Probabilistic Graphical Models

Florence Forbes

florence.forbes@inria.fr

Mistis team

http://mistis.inrialpes.fr

INRIA Grenoble RHONE-ALPES, Lab. Jean Kuntzman

Probabilistic graphical models

- Graphical models are used in various domains:
  - Machine learning and artificial intelligence
  - Computational biology
  - Statistical signal and image processing
  - Communication and information theory
  - Statistical physics…..

- Based on correspondences between graph theory and probability theory

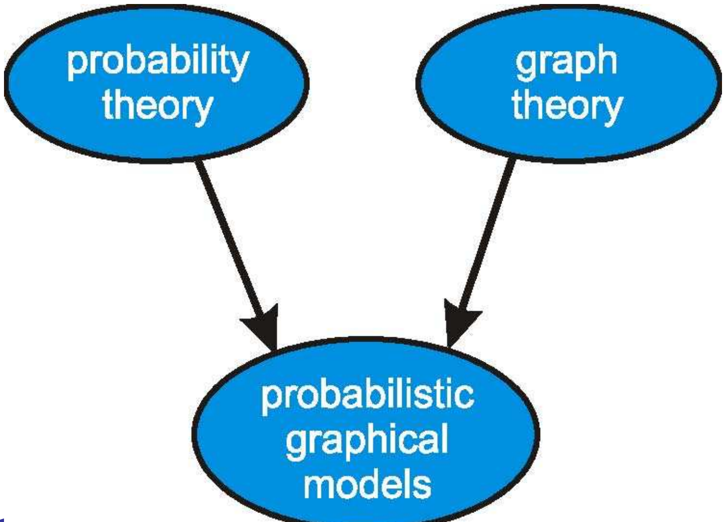
- Important but difficult problems:
  - Computing **likelihoods**, **marginal distributions**, **modes**
  - Estimating model **parameters** and **structure** from noisy data

Probabilistic Graphical Models

- **Role of the graphs:**

graphical representations of probability distributions

– Visualize the structure of a model
– Insights into the model properties (eg conditional independence)
– Design and motivate new models
– Design graph based algorithms for inference



Probability Theory

- Sum rule

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$

- Product rule

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

- From these we have Bayes' theorem

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

  – with normalization $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$

All probabilistic inference and learning manipulations amount to repeated application of these 2 equations

Outline of the talk

- Directed graphs: Bayesian Networks
- Conditional independence and Markov properties
- Undirected graphs: Markov Random Fields
- Inference and learning
- Some illustrations

Directed graphs

Bayesian Networks

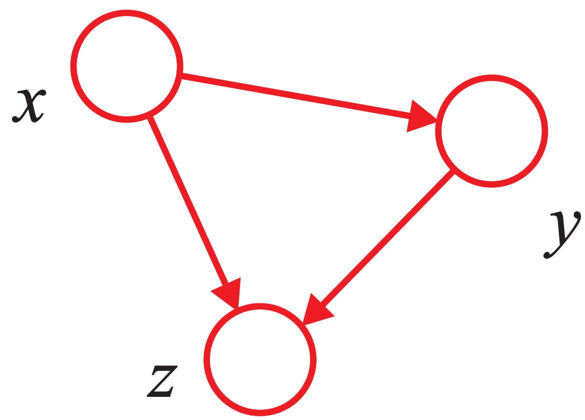
Directed Graphs: Decomposition

- Consider an arbitrary joint distribution

$$p(x, y, z)$$

- By successive application of the product rule

$$\begin{aligned} p(x, y, z) &= p(x)p(y, z|x) \\ &= p(x)p(y|x)p(z|x, y) \end{aligned}$$



General Case

- **Arbitrary** joint distribution,

$$P(x_1, \ldots, x_n)$$

- Successive application of the product rule

$$P(x_1, \ldots, x_n) = P(x_1)P(x_2|x_1) \ldots P(x_n|x_1 \ldots x_{n-1})$$

- Can be represented by a **fully connected graph** (links to all lower-numbered nodes)

