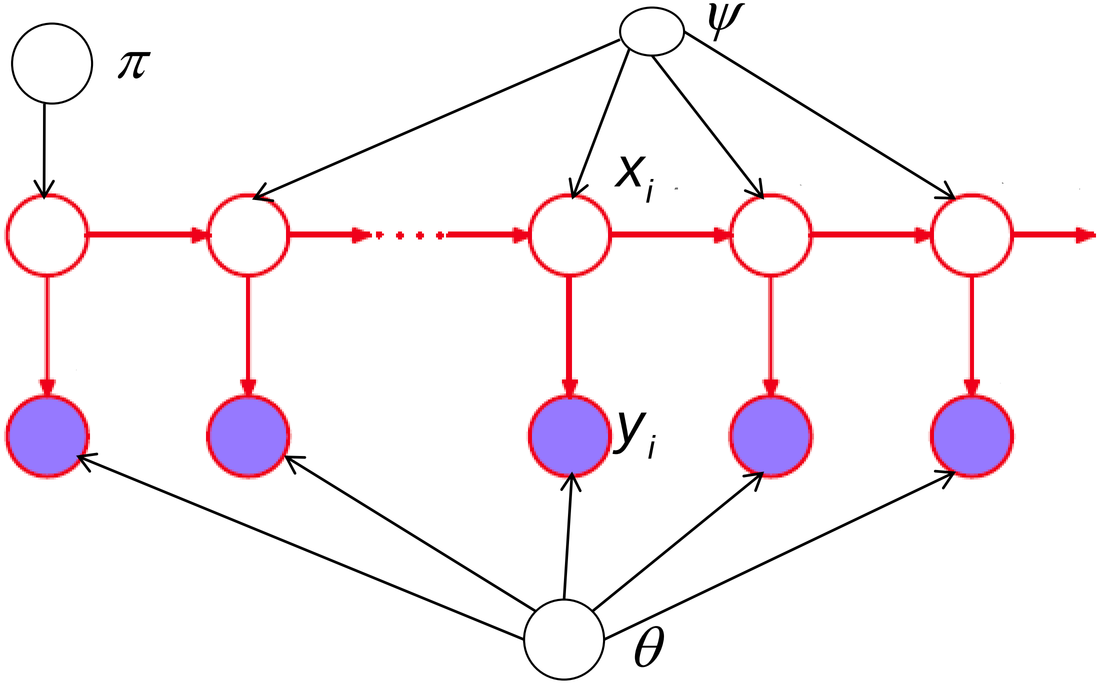


$$p(\mathbf{x},\mathbf{y}) = p(x_1)p(y_1 \mid x_1)p(x_2 \mid x_1)....p(x_i \mid x_{i-1})p(y_i \mid x_i)....$$

- For linear Gaussian model for the conditional distributions, i.e. $p(x \mid z) = \mathcal{N}\left(Az+b,\Sigma\right)$ the model is a Kalman Filter. In this case, the joint distribution over all variables is also a highly structured Gaussian.

# Example: Bayesian State Space Models

- Introduce a prior for $x_1$, and parametric models (parameters $\theta$ and $\psi$ - each with its own prior – are the same for all models):

