Directed Graphical Models

Reference: Machine Learning – A Probabilistic Perspective
by Kevin Murphy

Introduction

- Observe multiple correlated variables
  e.g. words in document, pixels in an image or genes in a microarray

→ Compactly represent the joint distribution $p(\mathbf{x}|\boldsymbol{\theta})$

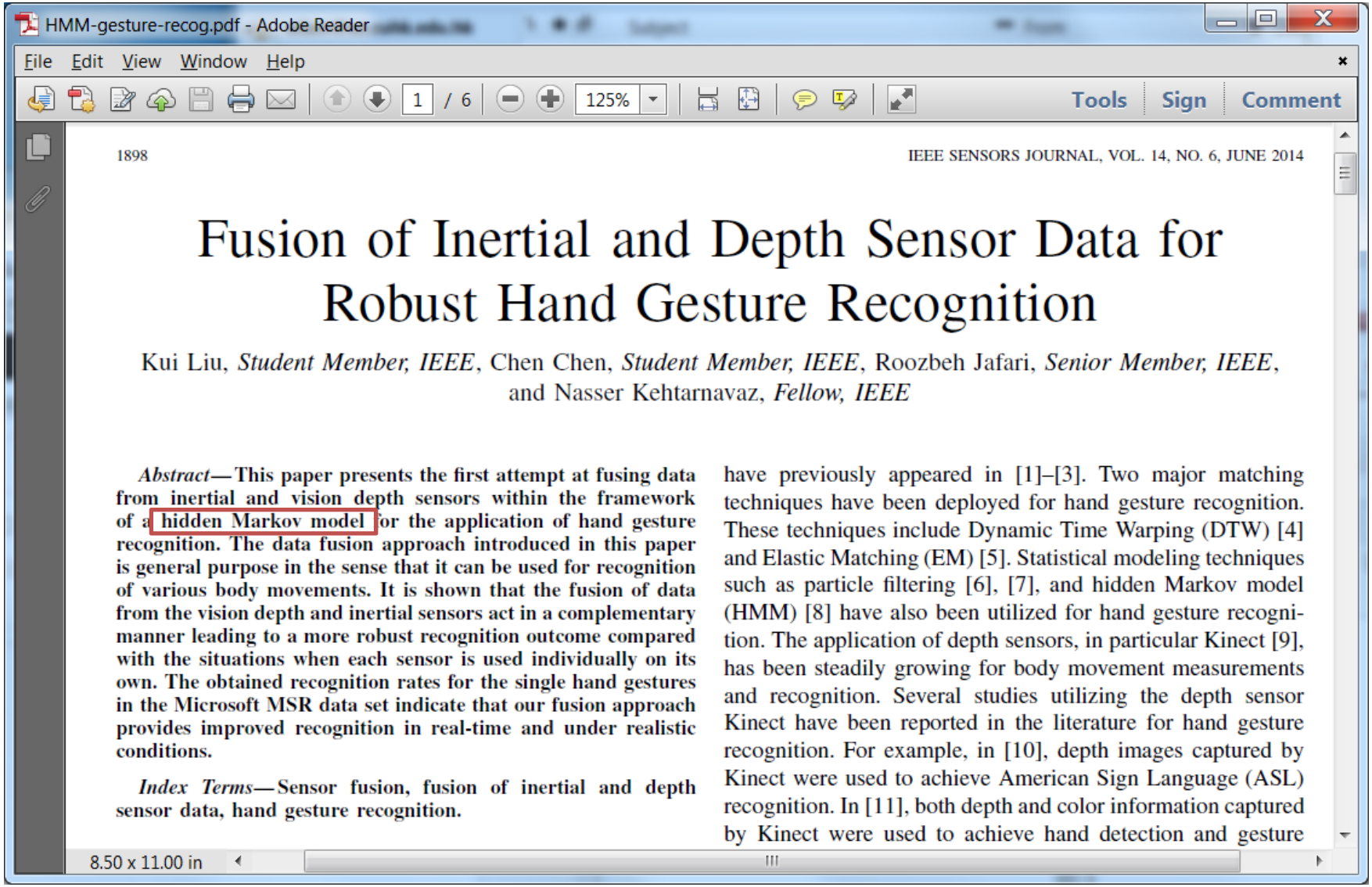
→ Use distribution to infer one set of variables given another in a reasonable amount of computation time

  - Wide range of applications such as recommender models, topic models, etc.