Information is in the absence of links

General relationship

- Factorization property

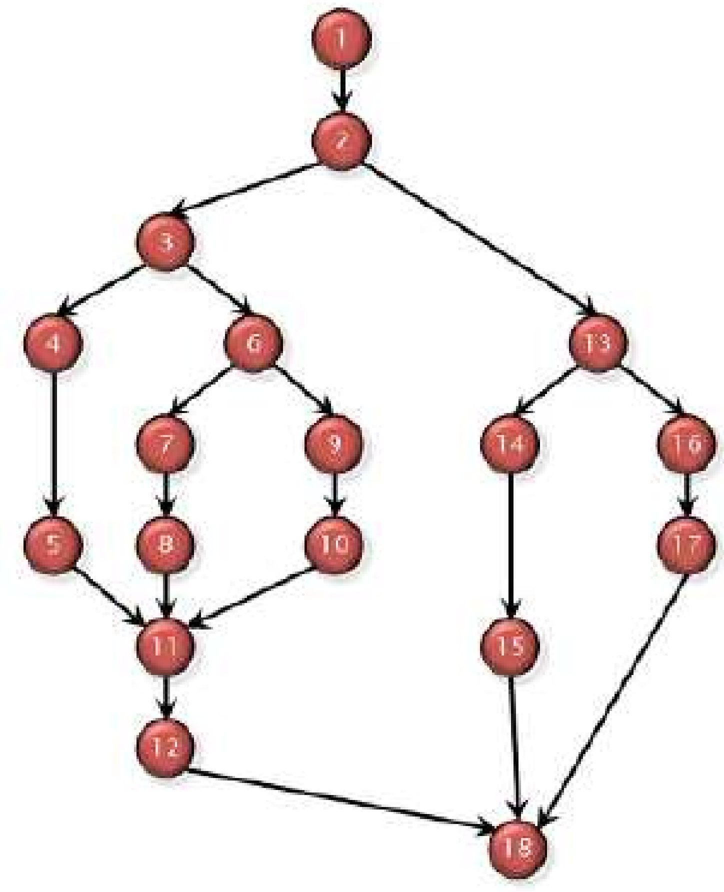
$$P(x_1, \ldots x_n) = \prod_{k=1}^{n} P(x_k | pa_k)$$

Where pa_k denotes the parents of x_k

- Missing link imply conditional independencies

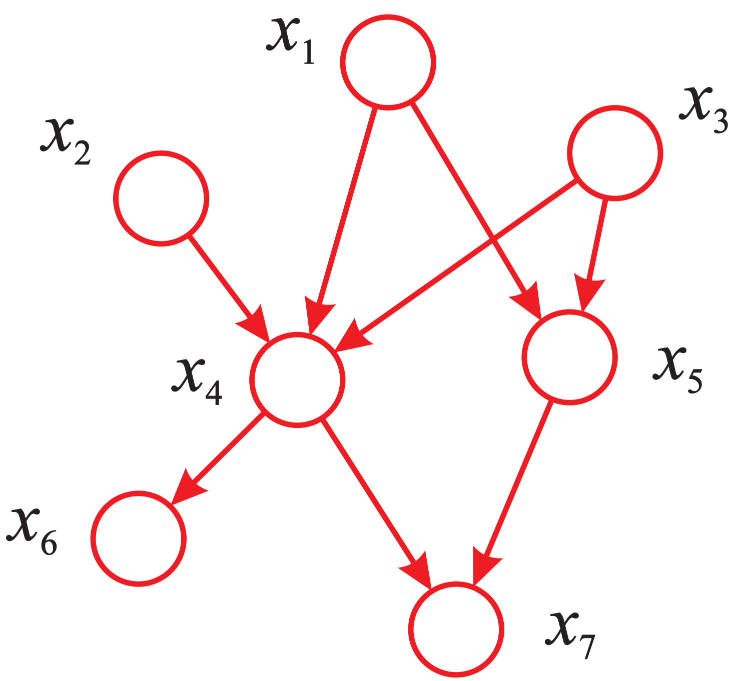
Graph Terminology

- Directed graph G: set of nodes and directed edges
- Acyclic graph: no loop in the graph
- Parents of a node X:
  Y such that Y → X in G
- Descendent of a node X:
  Y that can be reached from X following directed edges



Directed Acyclic Graphs: Bayesian Networks

- The graph can be used to impose constraints on the random vector $(x_1, \ldots, x_7)$ (ie. on the distribution P):