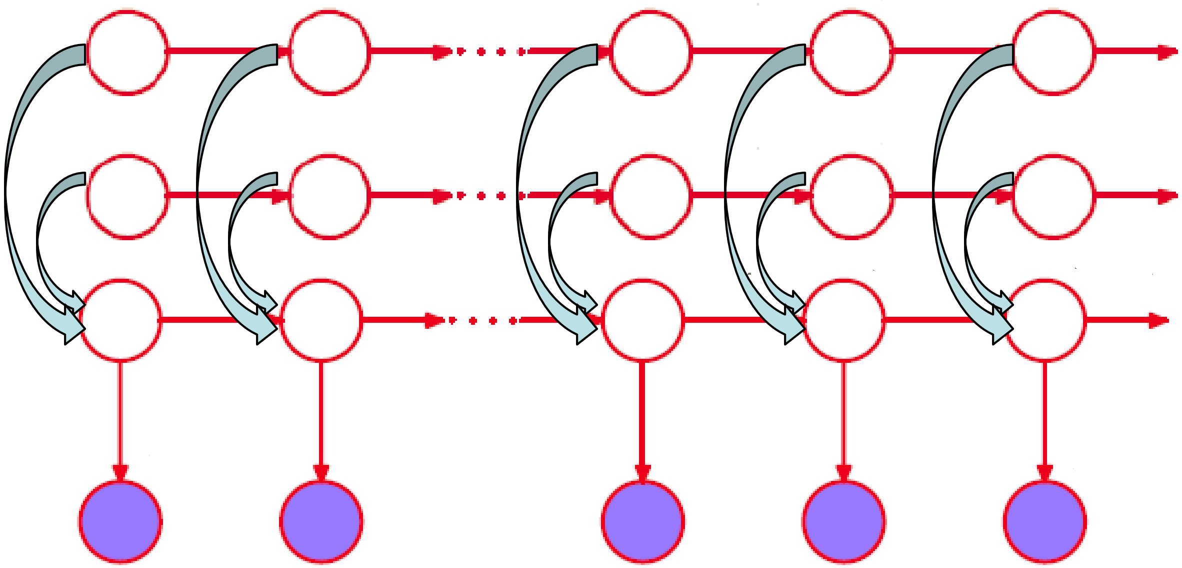


$$....p(x_i \mid x_{i-1};\psi)p(y_i \mid x_i;\theta)....$$

- *In this graph we have loops (not like the tree structure of the HMM model shown earlier). This makes approximate (rather than exact) inference the only option.*

# Example: Factorial State Space Model

- Consider multiple Hidden Sequences (if each state has m configurations, then a total of $m^3$ configurations at each time step)



- *This model is more tractable than a single hidden chain with $m^3$ configurations at each time.*

# Context Specific Independence

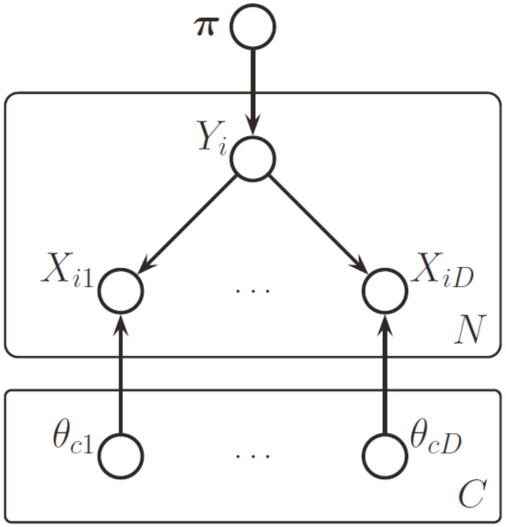
- On the right a naive Bayes classifier has been unrolled for *D* features and uses a plate notation over data $i = 1 : N$. The Fig. on the bottom right shows a *nested plate* notation for the same model.

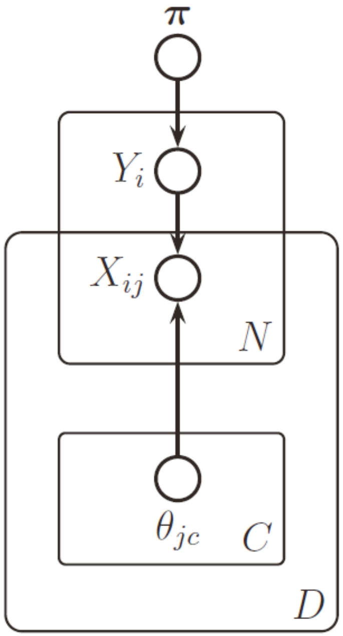
- A variable is inside two plates has two sub-indices (e.g. $\theta_{jc}$ is the parameter for feature *j* in class-conditional density *c*).

- Note that *plates can be nested or crossing.*

- Note that $\theta_{jc}$ is used to generate $x_{ij}$ iff $y_i = c$, otherwise it is ignored (this is certainly not clear from the nested notation).

- This is context specific independence, since the CI $x_{ij} \perp \theta_{jc}$ only holds if $y_i \neq c$.





- Heckerman, D., C. Meek, and D. Koller (2004). Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research.

# Learning from Complete Data

- *If all the variables are fully observed in each data case (no missing data and no hidden variables) we say the data is complete.*

- For a DGM with complete data, the likelihood is given by

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{t=1}^{V} p(x_{it}|\boldsymbol{x}_{i,pa(t)},\boldsymbol{\theta}_t) = \prod_{t=1}^{V} p(\mathcal{D}_t|\boldsymbol{\theta}_t)$$

- $\mathcal{D}_t$ is the data associated with node *t* and its parents (t'th family). This is a *product of terms, one per CPD.* ***The likelihood decomposes according to the graph structure.***

- Assume a prior that factorizes as well: $p(\boldsymbol{\theta}) = \prod_{t=1}^{V} p(\boldsymbol{\theta}_t)$

- Then clearly *the posterior also factorizes:*

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \prod_{t=1}^{V} p(\mathcal{D}_t|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t)$$

- *Thus we **can compute the posterior of each CPD independently**.* More details will be provided in a forthcoming lecture.

