EE613

Machine Learning for Engineers

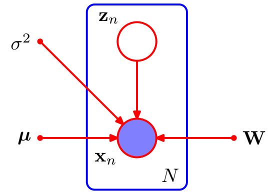
Generative models. Introduction to Graphical models

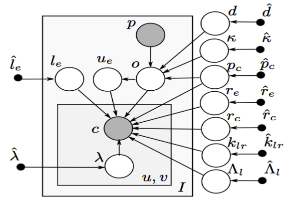
jean-marc odobez

2019

overview

- Graphical models fundamentals
  - bayesian networks, representations
  - conditional independence
  - undirected graphical models
- Learning
  - ML, MAP, Bayesian
  - the EM algorithm, latent variable models
    - Gaussian Mixture Model (GMM)
    - Hidden Markov Model (HMM)
- PCA, Probabilistic PCA
- Inference algorithms





resources

- **textbooks**
  - c. bishop,pattern recognition and machine learning, springer, 2006

    => will mainly follow this book
  - d. mackay, information theory, inference, and learning algorithms cambridge univ. press, 2003
  - m. i. jordan, (ed.), learning in graphical models, mit press, 1998
  - d. barber, Bayesian reasoning and machine learning, Cambridge university press

- **other tutorials**
  - plenty of tutorials and course materials available online (read the textbooks!)

why probabilistic models ?

X

male	10	3.3
female	45	4.6
female	78	4.2
male	30	3.8

Y

A
B
B
A

Model

X_{test}

male	10	3.3

$\hat{Y}$

?

- **Machine learning (supervised)**
  - function estimation y=f(x)
  - output is uncertain: formulate problem as p(y|x)
  - goal: estimate conditional density
    - integrate model uncertainties (e.g. from available training data)
    - interest in knowing the confidence of the decision

why probabilistic models ?

X

low	1	3.3
high	45	4.6
high	22	4.2
med	3	3.8

Model

$\hat{X}$

med	12	2
?	?	6

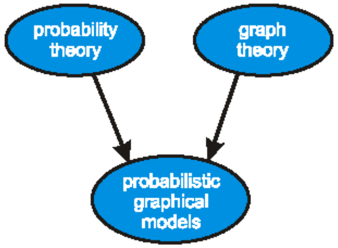
- **Unsupervised learning**
  - density estimation, outlier detection
  - knowledge discovery
  - predict x_i given X (partial observations)

need joint probability model p(x)
(x is often multivariate => interest in defining/finding structure and relations)

Bayesian network: representation

probabilistic graphical models

- **graphical representations of probability distributions**
    - a marriage between probability theory and graph theory
    - visualization of the structure of probability distributions
    - new insights into existing models (e.g. conditional independence)
    - computation (learning and inference) using graph-based algorithms



- several well known algorithms have a probabilistic version, eg.
    - kmeans -- GMM
    - PCA -- Probabilistic PCA
    - LSI or Non-negative Matrix Factorization -- PLSA (or LDA) ...

representing joint distributions

$$p(x_1, \ldots, x_K) \text{ = ?}$$

$$p(x_1, \ldots, x_K) = p(x_K | x_1, \ldots, x_{K-1}) \ldots p(x_2 | x_1) p(x_1)$$

- **Chain rule of probability**
    - no information on variable dependencies
    - number of parameters can be high
      => $p(x_i \mid x_{1:i-1})$ requires $O(L^i)$ parameters if there are L states per variable

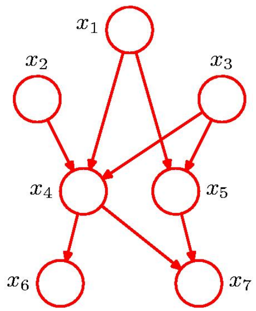
- **Objective**

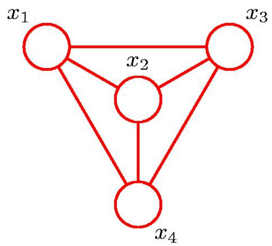
  define Conditional independence (CI) assumptions to simplify distributions

representing joint distributions with graph

- **nodes**
  - subsets of random variables (RVs)
  - discrete or continuous
- **vertices**
  - relations between RVs
  - directed (Bayes net) or
  - undirected (Markov Random Fields)

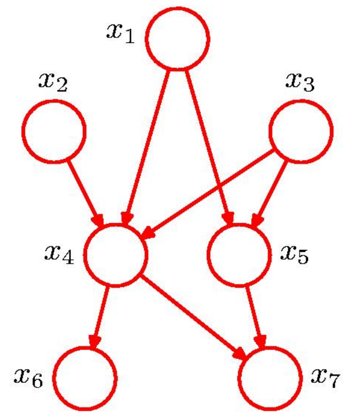
$$p(x_1, \ldots, x_K) \quad = ?$$





bayesian networks (BNs): directed graphical models

- **directed acyclic graphs (DAG)**
  - no closed paths within the graph $\longrightarrow$ we can't go from node to node along vertices on the direction of the arrows and end up at the original node
  - nodes have 'parents' and 'children'
  - x_4 is a child of $x_1 x_2$, x_3 and is a parent of x_6
  - x_1 has no parents



no directed cycles

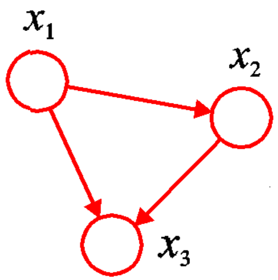
BNs: decomposition

- take any joint distribution over three variables

$$\mathbf{x} = x_{1:3} = (x_1, x_2, x_3)$$

- by applying the product rule

$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2, x_3|x_1) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \end{aligned}$$



symmetrical w.r.t. the three variables

not symmetrical

by choosing a different ordering we would get a different representation and a different graph

BNs: the 'canonical' equation (1)

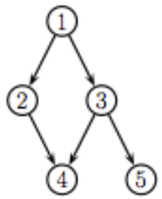
- Graph is a DAG
  - order nodes such that parents come before children (ancestral ordering)
- Ordered Markov property
  - assumption: a node only depends on its immediate parents (pa), not on all predecessor (pred) in the ordering

$$p(x_i|x_{\mathrm{pred}_i}) = p(x_i|x_{\mathrm{pa}_i})$$

  - generalization of first-order Markov property from chains to general DAGs

Consequence on joint distribution

BNs: the ‘canonical’ equation (2)



chain rule +
/ : simplifications due to ordered Markov property assumption

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

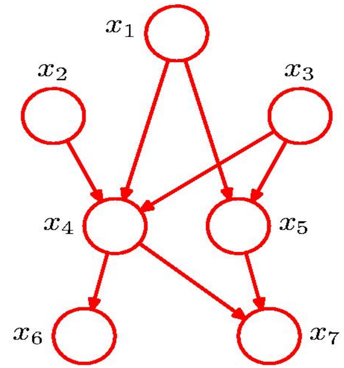
BNs: the ‘canonical’ equation (3)

- joint distribution for $\mathbf{x} = \{x_1, \ldots, x_L\}$

$$p(\mathbf{x}) = \prod_{k=1}^{L} p(x_k|\mathrm{pa}_k)$$

pa_k : set of parents of x_k

- **factorized representation**: product of ‘local’ conditional distributions



$p(\mathbf{x}) =$.