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

No directed cycles

Bayesian Network

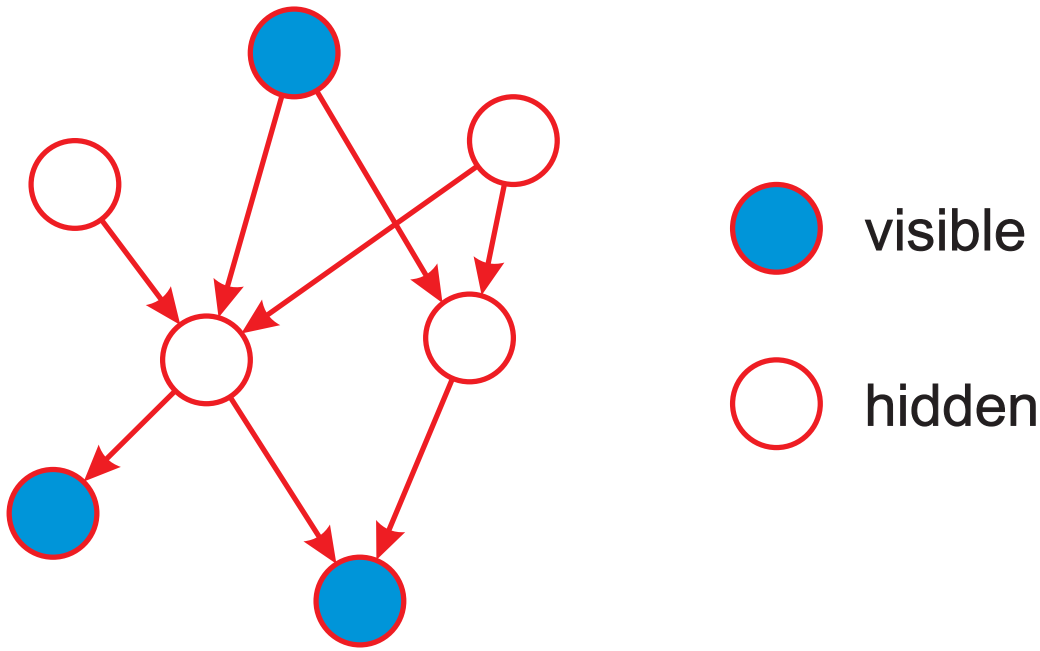
- A couple (p, G) so that

$$p(x_1, \ldots, x_n) = \prod_k p(x_k | pa_k^G)$$

$$p \sim \mathcal{L}(G)$$

Hidden variables

- Variables may be hidden (latent) or visible (observed)



- Latent variables may have a specific interpretation, or may be introduced to permit a richer class of distribution

Example 1: Mixtures of Gaussians

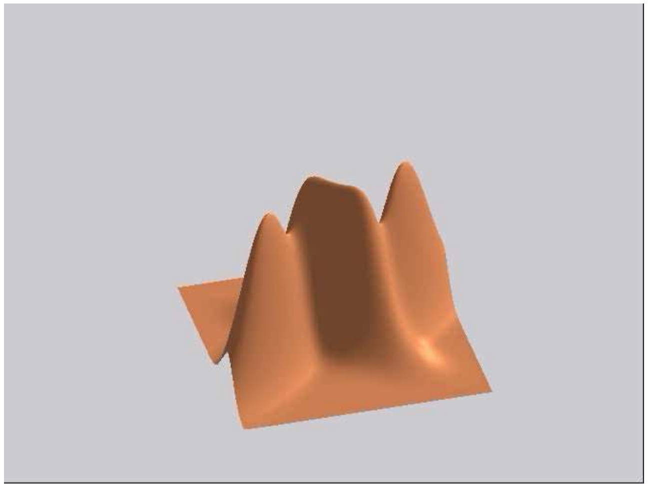
- Linear super-position of K Gaussians

$$P(y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y|\mu_k, \sigma_k^2)$$

- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

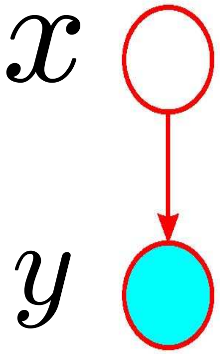
- illustration: mixture of 3 Gaussians



Latent Variable Viewpoint

- Discrete latent variable $x \in \{1, \dots K\}$ describing which component generated data point y
- Conditional distribution of observed variable

$$P(y|X = k) = \mathcal{N}(y|\mu_k, \sigma_k^2)$$



- Prior distribution of latent variable

$$P(X = k) = \pi_k$$

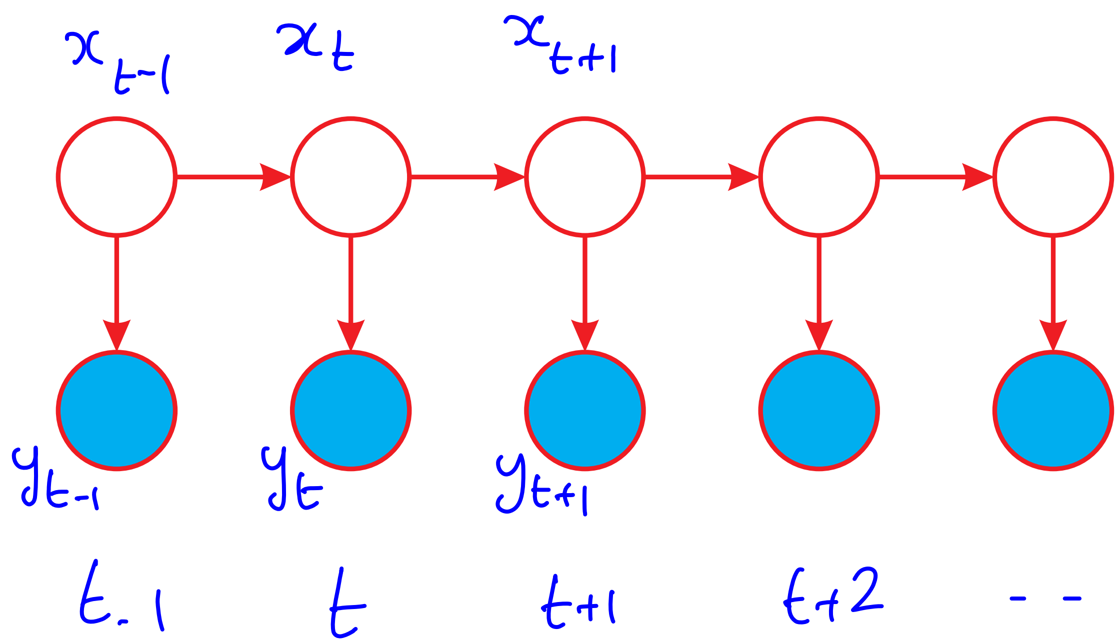
- Marginalizing over the latent variable we obtain

$$P(y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y|\mu_k, \sigma_k^2)$$

Example 2: State Space Models

- Hidden Markov chain
- Kalman filter

... $p(x_t|x_{t-1})\,p(y_t|x_t)\,p(x_{t+1}|x_t)$ ...