# Learning from Complete Data

- As an example consider that all CPDs are tabular. We have a separate row (i.e., a separate multinoulli distribution) for each conditioning case.

- The t'th CPT is $x_t \mid \boldsymbol{x}_{pa(t)} = c \sim \mathcal{C}at\ \ \boldsymbol{\theta}_{tc}$ , where $\sum_k \theta_{tck} = 1$ with:

$$\theta_{tck} = p\ \ x_t = k \mid \boldsymbol{x}_{pa(t)} = c\ \ , k = 1:K_t, c = 1:C_t, t = 1:T\ with\ C_t = \prod_{s \in pa(t)} K_s$$

- Assume a separate Dirichlet prior on each row of each CPT, i.e., $\boldsymbol{\theta}_{tc} \sim \mathcal{D}ir(\boldsymbol{\alpha}_{tc})$. We can compute the posterior by simply adding the pseudo counts to the empirical counts $\boldsymbol{\theta}_{tc} | \mathcal{D} \sim \mathcal{D}ir\ (\mathbf{N}_{tc} + \boldsymbol{\alpha}_{tc})$, where $N_{tck}$ is the number of times that node *t* is in state *k* while its parents are in state *c*:

$$\mathrm{N}_{tck} \triangleq \sum_{i=1}^{N} \mathbb{I}\left(x_{i,t} = k, x_{i,pa(t)} = c\right)$$

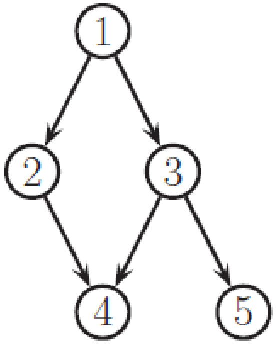
- The mean of this distribution is given by the following:

$$\bar{\theta}_{tck} = \frac{\mathrm{N}_{tck} + \boldsymbol{\alpha}_{tck}}{\sum_{k'} \mathrm{N}_{tck'} + \boldsymbol{\alpha}_{tck'}}$$

# Learning from Complete Data

- Consider the DGM shown, Suppose the training data is:

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |



- Below we list all the sufficient statistics $N_{tck}$, and *the posterior mean parameters* $\bar{\theta}_{tck}$ *under a Dirichlet prior with* $\alpha_{tck}$ *= 1 (add-one smoothing) for the t = 4 node:*

| $x_2$ | $x_3$ | $N_{tck=1}$ | $N_{tck=0}$ | $\bar{\theta}_{tck=1}$ | $\bar{\theta}_{tck=0}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1/2 | 1/2 |
| 1 | 0 | 1 | 0 | 2/3 | 1/3 |
| 0 | 1 | 0 | 1 | 1/3 | 2/3 |
| 1 | 1 | 2 | 1 | 3/5 | 2/5 |

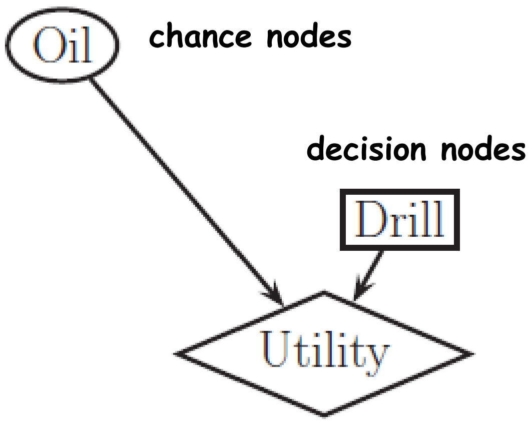
- The *MLE* has the same form without the $\alpha_{tck}$ terms:

$$\hat{\theta}_{tck} = \frac{N_{tck}}{\sum_{k'} N_{tck'}}$$

- The MLE suffers from the zero-count problem so it is important to use a prior.