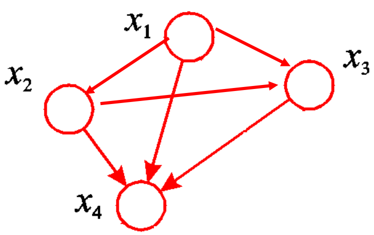
BNs: the importance of being absent

- applying the product rule to **any** joint distribution of K variables

$$p(x_1, \dots, x_K) = p(x_K | x_1, \dots, x_{K-1}) \dots p(x_2 | x_1) p(x_1)$$

produces a fully-connected graph (a link between every pair of nodes)

- the **absence** of links conveys relevant information about the properties of the distributions represented by a graph



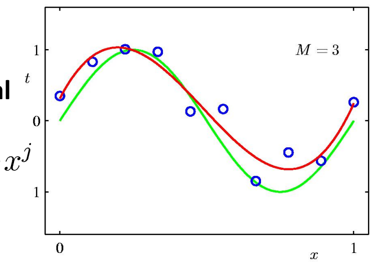
- many real problems can be modeled by ‘sparse’ links (conditional dependencies) among the variables

- it facilitates computation

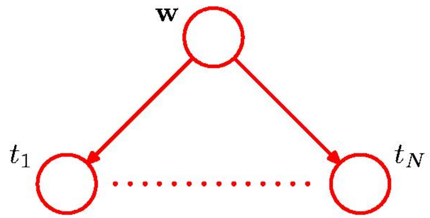
BNs: the plate notation

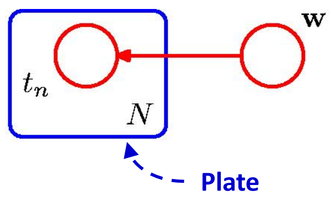
Polynomial

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j$$



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

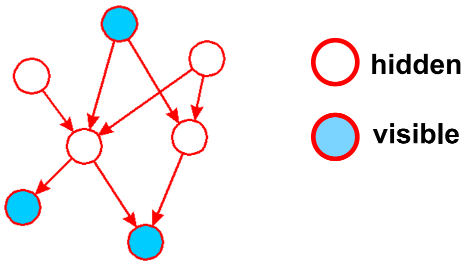




- t : predicted value from x (given) w weight
- t_i : drawn i.i.d from the model
- to avoid visual clutter, we draw a box around the repeated variables, and show the number of nodes of the same kind on bottom right

BNs: types of variables

- variables may be **hidden** (latent) or **visible** (observed)

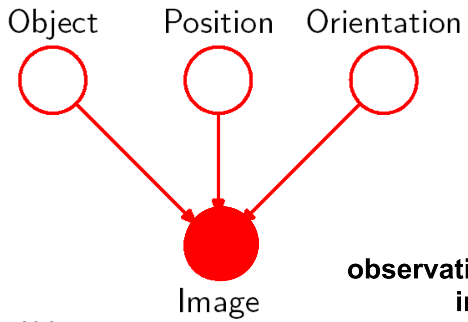


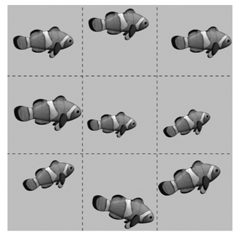
- visible variables: evidence
  - observations (known/given, e.g. physical measurements)
- latent variables (hidden, unknown)
  - included to define richer models
  - often have a clear (physical) interpretation
- depending on the problem, a variable might be observed or hidden (class: known during training, unknown at test time)

BNs: generative models

- a BN can be interpreted as expressing the processes by which the observations were generated (i.e sampled)
- example

object, discrete variable: i
position, continuous variable: (x,y)
orientation, continuous variable: Θ



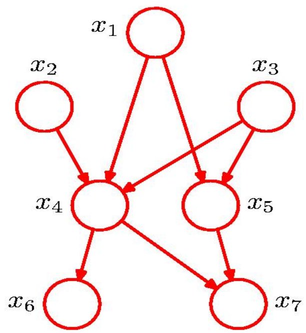


observations: vector of pixel intensities: f

generative models: ancestral sampling

- goal: draw a sample $\widehat{x}_1, \ldots, \widehat{x}_K$ from $p(x_1, \ldots, x_K)$
- step 0: define an ancestral ordering: each node has higher number than any of its parents
- step 1: for n=1:K,
  - sample $\widehat{x}_n$ from $p(x_n | \mathrm{pa}_n)$
  - parent values are always available
- Note: a sample of any marginal is easy to get

© c. bishop

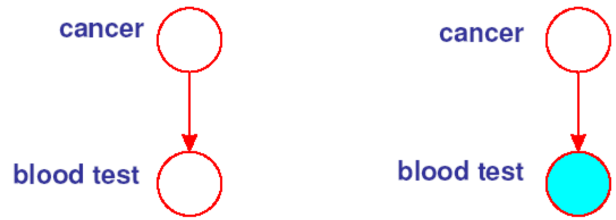


$$p(x_2, x_4) \longrightarrow \widehat{x}_2, \widehat{x}_4$$
$$\not\longrightarrow \{\widehat{x}_{j \neq 2,4}\}$$

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

BNs: causality

- a BN can express causal relationships
- typical problem: inferring hidden **parent** variables from observed **child** variables



© c. bishop

- 'hand-coded' BNs: we assume we know the relation between 'cause' and 'effect'
- discovering causal structure directly from data is a much more complex and subtle problem (e.g. Pearl, 2000)

examples of BNs (1)

- Naive Bayes classifier
- Gaussian Mixture Models (GMM)
- Hidden Markov Models (HMM)
- Kalman Filters (KM)
- Particle Filters (PF)
- Probabilistic Principal Component Analysis (PPCA)
- Factor Analysis (FA)
- Transformed Component Analysis (TCA)
- ....