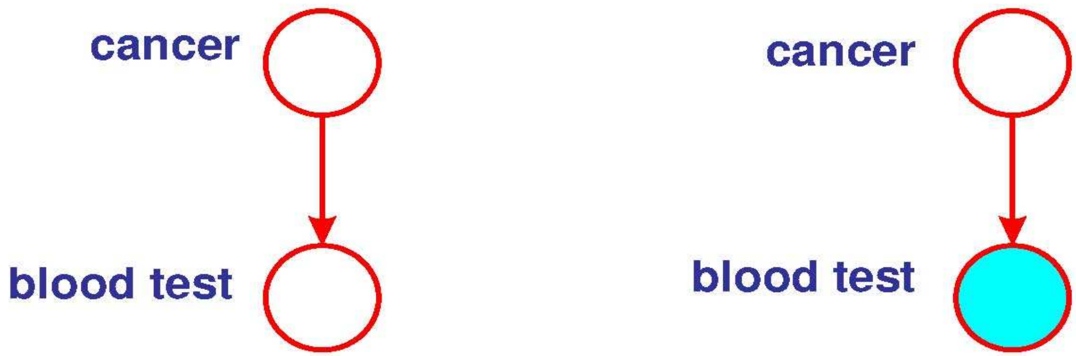


- Frequently wish to solve the problem of computing

$$P(x_t|y_1,\ldots,y_n)$$

Causality

- Directed graphs can express causal relationships
- Often we observe child variables and wish to infer the posterior distribution of parent variables
- Example:



- Note: inferring causal structure from data is subtle

Conditional independence and Markov properties

Conditional independence

- X independent of Y given Z if for all values of z,

$$P(x|y,z) = P(x|z)$$

- Notation:

$$X \perp Y | Z$$

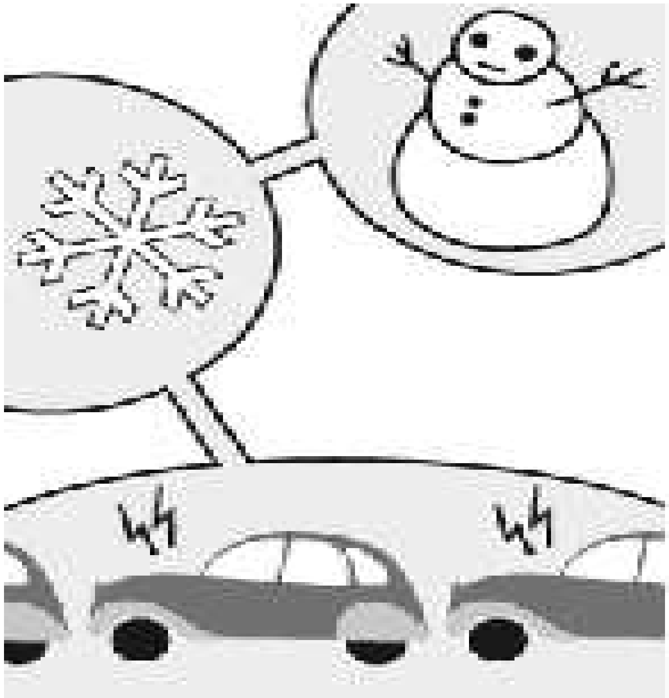
- Equivalently

$$\begin{aligned} P(x,y|z) &= P(x|y,z)P(y|z) \\ &= P(x|z)P(y|z) \end{aligned}$$

- Conditional independence crucial in practical applications since we can rarely work with a general joint distribution

Difference between dependence and conditional dependence

- Traffic jams and snowmen are correlated
- But conditionally on snow falls, the size of the traffic jams and the number of snowmen are independent



The concept of conditional dependence is more suited than dependence to capture « direct » dependencies between variables

Markov properties

- Can we determine the conditional independence properties of a distribution directly from its graph?