# Influence Decision Diagrams

- We represent multi-stage decision problems using *decision (influence) diagrams*. [1,2]
- We extend directed graphical models by adding **decision (action) nodes** (rectangles), and **utility (value) nodes** (diamonds). The original random variables (**chance nodes)** represented as ovals.



*oil wild-catter problem*

- In the example shown, we decide whether to drill an oil well or not.

*d = 1 means drill, d = 0 means don’t drill*

- There are 3 states of nature:

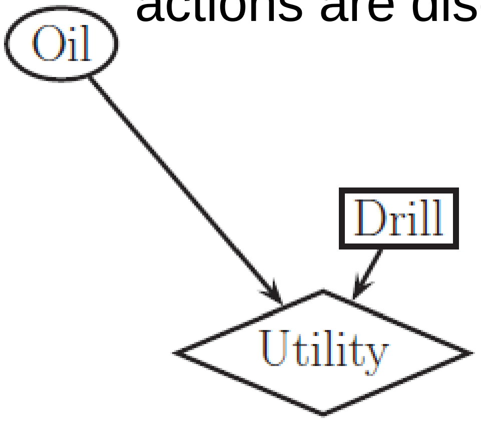
*o = 0 the well is dry, o = 1 it is wet (has some oil), and*
*o = 2 it is soaking (has a lot of oil)*

- Our *prior beliefs* are $p(o) = [0.5, 0.3, 0.2]$.

  - Howard, R. and J. Matheson (1981). Influence diagrams. In R. Howard and J. Matheson (Eds.), *Readings on the Principles and Applications of Decision Analysis, volume II*. Strategic Decisions Group.
  - Kjaerulff, U. and A. Madsen (2008). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer
  - Raiffa, H. (1968). *Decision Analysis*. Addison Wesley.

# Utility Function – Prior Expected Utility

- ❑ You must also specify *the utility function U(d, o).* Since the states and actions are discrete, we can represent it as a table (in $)



| | Well: | Dry, | Wet, | Soaking |
|---|---|---|---|---|
| | | $o=0$ | $o=1$ | $o=2$ |
| Don't drill | $d=0$ | 0 | 0 | 0 |
| drill | $d=1$ | -70 | 50 | 200 |

- ❑ The *prior expected utility if you drill* is given by

$$EU(d=1)=\sum_{o=0}^{2} p(o)U(d,o)=0.5\times(-70)+0.3\times 50+0.2\times 200=20$$

- ❑ The prior expected utility if you don't drill is 0. So the *max prior expected utility is*

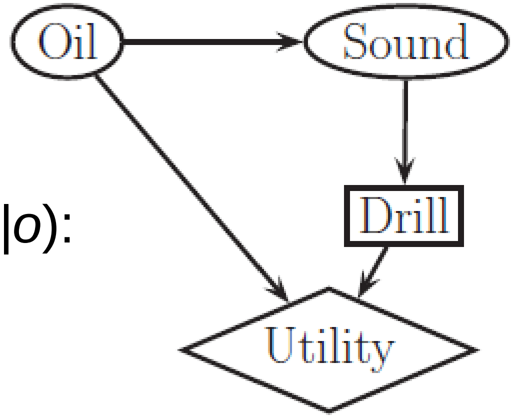
$$MEU=\max\{EU(d=0),EU(d=1)\}=\max\{0,20\}=20$$

and therefore the optimal action is to drill:

$$d^{*}=\operatorname{argmax}\{EU(d=0),EU(d=1)\}=1$$

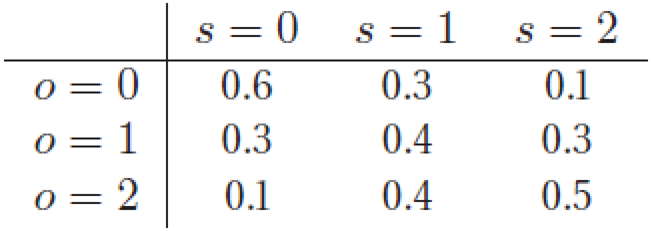
# Reliability of the Sound Test

- Now suppose you perform a sound test to estimate the state of the well. This leads to 3 possible states:
  - $s$ = 0 is a diffuse reflection pattern (no oil)
  - $s$ = 1 is an open reflection pattern, (some oil)
  - $s$ = 2 is a closed reflection pattern (lots of oil)
- Since *S* is caused by *O*, we add an *O* → *S* arc to our model.
- We also add an **information arc** from *S* to *D* since *the sound test will affect our decisions*.