→ Learn the parameters of this distribution with a reasonable amount of data

Applications in VR

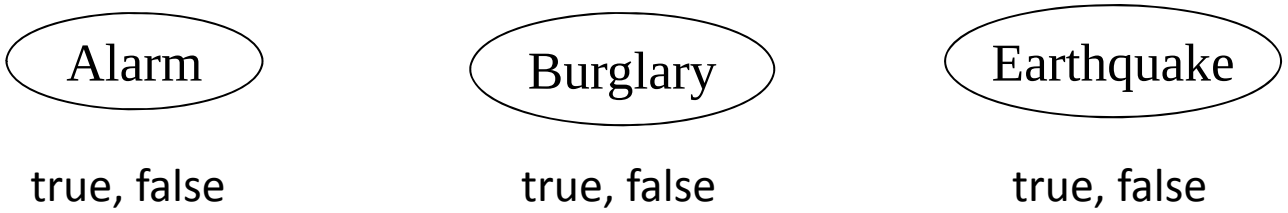
Applications in VR



A set of Variables

Consider a security system of a property asset

There are some variables which can take on binary values



We wish to conduct intelligent reasoning

Chain rule

- By chain rule of probability → represent a joint distribution using any ordering of the variables:
  $p(x_{1:V})$
  $= p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)p(x_4|x_1,x_2,x_3) \dots p(x_V|x_{1:V-1})$
  - where V is the number of variables
  - Matlab-like notation $1{:}V$ denotes the set $\{1, 2, \dots, V\}$
  - Dropped the conditioning on the fixed parameters θ for brevity
- More complicated to represent the conditional distributions $p(x_t|\mathbf{x}_{1:t-1})$ as t gets large

Chain rule

- Suppose all the variables have K states
- Represent $p(x_1)$ as a table of $O(K)$ numbers, representing a discrete distribution
- Represent $p(x_2|x_1)$ as a table of $O(K^2)$ numbers by writing $p(x_2 = j|x_1 = i) = T_{ij}$
- $\mathbf{T}$ is a stochastic matrix:
  - satisfies the constraint $\sum_j T_{ij} = 1$ for all rows i
  - $0 \leq T_{ij} \leq 1$ for all entries
  - Called **conditional probability tables** or CPTs

Chain rule

- There are $O(K^V)$ parameters in the model
- Need an awful lot of data to learn so many parameters
- This model is not useful for other kinds of prediction tasks
- Each variable depends on all previous variables

→ Need another approach

Conditional independence

- Make some assumption about conditional independence (CI) → representing large joint distribution
- X and Y are conditionally independent given Z, denoted $X \perp Y|Z$, if and only if (iff) the conditional joint can be written as a product of conditional marginals:

$$X \perp Y|Z \iff p(X,Y|Z) = p(X|Z)p(Y|Z)$$
$$\iff p(X|Z,Y) = p(X|Z)$$

Conditional independence

- Markov assumption:
  assume that $x_{t+1} \perp x_{1:t-1} | x_t$ (the future is independent of the past given the present)
- Join distribution (Markov assumption + chain rule): $p(\mathbf{x}_{1:V}) = p(x_1) \prod_{t=1}^{V} p(x_t | x_{t-1})$
- Called a (first-order) Markov chain
- Characterized by an initial distribution over states $p(x_1 = i)$,plus a state transition matrix $p(x_t = j | x_{t-1} = i)$

Graphical models

- Define distribution on arbitrary collections of variables
- Represent a joint distribution by making CI assumptions
- The nodes in the graph represent random variables
- The (lack of) edges represent CI assumptions
- Several kinds of graphical model, depending on whether the graph is directed, undirected, or some combination of directed and undirected

Graph terminology

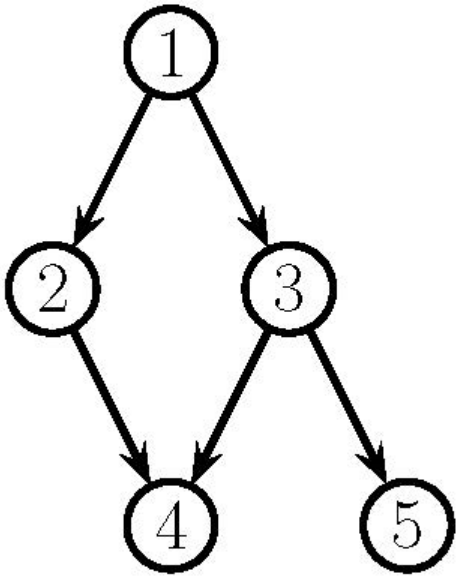
- A graph $G = (\mathcal{V}, \mathcal{E})$ consists of:
  - a set of nodes or vertices $\mathcal{V} = \{1, \ldots, V\}$
  - a set of edges $\mathcal{E} = \{(s, t): s, t \in \mathcal{V}\}$
- Represent the graph by its adjacency matrix: $G(s, t) = 1$ to denote $(s, t) \in \mathcal{E}$, i.e., if $s \rightarrow t$ is an edge in the graph
- Undirected: If $G(s, t) = 1$ iff $G(t, s) = 1$
  Otherwise it is directed
- No self loops: assume $G(s, s) = 0$