- YES: "d-separation", one subtleties due to the presence of head-to-head nodes, *explaining away effect*

Head-to-head node

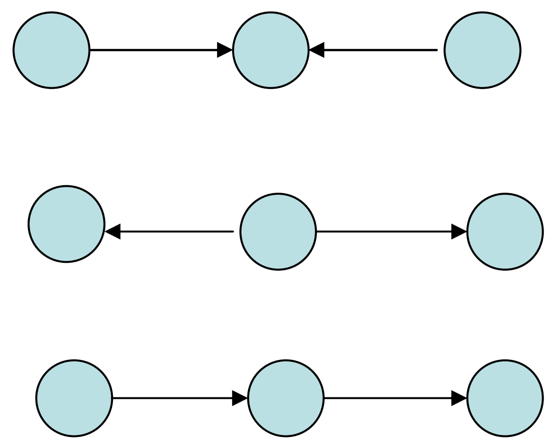
A common effect

Tail-to-tail

A common cause

Head-to-tail

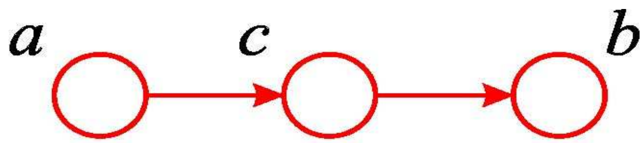
An indirect effect



Example 1: Tail-to-head node

- Joint distribution

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



$$a \not\perp b \quad (c \text{ not observed})$$

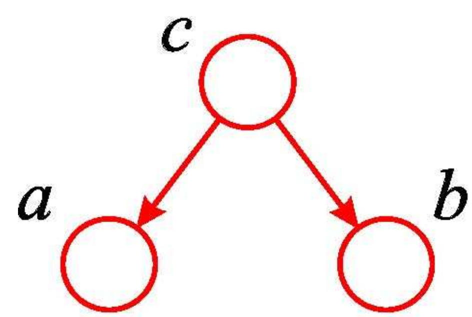
$$P(a,b|c) = P(a|c)P(b|c) \quad \Longrightarrow a \perp b|c \quad (c \text{ observed})$$

- An observed c *blocks the path* from a to b

Example 2: Tail-to-tail node

- Joint distribution

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



$$a \not\perp b \quad (c \text{ not observed})$$

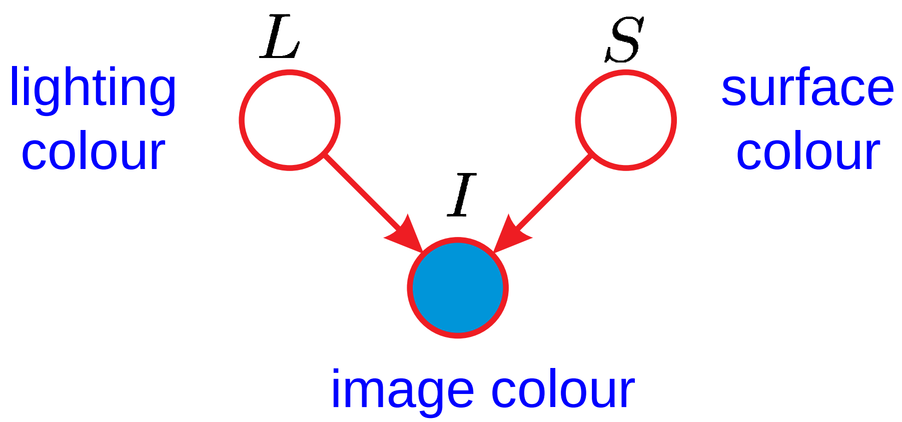
$$P(a,b|c) = P(a|c)P(b|c) \quad \Longrightarrow a \perp b|c \quad (c \text{ observed})$$

- An observed c *blocks the path* from a to b

Example 3: "Explaining Away" (V-structure)

Illustration: pixel colour in an image

$$p(I, L, S) = p(I|L, S)p(L)p(S)$$



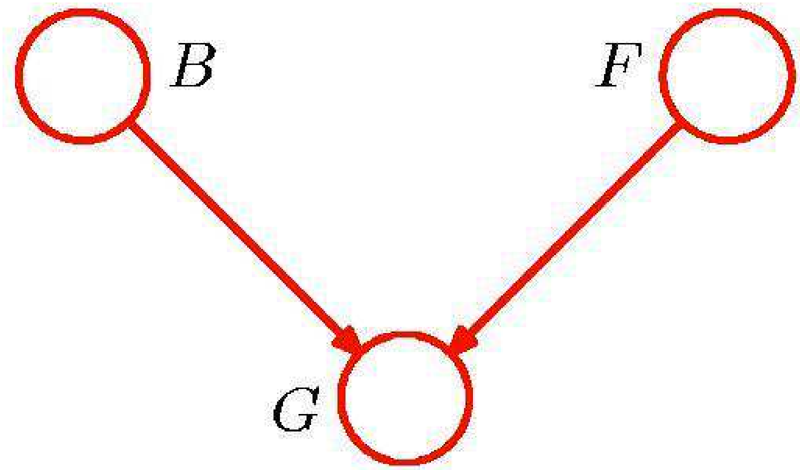
$$p(L, S) = p(L)p(S)$$

$$p(L, S|I) \neq p(L|I)p(S|I)$$

An observed I *unblocks* the path from S to L

Illustration: « Am I out of fuel? »