Random Variables

Discrete?

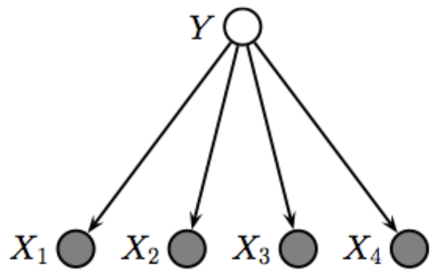
Continuous?

Mixed?

Static?

Dynamic?

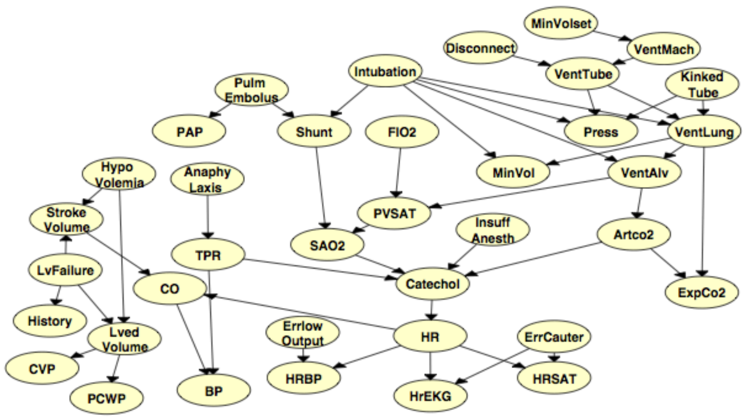
examples of BNs (2) Naive Bayes classifier



$$p(y, \mathbf{x}) = p(y) \prod_{i=1}^{N} p(x_i|y)$$

- y : class
- x_i : features

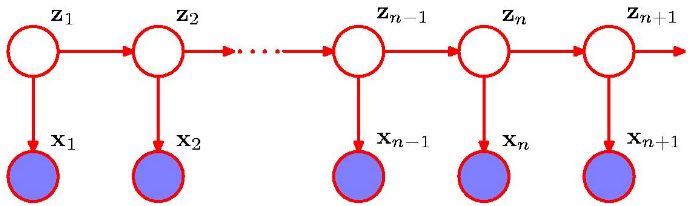
example of BN (3) – Alarm network



© k. murphy

- Surgery room: model has 37 variables, 504 parameters
- created by hand using knowledge elicitation (probabilistic expert system)

examples of BNs (4) - HMMs



$$p(\mathbf{x}_{1:N}, \mathbf{z}_{1:N}) = p(\mathbf{z}_{1:N})p(\mathbf{x}_{1:N}|\mathbf{z}_{1:N})$$
$$= p(\mathbf{z}_1)\prod_{i=2}^{N} p(\mathbf{z}_i|\mathbf{z}_{i-1})\prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{z}_i)$$

HMM: **discrete hidden variables (states), discrete/continuous observations**

Kalman Filter: **linear Gaussian model: continuous hidden variables, continuous observations**

Particle Filter: **non-linear, non-Gaussian model: continuous hidden variables, continuous observations**

Hybrid Models: **discrete/continuous hidden variables, discrete/ continuous observations**

examples of BNs (5)

- 2 important examples of BNs
  - BN over discrete variables
  - Linear Gaussian models

    general case for Probabilistic PCA, Factor Analysis. Linear Dynamical Systems (e.g. Kalman filters)

Categorical distribution (discrete variable)

- for a discrete variable taking 1 out of K values

$$p(\mathbf{x}=k|\boldsymbol{\mu}) = \mu_k \qquad \boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}} \qquad \sum_k \mu_k = 1$$

1-of-K coding scheme

$$\mathbf{x} = (x_1, ..., x_k, ..., x_K)^T, \quad x_k \ \{0,1\} \quad \sum_k x_k = 1$$

example: $\mathbf{x} = (0,0,1,0,0)^T$

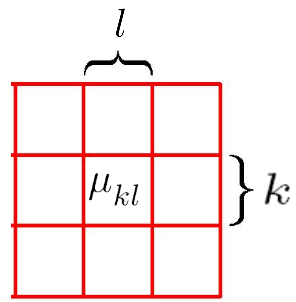
Interest

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \qquad \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}) = \sum_{k=1}^{K} \mu_k = 1$$

- for two joint discrete variables

$$p(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K}\prod_{l=1}^{K} \mu_{kl}^{x_{1k}x_{2l}}$$

- for M variables, K^M-1 parameters

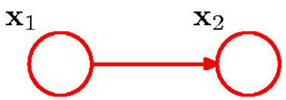


BN for discrete variables: two-node case

$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)$$

$$(K-1) + K(K-1) = K^2 - 1$$

number of parameters



$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1)p(\mathbf{x}_2)$$

$$(K-1) + (K-1) = 2(K-1)$$

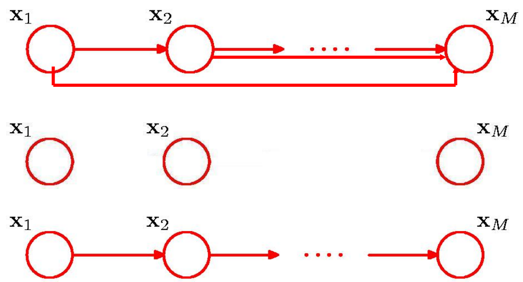
number of parameters

- dropping links
  - reduces the number of parameters
  - restricts the class of distributions modeled by the BN

BN for discrete variables: a general M-node model

number of parameters

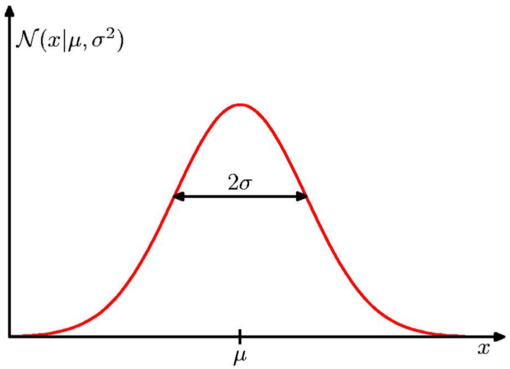
left as an exercise



- all other cases have intermediate complexity

the Gaussian distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



- mean and variance

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x \,\mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x^2 \,\mathrm{d}x = \mu^2 + \sigma^2$$