Directed graphical models

- Directed graphical model or DGM is a GM whose graph is a DAG
- Known as Bayesian networks
- Also called belief networks – “belief” refers to subjective probability
- Key property of DAGs:
  topological ordering – nodes can be ordered such that parents come before children
  can be constructed from any DAG

Directed Graphical Models

Example



Directed Graphical Models

Example

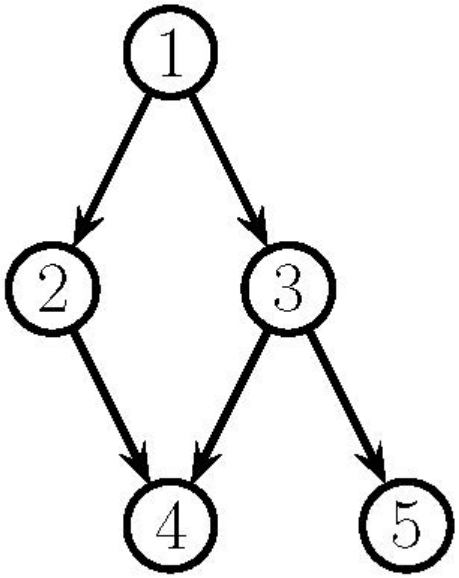
- Define the ordered Markov property: assume that a node only depends on its immediate parents, not on all predecessors in the ordering:

$$x_s \perp \mathbf{x}_{\mathrm{pred}(s)\setminus \mathrm{pa}(s)} | \mathbf{x}_{\mathrm{pa}(s)}$$

- $\mathrm{pa}(s)$ are the parents of node s
- $\mathrm{pred}(s)$ are the predecessors of node s in the ordering
- Natural generalization of the first-order Markov property to from chains to general DAGs

Directed Graphical Models

Example



- Joint distribution:

$$p(\mathbf{x}_{1:5})$$
$$= p(x_1)p(x_2|x_1)p(x_3|x_1, \cancel{x_2})p(x_4|\cancel{x_1}, x_2, x_3)p(x_5|\cancel{x_1}, \cancel{x_2}, x_3, \cancel{x_4})$$
$$= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)p(x_5|x_3)$$

Directed Graphical Models

Example

- In general:

$$p(\mathbf{x}_{1:V}|G) = \prod_{t=1}^{V} p(x_t|\mathbf{x}_{\mathrm{pa}(t)})$$

  where each term $p(x_t|\mathbf{x}_{\mathrm{pa}(t)})$ is conditional probability table (CPT)
- Written the distribution as $p(\mathbf{x}|G)$ → emphasize the equation only holds if the CI assumptions encoded in DAG G are correct
  - Drop this explicit conditioning in later discussions for brevity

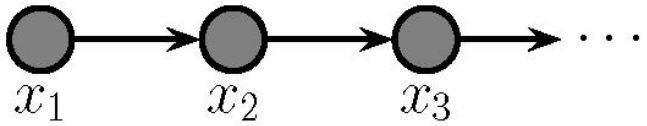
Directed Graphical Models

Example

- If each node has $O(F)$ parents and K states
  → the number of parameters in the model is $O(VK^F)$
- Less than the $O(K^V)$ needed by a model which makes no CI assumptions

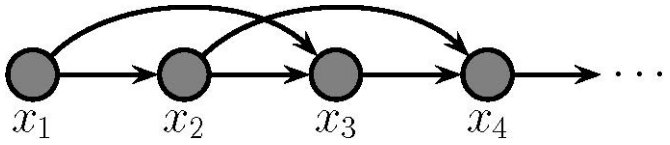
Markov Models

- A first-order Markov chain as a DAG



- The assumption that the immediate past x_{t-1} captures everything we need to know about the entire history