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



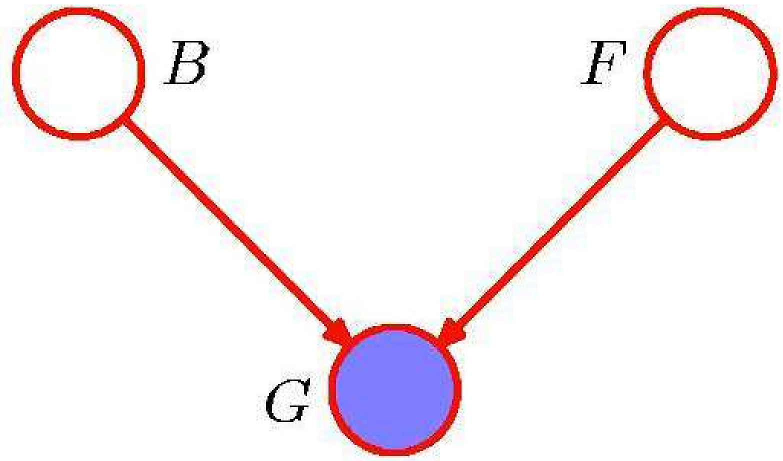
$$
\begin{aligned}
p(B=1) &= 0.9 \\
p(F=1) &= 0.9
\end{aligned}
$$

and hence

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

B = Battery (0=flat, 1=fully charged)
F = Fuel Tank (0=empty, 1=full)
G = Fuel Gauge Reading (0=empty, 1=full)

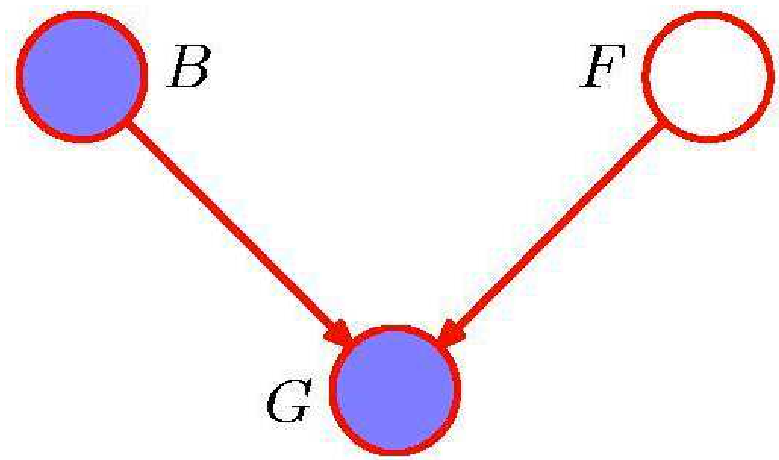
Illustration: « Am I out of fuel? »



$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)} \simeq 0.257$$

The probability of an empty tank increased by observing G = 0.

Illustration: « Am I out of fuel? »



$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)} \simeq 0.111$$

- The probability of an empty tank is **reduced** by observing B = 0. This is referred to as “explaining away”.
- B and F are **negatively correlated** conditioned on G despite being independent

d-separation: Consider 3 groups of nodes A, B, C

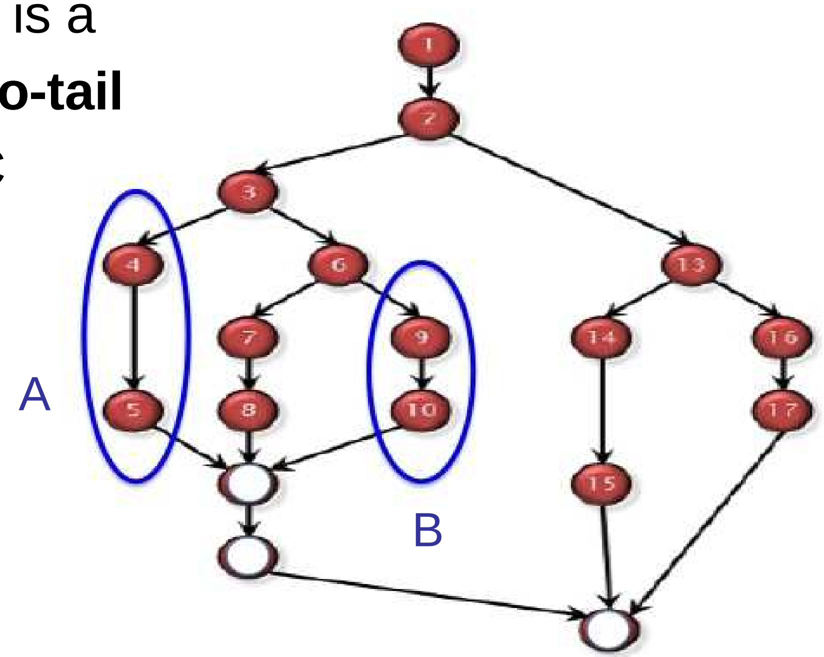
To determine whether $A \perp B | C$ is true, consider all possible paths from any node in A to any node in B