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

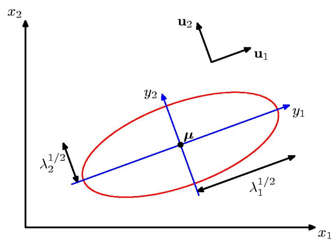
the multivariate Gaussian

$$\mathbf{x} = \{x_1, \ldots, x_K\}$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{K/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

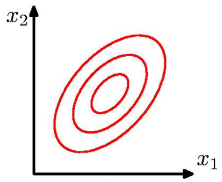
- moments

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma} \qquad \mathrm{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x}-\mathbb{E}[\mathbf{x}])(\mathbf{x}-\mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$

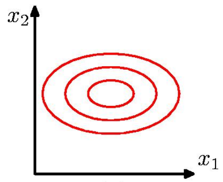


elliptical surfaces of constant probability density

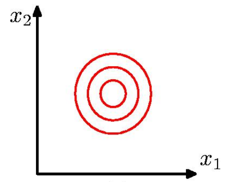
major axes of the ellipse are defined by the eigenvectors of the covariance matrix



(a)
general cov matrix
K(K+1)/2 parameters



(b)
diagonal cov matrix
K parameters



(c)
isotropic cov matrix
1 parameter

linear-Gaussian models (1)

- a multivariate Gaussian can be expressed as a directed graph that corresponds to a linear-Gaussian model over the components
- consider an arbitrary DAG over *D* variables, with all local CPD expressed as a linear gaussian distribution

$$\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$$

define

$$p(\underset{\text{node } i}{x_i}|\mathrm{pa}_i) = \mathcal{N}\left(x_i \,\middle|\, \underbrace{\sum_{j \in \mathrm{pa}_i} w_{ij}x_j + b_i}_{\text{mean}}, \underbrace{v_i}_{\text{variance}}\right)$$

then

$$\begin{aligned} \ln p(\mathbf{x}) &= \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}_i) \\ &= -\sum_{i=1}^{D} \underbrace{\frac{1}{2v_i}\left(x_i - \sum_{j \in \mathrm{pa}_i} w_{ij}x_j - b_i\right)^2}_{\text{quadratic}} + \text{const} \end{aligned}$$

$p(\mathbf{x})$ is a multivariate Gaussian

Important property: all marginals are also Gaussian, e.g. $p(x_i)$, $p(x_1, x_3)$, ...

linear-Gaussian models (1)

- a multivariate Gaussian can be expressed as a directed graph that corresponds to a linear-Gaussian model over the components
- in other words: consider an arbitrary DAG over *D* variables. If each local CPD is expressed as a linear gaussian distribution, then
  - the distribution over all components is a Gaussian
  - note: all marginals are also Gaussian, e.g. $p(x_i)$, $p(x_1, x_3)$, ..

$$\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$$

$$p(\underset{\text{node } i}{x_i}|\mathrm{pa}_i) = \mathcal{N}\left(x_i \,\middle|\, \underbrace{\sum_{j \in \mathrm{pa}_i} w_{ij}x_j + b_i}_{\text{mean}}, \underbrace{v_i}_{\text{variance}}\right)$$

$$\begin{aligned} \ln p(\mathbf{x}) &= \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}_i) \\ &= -\sum_{i=1}^{D} \frac{1}{2v_i}\left(x_i - \sum_{j \in \mathrm{pa}_i} w_{ij}x_j - b_i\right)^2 + \text{const} \end{aligned}$$

$p(\mathbf{x})$ is a multivariate Gaussian

© c. bishop

linear-Gaussian models (2)

- assuming ancestral ordering
- mean and covariance matrix elements of the joint pdf can be computed recursively from the linear weights + noise variance
  - start at the lowest numbered node, then work through the graph
- note: reverse not true, given arbitrary mean and covariance matrix, we can not in general find an equivalent weight and noise

for node i

$$x_i = \sum_{j \in \mathrm{pa}_i} w_{ij} x_j + b_i + \sqrt{v_i}\epsilon_i$$

where ϵ_i is a zero mean, unit variance Gaussian random variable satisfying $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = I_{ij}$, where I_{ij} is the i, j element of the identity matrix

$$\mathbb{E}[x_i] = \sum_{j \in \mathrm{pa}_i} w_{ij} \mathbb{E}[x_j] + b_i.$$

$$\begin{aligned} \mathrm{cov}[x_i, x_j] &= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])\right] \\ &= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left\{\sum_{k \in \mathrm{pa}_j} w_{jk}(x_k - \mathbb{E}[x_k]) + \sqrt{v_j}\epsilon_j\right\}\right] \\ &= \sum_{k \in \mathrm{pa}_j} w_{jk} \mathrm{cov}[x_i, x_k] + I_{ij} v_j \end{aligned}$$

linear-Gaussian models (3)

number of parameters

$w_{ij} = 0 \ \forall i, j$
mean: M parameters, $(b_1, ..., b_M)^T$
covariance matrix: M parameters, $diag(\nu_1, ..., \nu_M)$
M independent univariate Gaussians

$[w_{ij}]$: lower triangular matrix, zero diagonal: $\frac{M(M-1)}{2}$
covariance matrix: general symmetric: $\frac{M(M+1)}{2}$

- other graphs have intermediate complexity

Conditional independence analysis

conditional independence (C.I.)

a is conditionally independent of b given c iff

$$p(a|b,c) = p(a|c)$$

$$p(a,b|c) = p(a|b,c)p(b|c) = p(a|c)p(b|c)$$

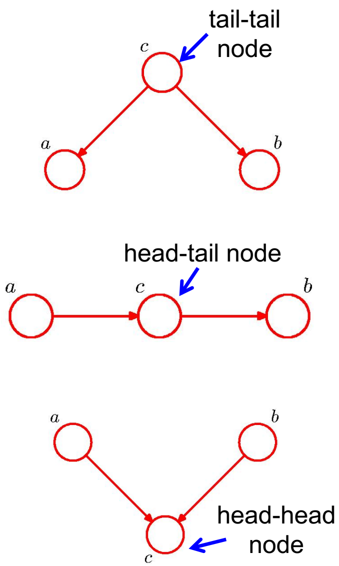
$$a \perp\!\!\!\perp b \mid c$$

conditional independence is key
+ simplify a model's structure
+ reduces computations for learning and inference

"reading" C.I. from a graphical model can be done
+ via the probability definition => time consuming
+ without analytical calculations using d-separation and Bayes ball algorithm

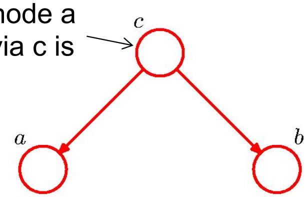
conditional independence (C.I.)