Markov Models



- Second order Markov chain: $x_{1:t-2}$ is a bit strong → relax by adding a dependence from x_{t-2} to x_t
- Corresponding joint has the following form:

$$p(\mathbf{x}_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3) \ldots$$

$$= p(x_1, x_2) \prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2})$$

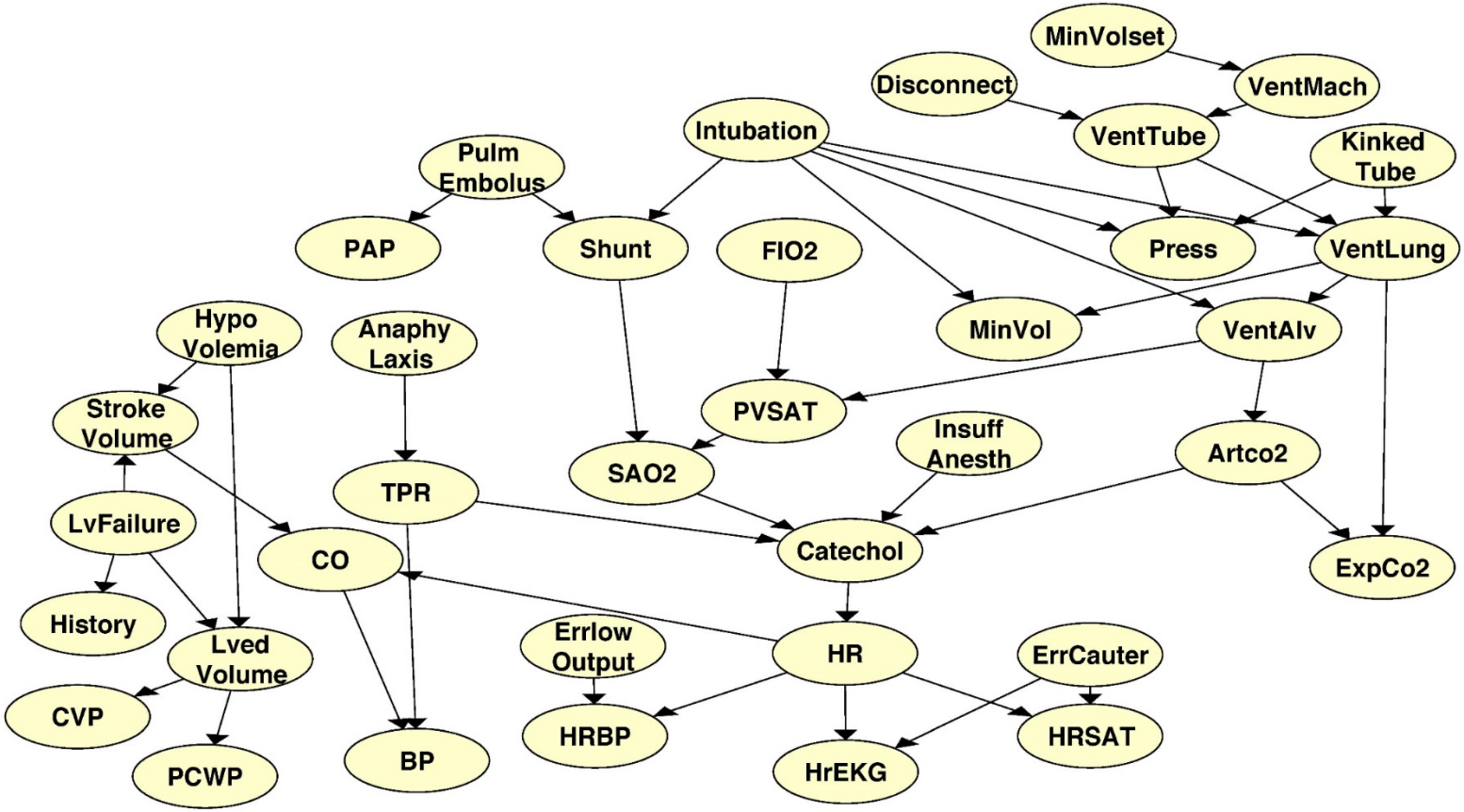
- Can create higher-order Markov models in a similar way

Markov Models

- Even the second-order Markov assumption may be inadequate if there are long-range correlations amongst the observations
- Cannot keep building even higher order models: the number of parameters will blow up