- Consider the following conditional distribution for $p(s|o)$:

reliability of the sound test



|  | $s = 0$ | $s = 1$ | $s = 2$ |
|---|---|---|---|
| $o = 0$ | 0.6 | 0.3 | 0.1 |
| $o = 1$ | 0.3 | 0.4 | 0.3 |
| $o = 2$ | 0.1 | 0.4 | 0.5 |

- Suppose we do the test and observe $s$ = 0. *The posterior over the oil state* is $p(o|s = 0)$ = [0.732, 0.219, 0.049]

# Posterior Expected Utility, Optimal Policy

- The *posterior expected utility of performing action d* is

$$EU(d \mid s=0) \;=\sum_{o=0}^{2} p(o \mid s=0)U(d,\, o)$$

$EU(d = 1|s = 0) = 0.732 \times (-70) + 0.219 \times 50 + 0.049 \times 200 = -30.5$

- However $EU(d = 0|s = 0) = 0$, since not drilling incurs no cost. So *if we observe s = 0, we are better off not drilling.*

- Suppose we observe $s = 1$: $EU(d = 1|s = 1) = 32.9 > EU(d = 0|s = 1) =0$

- Similarly, $EU(d = 1|s = 2) = 87.5 >> EU(d = 0|s = 2) = 0$.

- Hence the optimal policy $d^*(s)$ is as follows:
  - if $s = 0$, $d^*(0) = 0$ and get \$0;
  - if $s = 1$, $d^*(1) = 1$ and get \$32.9; and
  - if $s = 2$, $d^*(2) = 1$ and get \$87.5.

# Maximum Expected Utility

- ❑ Let us consider now different outcomes of the sound test and act optimally on each of them.

- ❑ We can then compute the ***expected profit*** *or maximum expected utility*

$$MEU \ = \sum_{sh=0}^{2} p(s) EU(d^*(s) \mid s),\ p(s) = \sum_{o} p(o) p(s \mid o) = [0.41, 0.35, 0.24]$$

- ❑ *p(s) here is the prior marginal on the outcome of the test.*

- ❑ Hence the maximum expected utility if we do the test is

$$MEU = 0.41 \times 0 + 0.35 \times 32.9 + 0.24 \times 87.5 = 32.2$$

# Accounting for the Cost of the Test

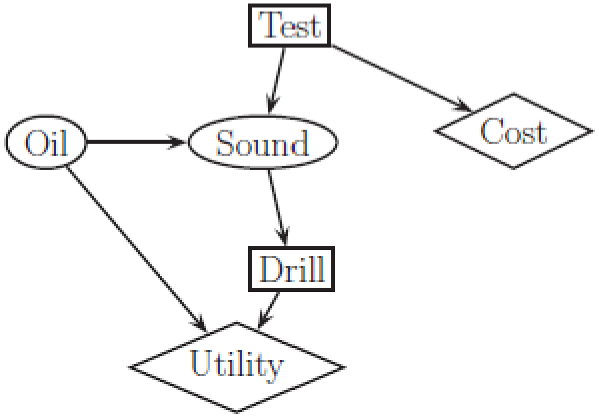
- Now suppose we can choose whether to do the test or not. A modified decision diagram is introduced with a test node *T*.

- If *T* = 1, we do the test, and *S* can enter 1 of 3 states, determined by *O*, exactly as above. If *T* = 0, we don't do the test, and *S* enters a special unknown state.

- There is also some cost associated with performing the test.

- Is it worth doing the test? This depends on how much our MEU changes if we know the outcome of the test (namely the state of *S*).

- If you don't do the test, we have *MEU* = 20 (prior expected utility). If you do the test, you have *MEU* =32.2. So the improvement in utility if you do the test (and act optimally on its outcome) is $12.2 (*value of perfect information, VPI*).



- *DO THE TEST IF IT COSTS LESS THAN $12.2.*

# Value of Perfect Information

- VPI(T) = MEU($I + T \rightarrow D$) − MEU($I$)

  - where $D$ is the decision node
  - T is the variable we are measuring
  - I is the base influence diagram

- One can modify the variable elimination algorithm (to be discussed in a follow up lecture) so that it computes the optimal policy given an influence diagram.