- General approach: **look whether variables along paths** are dependent or not
- Preliminary
  - look at 3 canonical path segments of 3 variables a-c-b
  - study the dependency between a and b depending on whether c is observed (i.e *conditioned* on c)



conditional independence: "canonical" graph 1

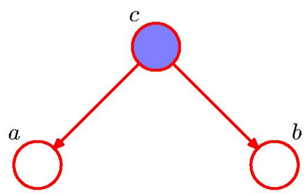
path from node a to node b via c is **tail-to-tail**



$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

$$p(a, b) = \sum_c p(a|c)p(b|c)p(c)$$

$$a \not\!\perp\!\!\!\perp b \mid \emptyset$$ empty set



$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$

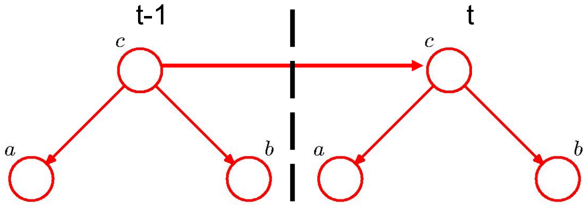
$$a \perp\!\!\!\perp b \mid c$$

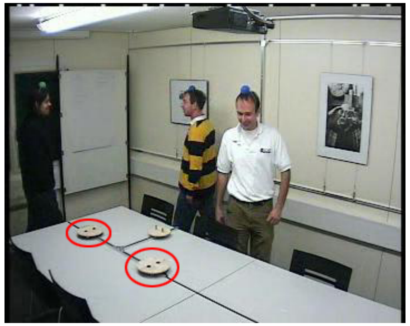
When not observed, c causes a and b to be dependent

Given the knowledge of c, node a and node b are made **independent**

Definition: when observed, c blocks the path from a to b

example of graph 1 : tracking a head from audio-visual observations





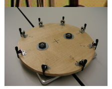
c : object location

b : video observation a : audio observation

Given a person location . room geometry =>
-model of how the image is generated
-model of how the sound is produced
=> conditional independance

camera

microphone array

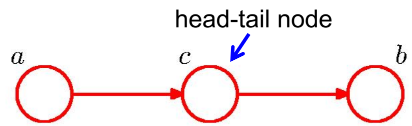


Person location unknown

audio and video dependent

e.g. loud noise may imply the person is in a given area of the image

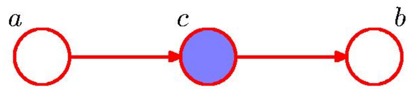
conditional independence: "canonical" graph 2



$$p(a,b,c) = p(a)p(c|a)p(b|c)$$

$$p(a,b) = p(a)\sum p(c|a)p(b|c) = p(a)p(b|a)$$

$$\underline{a \not\perp\!\!\!\perp b \mid \emptyset}$$



$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

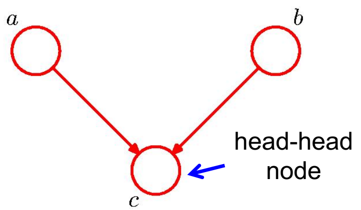
$$\underline{a \perp\!\!\!\perp b \mid c}$$

When not observed, c causes a and b to be dependent

Given c, the additional knowledge of a does not alter the probability of b

=> when observed, c blocks the path from a to b (a and b are independent)

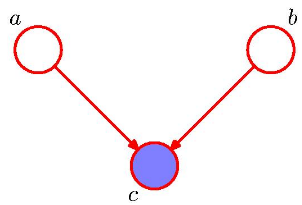
conditional independence: "canonical" graph 3



$$p(a,b,c) = p(a)p(b)p(c|a,b)$$

$$p(a,b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset$$



$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(b)p(c|a,b)}{p(c)} \end{aligned}$$

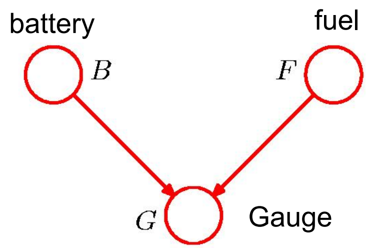
$$a \not\!\perp\!\!\!\perp b \mid c$$

When **not observed**, c **blocks** the path from a to b (a and b are independent)

When **observed**, c un**blocks** the path and a and b become conditionally dependent

Note: general rule for this case needs to look at descendents of node c
cf later and lab

conditional independence: ‘explaining away’



Battery: charged (*B*=1) or flat (*B*=O)

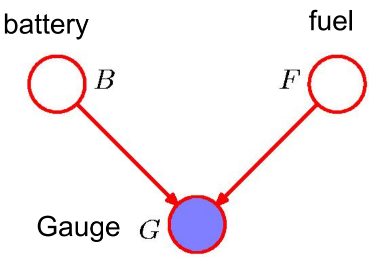
Fuel: full (*F*=1) or empty (*F*=O)

Gauge: Fuel gauge reading (O empty, 1 full)

$$\begin{aligned} p(B=1) &= 0.9 \\ p(F=1) &= 0.9 \end{aligned}$$

$$\begin{aligned} p(G=1|B=1,F=1) &= 0.8 \\ p(G=1|B=1,F=0) &= 0.2 \\ p(G=1|B=0,F=1) &= 0.2 \\ p(G=1|B=0,F=0) &= 0.1 \end{aligned}$$

conditional independence: ‘explaining away’ (2)



assume we observe the Gauge and it is says empty (G=O)

what is the posterior probability that there is no fuel?

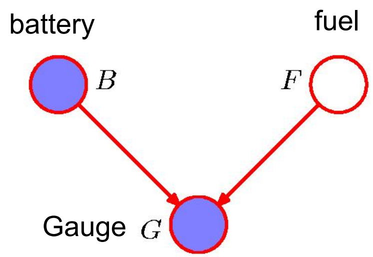
$$p(F=0|G=0)$$

$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)}$$

$$p(F=0) = 0.1$$

it is more likely that the tank is empty

conditional independence: ‘explaining away’ (3)



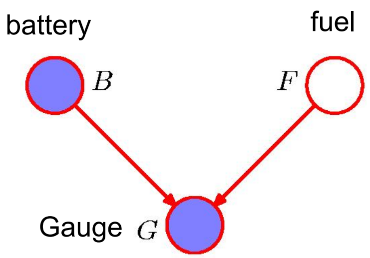
now assume we observe the state of the gauge (empty: G=0), and *additionally*, we observe the state of the battery, and it is flat (B=0)

what is the posterior probability that there is no fuel?