Example in Medical Domain

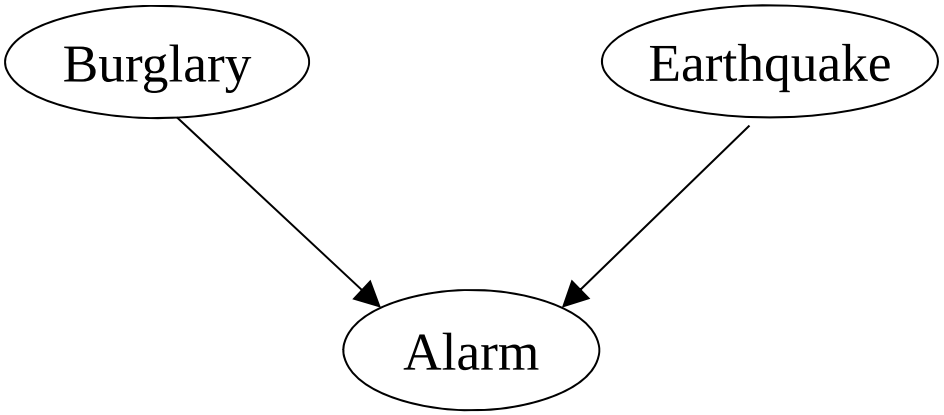
Alarm Network



Modeling relationship among variables in an intensive care unit (ICU)

Simple Example for Security System

Modeling relationship among variables

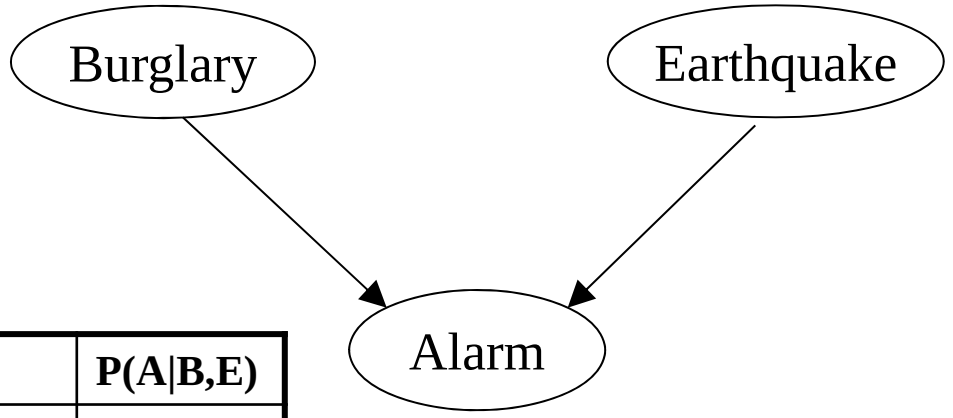


- Intuitively, an arrow from node X to node Y means that X has a direct influence on Y
  - Or X has a casual effect on Y
- Sometimes it is easy for a domain expert to determine these relationships

Directed Graphical Models

CPT Example

B	P(B)
false	0.999
true	0.001



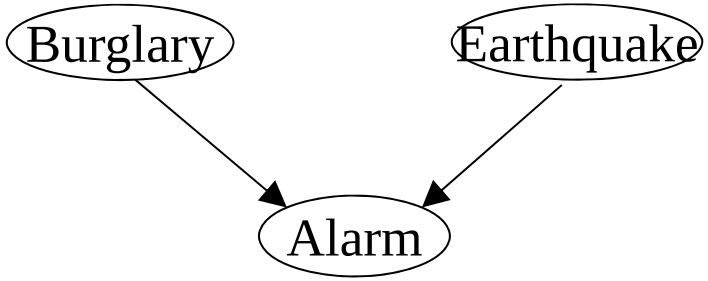
E	P(E)
false	0.998
true	0.002

B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

- Each node has a CPT that quantifies the effect of the parents on the node.
- CPT can be regarded as one kind of parameters.

Directed Graphical Models

CPT Example



- Consider the CPT for the node Alarm (A).
- For a given combination value of the parents (B and E in this example), the entries for P(A=true|B,E) and P(A=false|B,E) must add up to 1.

B	**E**	**A**	**P(A|B,E)**
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

Inference

- Graphical models provide a compact way to define joint probability distributions
- Joint distribution – perform probabilistic inference
- Estimating unknown quantities from known quantities

Inference

- Inference problem: a set of correlated random variables with joint distribution $p(\mathbf{x}_{1:V}|\boldsymbol{\theta})$
- Assuming parameters $\boldsymbol{\theta}$ of the model are known
- Partition this vector into
  - visible variables $\mathbf{x}_v$ (observed)
  - hidden variables $\mathbf{x}_h$ (unobserved)
- Computing the posterior distribution of the unknowns given the knows:

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

Inference

1. Conditioning on the data by clamping the visible variables to their observed values $\mathbf{x}_v$

2. Normalizing, go from $p(\mathbf{x}_h, \mathbf{x}_v)$ to $p(\mathbf{x}_h|\mathbf{x}_v)$

- Probability of the evidence: normalization constant $p(\mathbf{x}_v|\boldsymbol{\theta})$ is the likelihood of the data