- $A \perp B | C$ true if all paths from A to B are blocked by C

- Any such **path is blocked** if there is a node X which is **head-to-tail or tail-to-tail** with respect to the path and **X is in C**

Or

if the node is **head-to-head** and neither the **node nor any** of **its descendants** is in C



Undirected graphs

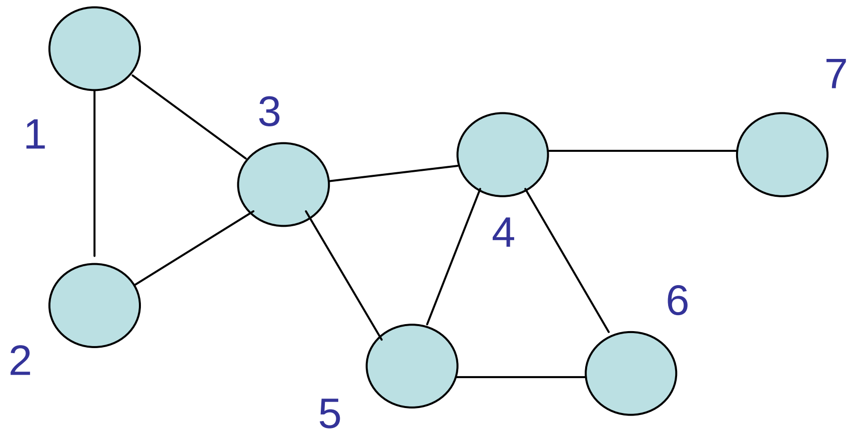
Markov Random Fields

Undirected graphical models

- The second major class of graphical models
- Graphs specify **factorizations** of distributions and sets of conditional independence relations (**Markov properties**)
- **Markov Random Fields** or Markov network

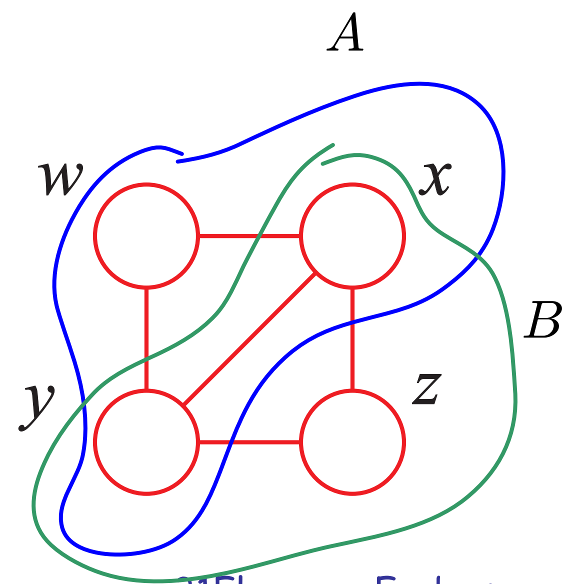
Cliques and maximal cliques

- A clique C is a subset of vertices all joined by edges



- Cliques: (1), (2), ….(12), (23)…..
- Maximal cliques: (123), (345), (456), (47)

$$p(w,x,y,z) = \frac{1}{Z}\psi_A(w,x,y)\psi_B(x,y,z)$$



Undirected Graphs: Factorization

- Provided $p(\mathbf{x}) > 0$ then joint distribution is product of non-negative functions over the *cliques* of the graph

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

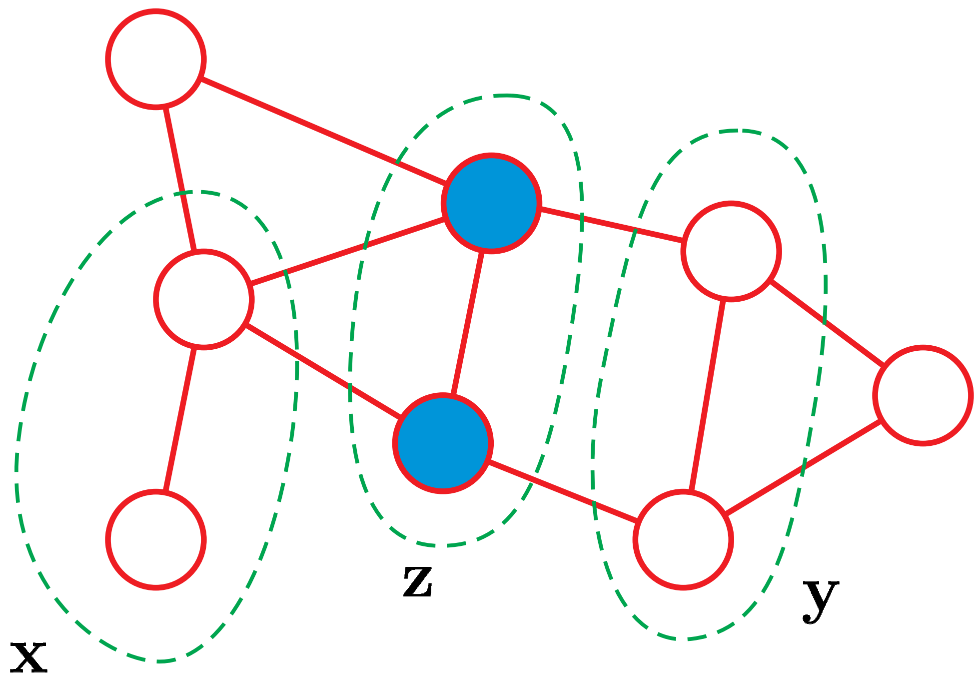
$$X = \{X_i, i \in V\} \qquad X_C = \{X_i, \quad i \in C\}$$