$$p(F=0|G=0,B=0)$$

$$\begin{aligned}
p(F=0|G=0,B=0) &= \frac{p(G=0,B=0,F=0)}{p(G=0,B=0)} \\
&= \frac{p(G=0|B=0,F=0)p(B=0,F=0)}{\sum_{F\in\{0,1\}} p(G=0,B=0,F)} \\
&= \frac{p(G=0|B=0,F=0)p(B=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(B=0)p(F)} \\
&= \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)} = 0.111
\end{aligned}$$

conditional independence: 'explaining away' (4)



$$p(F = 0) = 0.1$$
$$p(F = 0|G = 0) = 0.257$$
$$p(F = 0|G = 0, B = 0) = 0.111$$

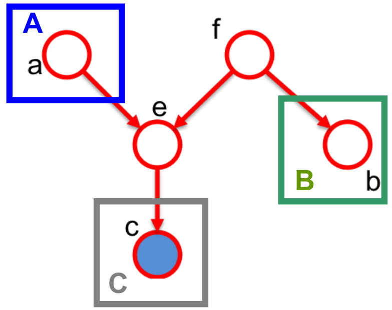
+ the probability that the tank is empty has **decreased** as a result of the observation of the battery

+ observing that the battery is flat **explains away** the observation that the gauge indicates an empty tank

+ states of the battery and fuel tank **became dependent** after observing the state of the gauge

+ posterior is still higher than prior: observing that the gauge reads empty gives evidence in favor of an empty tank

d-separation – general case



- Consider a directed acyclic graph
  - A, B and C are arbitrary non-intersecting sets of nodes
  - union could be smaller than full set of nodes
- Question: is A independent of B given C?
- Definition:

If **all paths** from any node in A to any node in B **are blocked**, then A is d-separated from B by C, and the joint pdf will satisfy

$$A \perp\!\!\!\perp B | C$$

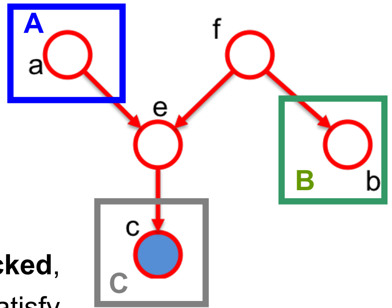
A path is blocked if two nodes on the path get independent,

i.e. if it includes a node such that

a) the arrows on the path meet head-to-tail or tail-to-tail at the node AND the node is in C

b) the arrows meet head-to-head at the node AND **neither the node nor any of its descendants** is in C

d-separation – example 1



- Question: is A independent of B given C?
- Definition:

If **all paths** from any node in A to any node in B **are blocked**, then A is d-separated from B by C, and the joint pdf will satisfy

$$A \perp\!\!\!\perp B | C$$

A path is blocked if two nodes on the path get independent, i.e. if it includes a node such that

a) the arrows on the path meet head-to-tail or tail-to-tail at the node AND the node is in C

b) the arrows meet head-to-head at the node AND **neither the node nor any of its descendants** is in C

Example:

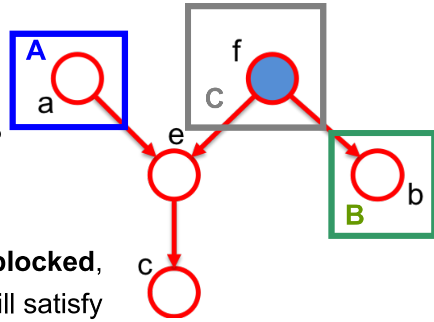
path a – e – f – b

e: head-to-head, not in C, but its descendent is in C: **not blocked**

f: tail-to-tail, not in C: **not blocked**

$$a \not\perp\!\!\!\perp b \mid c$$

d-separation – example 2



- Question: is A independent of B given C?
- Definition:

If **all paths** from any node in A to any node in B **are blocked**, then A is d-separated from B by C, and the joint pdf will satisfy

$$A \perp\!\!\!\perp B | C$$

A path is blocked if two nodes on the path get independent, i.e. if it includes a node such that

a) the arrows on the path meet head-to-tail or tail-to-tail at the node AND the node is in C

b) the arrows meet head-to-head at the node AND **neither the node nor any of its descendants** is in C

Example:

path a – e – f – b

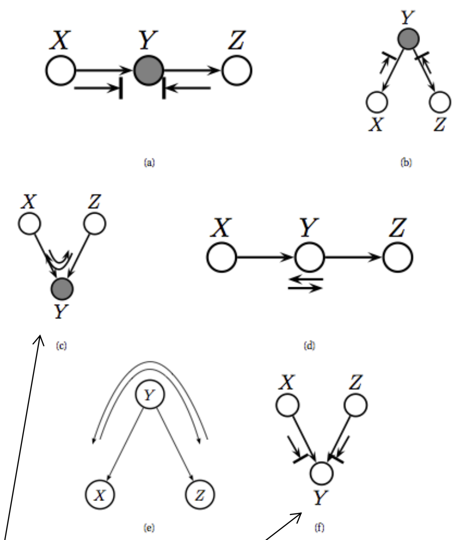
e: head-to-head, not in C, its descendent is not in C: **blocked**

(*f*: tail-to-tail, in C: **blocked**)

$$a \perp\!\!\!\perp b \mid f$$

d-separation : the Bayes ball algorithm

$$A \perp\!\!\!\perp B \mid C$$

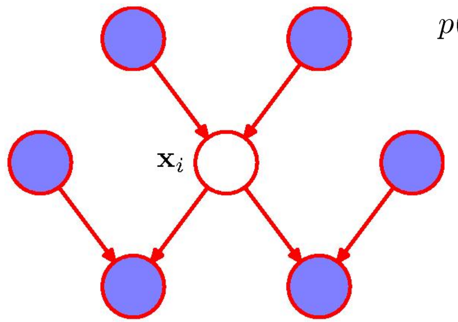


Visual way to assess C.I.