Inference

- Only interested in some of the hidden variables
- Partition the hidden variables:
  - query variables $\mathbf{x}_q$ the value wish to know
  - remaining nuisance variables $\mathbf{x}_n$ are no interested in
- Compute the interested variables by marginalizing out the nuisance variables:

$$p(\mathbf{x}_q|\mathbf{x}_v, \boldsymbol{\theta}) = \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_n|\mathbf{x}_v, \boldsymbol{\theta})$$

Inference

- If we have discrete random variables, with K states each, we can perform exact inference in $O(K^V)$ time, where V is the number variables.
- For "tree-like" graph structure, we can perform inference in $O(VK^{w+1})$ where w is related to the treewidth of the graph.
- For more general graphs, exact inference:
  - Can take time exponential in the number of nodes.
  - Complicated to derive

Inference

Monte Carlo Inference

- Approximate inference algorithms are commonly used.
- Generate samples from posterior
  $x^s \sim p(x|D)$ where D is the data
- Then use it to compute any quantity of interest such as:
  - posterior marginal $p(x_1|D)$
  - posterior predictive $p(y|D)$
- The above quantities can be approximated by:

$$\mathbb{E}[f|D] \approx \frac{1}{S}\sum_{s=1}^{S} f(x^s)$$

  for some suitable function f

Inference

Monte Carlo Inference

- By generating enough samples, we can achieve any desired level of accuracy we like.
- The main issue: how do we efficiently generate samples from a probability distribution, particularly in high dimensions?
  - Non-iterative methods

    e.g. Importance Sampling
  - Markov Chain Monte Carlo (MCMC) produces dependent samples

    e.g. Gibbs Sampling

Learning

- In graphical models literature, distinguish between inference and learning
- Inference:
  computing (functions of) $p(\mathbf{x}_h|\mathbf{x}_v, \boldsymbol{\theta})$, where
  - v are the visible nodes
  - h are the hidden nodes
  - $\boldsymbol{\theta}$ are the parameters of the model (assume to be known)

Learning

- Learning:
  Given a set of training data of N records (cases), we need to compute a MAP estimate of the parameters given data

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \log p\left(\mathbf{x}_{i,v} \middle| \boldsymbol{\theta}\right) + \log p(\boldsymbol{\theta})$$

  where $\mathbf{x}_{i,v}$ are the visible variables in case i

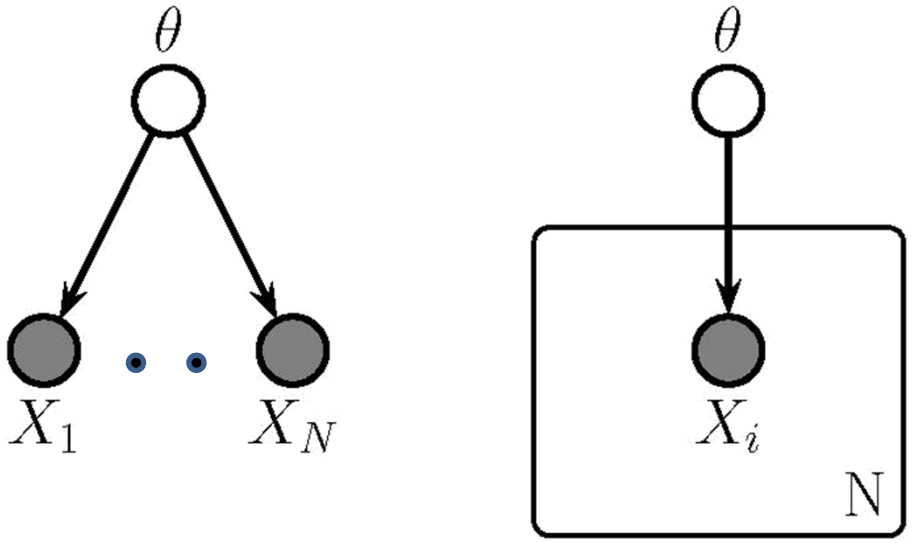
  - Uniform prior $p(\boldsymbol{\theta}) \propto 1$ → reduces to the Maximum Likelihood as usual

Graphical Models

- If we adopt a Bayesian view,
  - we can model the parameters as unknown variables (nodes)
  - then infer the values (similar to inference)
- Under this modeling, the number of hidden variables grows with the amount of training data
- Inferring parameters from data: assume the data is iid

Graphical Models

- Represent using a graphical model:



- Left – data points x_i are conditionally independent given $\boldsymbol{\theta}$
- Right – plate notation (same model as left) repeated x_i nodes are inside a box (plate) number in lower right hand corner N, specifies the number of repetitions of the X_i node