- These methods essentially work backwards from the final time-step, computing the optimal decision at each step assuming all following actions are chosen optimally.[1,2]

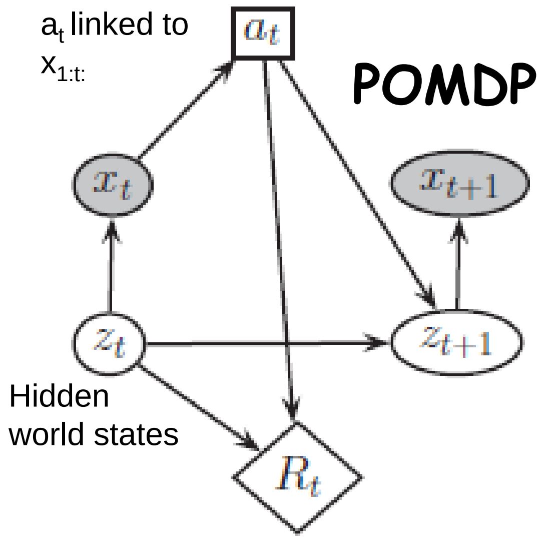
- Kjaerulff, U. and A. Madsen (2008). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis.* Springer
- Lauritzen, S. and D. Nilsson (2001). Representing and solving decision problems with limited information. *Management Science 47*, 1238–1251.

# Partially Observed Markov Decision Process (POM-D-P)

- We could continue to extend the model in various ways. E.g., consider a dynamical system in which we test, observe outcomes, perform actions, move on to the next oil well, and continue drilling  in this way.

- Many problems in robotics, business, medicine, can be usefully formulated as influence unrolled over time.[1,2,3]

- This is known as a *partially observed Markov Process (POMDP, "pom-d-p").*



- *This is an HMM augmented with action and reward nodes.* Can model the *perception-action* cycle that all intelligent agents use.[4]

1. Raiffa, H. (1968). *Decision Analysis*. Addison Wesley.
2. Lauritzen, S. and D. Nilsson (2001). Representing and solving decision problems with limited information. *Management Science 47*, 1238–1251.
3. Kjaerulff, U. and A. Madsen (2008). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis.* Springer
4. Kaelbling, L. P., M. Littman, and A. Cassandra (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence 101*.

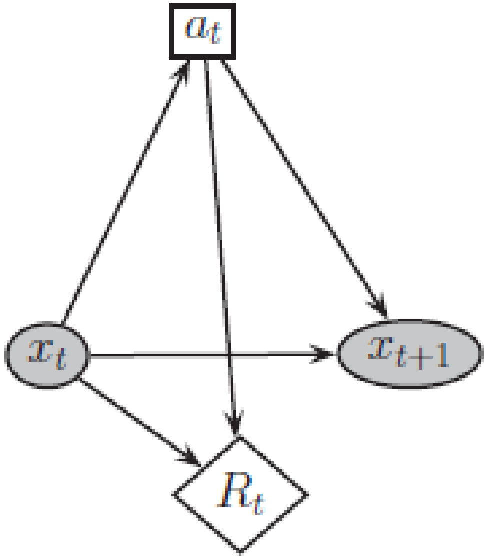
# Markov Decision Process

- A special case of a POMDP, in which the states are fully observed, is called a *Markov Decision Process or MDP.*

- This is easier to solve, since we only *need to compute a mapping from observed states to actions* using dynamic programming.

- In the POMDP case, the information arc from $x_t$ to $a_t$ is not sufficient to uniquely determine the best action, since the state is not fully observed.

- Instead, we need to choose actions based on our *belief state, $p(z_t|x_{1:t}, a_{1:t})$.* Since the belief updating process is deterministic, we can compute a *belief state MDP.*



MDP

- Sutton, R. and A. Barto (1998). *Reinforcment Learning: An Introduction*. MIT Press.
- Kaelbling, L. P., M. Littman, and A. Cassandra (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence 101*.
- Spaan, M. and N. Vlassis (2005). Perseus: Randomized Point-based Value Iteration for POMDPs. *J. of AI Research 24*, 195–220.

# Representation, Inference & Learning

- Representation
  - Undirected graphical models
  - Markov properties of graphs
- Inference
  - Models with discrete hidden nodes
    - ✓ Exact (e.g., forwards backwards for HMMs)
    - ✓ Approximate (e.g., loopy belief propagation)
  - Models with continuous hidden nodes
    - ✓ Exact (e.g., Kalman filtering)
    - ✓ Approximate (e.g., sampling)
- Learning
  - Parameters (e.g., EM)
  - Structure (e.g., structural EM, causality)