- throw balls from any node in A, and follow (undirected) links
- if a ball can reach B without being blocked along the path, then C.I. assumption does not hold
- blocking rules: cf d-separation

Note: v structure: needs to check descendants

Markov blanket for BN

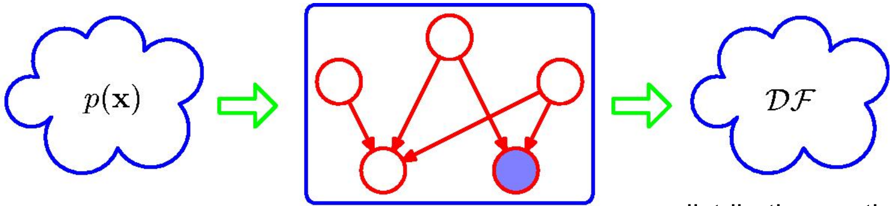


$$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\int p(\mathbf{x}_1,\ldots,\mathbf{x}_M)\,\mathrm{d}\mathbf{x}_i} = \frac{\prod_k p(\mathbf{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k|\mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$

Factors independent of $\mathbf{x}_i$ cancel between numerator and denominator.

- Definition: the set of nodes that renders a node x_i conditionally independent of all the other nodes is called the Markov blanket of x_i
- In a BN, the blanket includes the parents, children, and co-parents

BN: graph and conditional independence



all distributions

graph G

distributions satisfying (at least) all the C.I. assumptions of G

- a graph G essentially encodes a set of C.I. assumptions, denoted CI(G)
- a graph G is an independent map (**I-map**) for a distribution p if the set CI(G) holds true for p (CI(G) included in CI(p))

  ex: the full connected graph is an I-map for all distributions
- a graph G is a **perfect map** for a distribution p if   CI(p) = CI(G)
- G defines thus implicitly a class (set) of distributions; this allows us to use G as a proxy to reason about the C.I. of a distribution

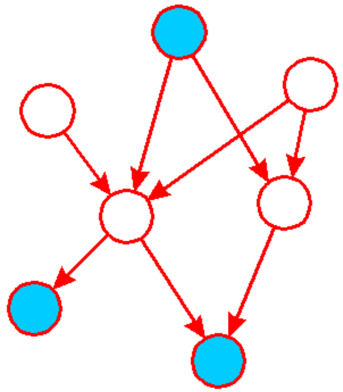
BNs: two fundamental problems

- given a factorized form for

$$p(\mathbf{x}|\theta) = \prod_{k=1}^{L} p(x_k|\mathrm{pa}_k, \theta)$$

- **learning**: given training data (i.e. a set of values for the observed nodes), estimate the parameters for the full BN

  => see second course



- **inference**: given a learned model, compute probabilities in the BN
  - often interested in probabilities of hidden nodes
  - conditioning on evidence
- this is easier said than done!

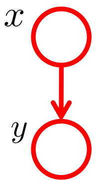
inference (1)

- infer posterior distribution (or its max) of hidden variables given visible ones

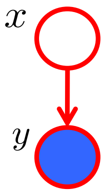
$$p(\mathbf{x}_h|\mathbf{x}_v,\theta) = \frac{p(\mathbf{x}_h,\mathbf{x}_v|\theta)}{p(\mathbf{x}_v|\theta)} = \frac{p(\mathbf{x}_h,\mathbf{x}_v|\theta)}{\sum_{\mathbf{x}'_h} p(\mathbf{x}'_h,\mathbf{x}_v|\theta)}$$

evidence probability

- the general principle is to express the wanted probability in function of the **joint distribution, as we know how to decompose the joint into a product of local probabilities** (that implicitly take advantage of the C.I. assumption).
- normalizing constant: data likelihood or **probability of the evidence**



$$p(x,y) = p(x)p(y|x)$$



$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}$$

$\sum_x p(x)\, p(y|x)$

inference (1)

- infer posterior distribution (or its max) of hidden variables given visible ones

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

evidence probability

- the general principle is to express the wanted probability in function of the **joint distribution, as we know how to decompose the joint into a product of local probabilities** (that implicitly take advantage of the C.I. assumption).
- normalizing constant: data likelihood or **probability of the evidence**

- often, one is interested in only a subset of the hidden variables (the query $\mathbf{x_q}$). To obtain the probabilities of interest, we marginalize out the remaining hidden variables $\mathbf{x_n}$ (called nuisance variables in this context).

  for instance: this can be done by

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

inference (2)

- infer posterior distribution (or its max) of hidden variables given visible ones

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

evidence probability

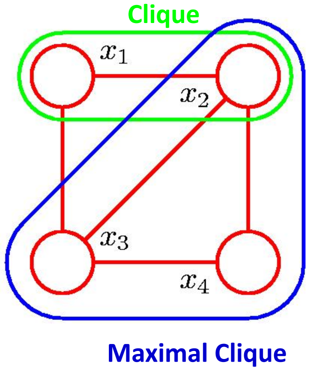
- general case: V random variables, K state each
  - if joint distribution represented by multi-dimensional table => perform exact inference in $O(K^V)$ time .....
- reducing complexity: exploit the graphical structure to find efficient algorithms to compute such pdfs
  - many algorithms can be expressed as propagation of local messages around the graph
  - exact inference and approximate inference algorithms: more later

Undirected graphical models (UGM)

undirected graphical model (UGM)

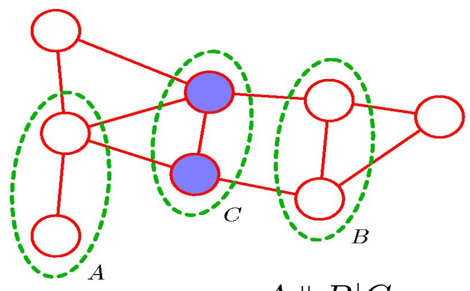
- Markov random field, Markov networks
- undirected links
  - set of nodes, set of edges (symmetric)
- induces a neighbourhood system
  - $Nei(x_s)$: all nodes with a link to x_s

$$Nei(x_1) = \{x_2, x_3\}$$



  - cliques:
    subset of nodes with links between all pairs
    clique C : we denote x_C the set of nodes belonging to this clique
  - maximal cliques:
    clique for which it is not possible to add any other nodes while staying fully connected

undirected graphical model: conditional assumptions



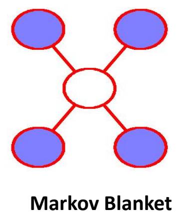
- Global Markov property
  A is C.I. of B given C if, when removing all C nodes there is no path between A and B

$$A \perp\!\!\!\perp B | C$$

- Equivalent to the local Markov property

$$p(x_s | x_i, i \neq s) = p(x_s | x_i, x_i \in Nei(x_s))$$

- The Markov blanket is only made of the neighbours !