Graphical Models

- Assume each data case was generated independently but from the same distribution
- Data cases are only independent conditional on the parameters $\boldsymbol{\theta}$
- Marginally, the data cases are dependent
- The order in which the data cases arrive makes no difference to the benefits about $\boldsymbol{\theta}$
  (all orderings have same sufficient statistics)
  → data is exchangeable

Plate Notation

- Avoid visual clutter:
  use a form of syntactic sugar, called plates
- Draw a little box around the repeated variables
- With the convention that nodes within the box is repeated when the model is unrolled
- Bottom right corner of the box: number of copies or repetitions
- The corresponding joint distribution has the form:

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

Learning from Complete Data

- If all variables are fully observed in each data instance (case):
  - No missing data and there are no hidden variables
  - The data is *complete*
- The likelihood is:

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

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

where $\mathcal{D}_t$ is the data associated with node t and its parents. This is a product of terms, one per CPD.

Learning from Complete Data

- The likelihood *decomposes* according to the graph structure.
- The likelihood factorizes

Learning with Missing Data / Latent Variables

- If we have missing data and/or hidden variables, the likelihood no longer factorizes.
- We can only compute a locally optimal Maximum Likelihood / MAP estimate

Conditional Independence Properties of DGMs

- Any graphical model is a set of conditional independence (CI) assumptions
- $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_C$: if A is independent of B given C in the graph G
- Using the semantics to be defined below

Conditional Independence Properties of DGMs

- Let $I(G)$ be the set of all such CI statements encoded by the graph
- G is an I-map (independence map) for p, or p is Markov wrt G, iff $I(G) \subseteq I(p)$
  where $I(p)$ is the set of all CI statements that hold for distribution p
- The graph is an I-map if it does not make any assertions of CI that are not true of the distribution

Conditional Independence Properties of DGMs

- Allow to use the graph as a safe proxy for p when reasoning about p's CI properties
- Helpful for designing algorithms that work for large classes of distributions, regardless of their specific numerical parameters $\boldsymbol{\theta}$
- Fully connected graph is an I-map of all distributions: makes *no* CI assertions at all since it is not missing any edges

Conditional Independence Properties of DGMs

- G is a minimal I-map of p if:
  - G is an I-map of p
  - there is no $G' \subseteq G$ which is an I-map of p
- Specify how to determine if $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_C$
- Easy to derive these independencies for undirected graph
- DAG situation is complicated, because of the need to respect the orientation of the directed edges

d-separation

- There is a general topological criterion called **d-separation**
- d-separation determines whether a set of node X is independent of another set Y given a third set E.

d-separation

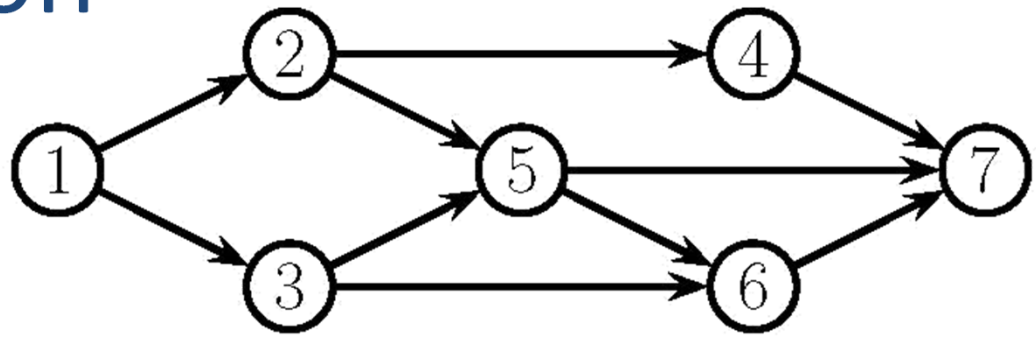
- An undirected path P is **d-separated** by a set of nodes E (containing the evidence) iff at least one of the following conditions hold:

1. P contains a chain, $s \rightarrow m \rightarrow t$ or $s \leftarrow m \leftarrow t$, where $m \in E$
2. P contains a tent or fork, ${}_{s}\swarrow^{m}\searrow_{t}$, where $m \in E$
3. P contains a collider or v-structure, ${}^{s}\searrow_{m}\swarrow^{t}$, where m is not in E and nor is any descendant of m

d-separation

- A set of nodes A is d-separated from a different set of nodes B given a third observed set E iff each undirected path from every node $a \in A$ to every node $b \in B$ is d-separated by E
- Define the CI properties of a DAG:
  $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_E \Leftrightarrow A$ is d-separated from B given E

d-separation



- We can conclude that $x_2 \perp x_6 | \{x_1, x_5\}$ since:
  - $2 \rightarrow 5 \rightarrow 6$ path is blocked by x_5
  - $2 \rightarrow 1 \rightarrow 3 \rightarrow 6$ path is blocked by x_1
  - $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path contains a v-structure at x_7 and x_7 is not in the given (observed) set; therefore blocked by x_7
- $x_2 \not\perp x_6 | \{x_1, x_5, x_7\}$ since $2 \rightarrow 4 \rightarrow 7 \rightarrow 6$ path no longer blocked by x_7

Explaining Away

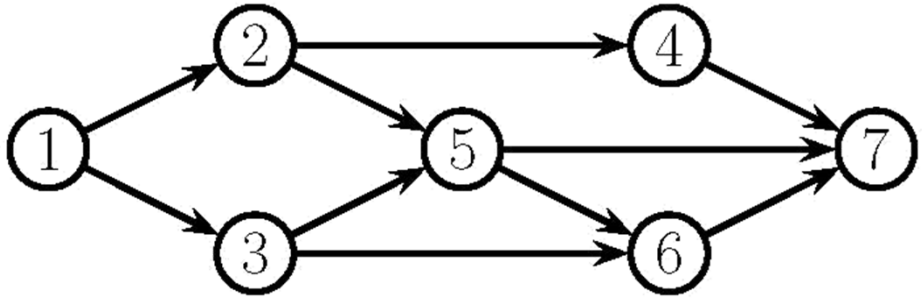
- The v-structure $^{s}\searrow_{m}\swarrow^{t}$ has the effect of *explaining away*.
  - Also called inter-causal reasoning or Berkson's paradox
- s and t are marginally independent
- Conditioning on a common child, i.e. m, its parents, i.e. s and t become dependent.
- As an example, suppose we toss two coins representing binary numbers 0 and 1, and we observe their sum.
  - Suppose that we observe the sum is 1:
  - If the first coin is 0, then we know the second coin is 1

Markov Blanket and Full Conditionals

- t's Markov blanket $\mathrm{mb}(t)$:
  the set of nodes that renders a node t conditionally independent of all the other nodes in the graph
- Markov blanket of node in a DGM is equal to the parents, the children, and the co-parents
  i.e., other nodes who are also parents of its children:

$$\mathrm{mb}(t) \triangleq \mathrm{ch}(t) \cup \mathrm{pa}(t) \cup \mathrm{copa}(t)$$

Markov Blanket and Full Conditionals



- $\mathrm{mb}(5) \triangleq \{6,7\} \cup \{2,3\} \cup \{4\} = \{2,3,4,6,7\}$
  where 4 is a co-parent of 5 because they share a common child, namely 7

Markov Blanket and Full Conditionals

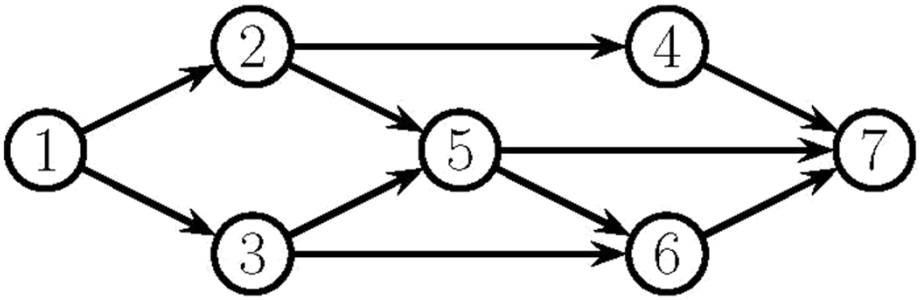
- Co-parents are in the Markov blanket
- When we derive

$$p(x_t|\mathbf{x}_{-t}) = p(x_t, \mathbf{x}_{-t})/p(\mathbf{x}_{-t})$$

- all terms do not involve x_t will cancel out between numerator and denominator
→ left with a product of CPDs which contain x_t in their scope

- $p(x_t|\mathbf{x}_{-t}) \propto p\left(x_t|\mathbf{x}_{\mathrm{pa}(t)}\right) \prod_{s\in \mathrm{ch}(t)} p\left(x_s|\mathbf{x}_{\mathrm{pa}(s)}\right)$

Markov Blanket and Full Conditionals



- $p(x_5|\mathbf{x}_{-5}) \propto$
  $p(x_5|x_2, x_3)p(x_6|x_3, x_5)p(x_7|x_4, x_5, x_6)$
- Resulting expression: *t*'s *full conditional* which is useful for Gibbs sampling