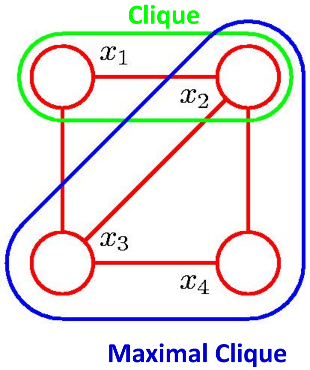


- C.I. much simple to identify than in BN

undirected graphical model: joint distribution

- **Hammersley-Clifford fundamental theorem**.
  Given a graph (nodes, edges), any distribution p(**x**) s.t. p(**x**) > 0 for all **x** can be factorized as:

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{C}\psi_C(\mathbf{x}_C) \qquad Z = \sum_{\mathbf{x}}\prod_{C}\psi_C(\mathbf{x}_C)$$



where C denotes the maximal cliques, and the normalization Z is called the partition function

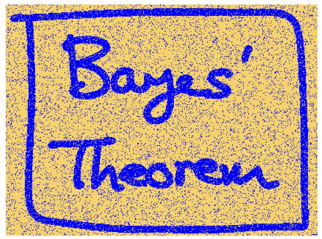
- the potential functions are often written as energy terms

$$\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\}$$

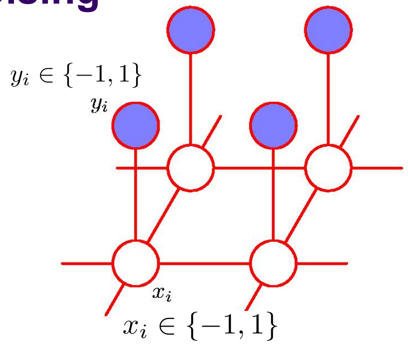
- defining an MRF : defining a set of energy functions over the maximal cliques

example : Ising model – image denoising

Original Image x (hidden)
$\mathbf{x} = \{x_i, i = 1 \ldots N\}$



Noisy Image y (observed)
$\mathbf{y} = \{y_i, i = 1 \ldots N\}$

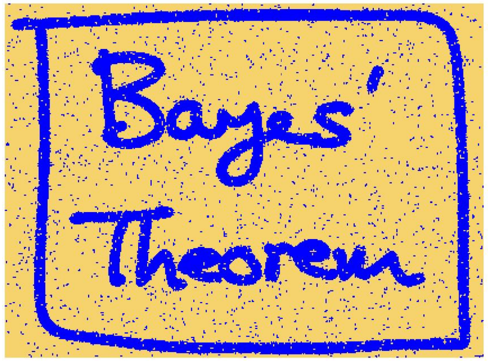


$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z}\exp\{-E(\mathbf{x}, \mathbf{y})\} \qquad E(\mathbf{x}, \mathbf{y}) = -\beta\underbrace{\sum_{i,j} x_i x_j}_{\text{smoothing prior (favors pixels with same sign)}} - \eta\underbrace{\sum_i x_i y_i}_{\text{favors restored pixels to be similar to observations}}$$

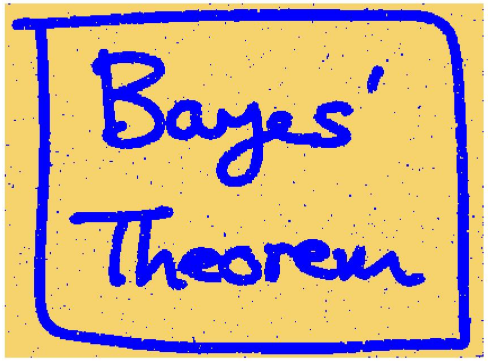
- pairwise cliques
- find the (hidden) original image **x** given the (observed) noisy version

$$\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x}, \mathbf{y}) \Rightarrow \min_{\mathbf{x}} E(\mathbf{x}, \mathbf{y})$$

example : Ising model – image denoising



Restored Image (ICM)



Restored Image (Graph cuts)

- Iterated Conditional Modes (ICM): local inference scheme
- Graph cuts: optimal solution

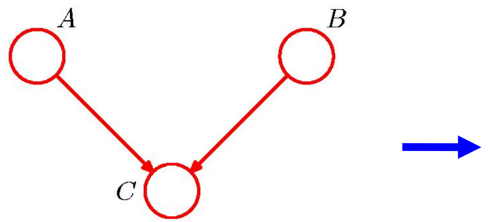
Converting a BN into an UGM

- can we convert a BN into an UGM ?

$$p(\mathbf{x}) = \underbrace{p(x_1)p(x_2|x_1)}_{}\; p(x_3|x_2)\cdots p(x_N|x_{N-1})$$

$$p(\mathbf{x}) = \frac{1}{Z}\,\psi_{1,2}(x_1,x_2)\;\psi_{2,3}(x_2,x_3)\cdots\psi_{N-1,N}(x_{N-1},x_N)$$

Converting a BN into an UGM

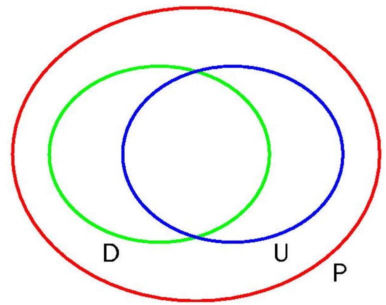


$$p(A,B,C) = p(A)p(B)p(C|A,B)$$
$$= \frac{1}{Z}\psi_1(A)\psi_2(B)\psi_3(A,B,C)?$$

- we need to mary the parents to respect the UGM decomposition : this is called moralization
- note that we have lost some C.I. properties of the initial graph

$$A \perp\!\!\!\perp B \mid \emptyset$$

UGM vs BN : conditional independence



- P : set of distributions
- D : set of distributions for which there exists a directed graph (BN) that is a perfect map (i.e. that has exactly the same C.I. properties)
- U : set of distributions for which there exists an undirected graph that is a perfect map
- examples in D and U: trees

undirected GM vs BN

- advantages over BN
  - symmetric and more 'natural' for some domains such as spatial statistics, relational data
  - UGM handles conditioning on features to give p(y|x) in a more desirable way (Conditional Random Field or CRFs)
- disadvantages
  - parameter learning in UGMs is more computationaly expensive
  - UGMs are not 'modular': it is not possible to plug-in off-the-shelf CPDs
- inference is (basically) the same in BNs and UGMs

Summary

- probabilistic graphical models
  - probability distribution over graphs
  - directed and undirected versions
- Bayesian networks
  - factorized distribution over DAGs, parents and children
  - generative models: ancestral sampling
  - basic cases: discrete and linear Gaussian models
  - conditional independence, explaining away phenomenon, d-separation
  - key tasks: learning and inference
- Markov random field
  - factorized distribution as potential product
  - Conditional independence as local Markov properties