- where $\psi_C(\mathbf{x}_C)$ are the *clique potentials*, and *Z* is a normalization constant

Undirected graphs: conditional independencies

- Conditional independence given by graph **separation**

$\mathbf{x}$ independent of $\mathbf{y}$ given $\mathbf{z}$



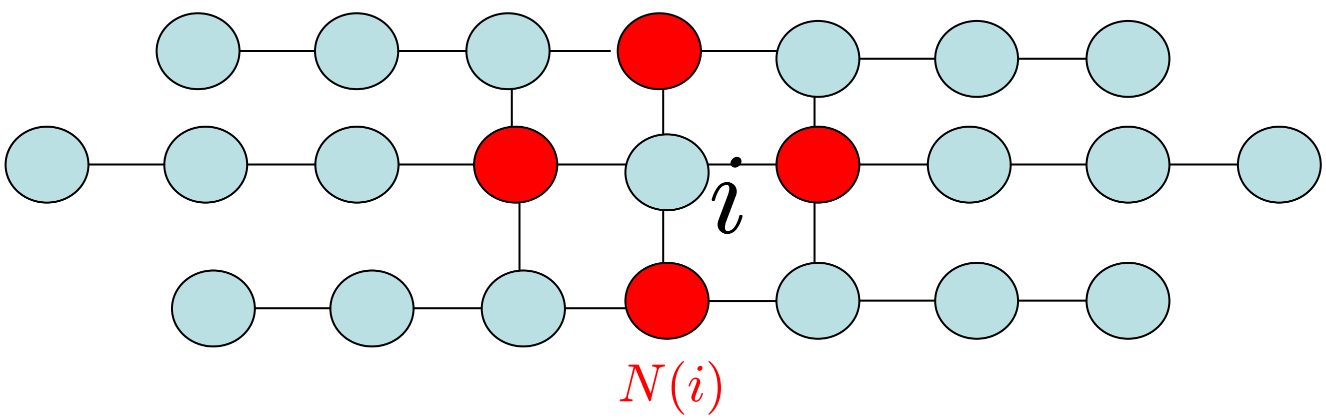
Conditional independencies: Markov properties

Terminology: Markov blanket or Markov Boundary

of a node x_i is the set of nodes $N(i)$ such that

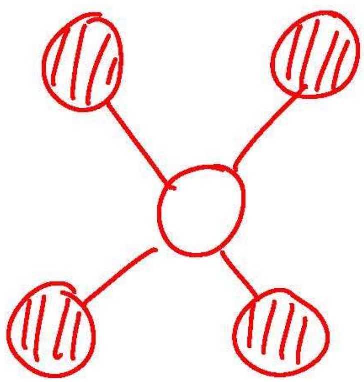
$$P(x_i | x_{-i}) = P(x_i | x_{N(i)})$$

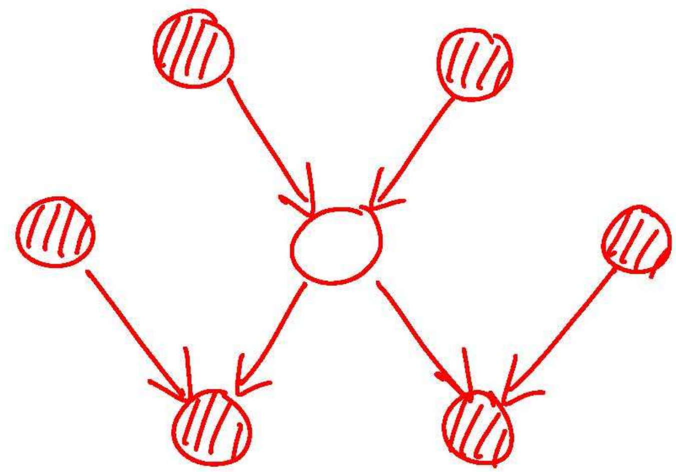
or equivalently $X_i \perp X_{-i \cup N(i)} | X_{N(i)}$



Markov blankets on the graph

- Directed case: Parents, Children, Co-parents
- Undirected case: Neighbors





Markov property: for X (p) wrt G

- Graph G=(V,E)

- $X = \{X_i, i \in V\}$ random vector
- $X_A = \{X_i, i \in A\}$

- X **is Markov wrt G**

  if X_A and X_B are *conditionally independent* given X_C

  whenever C *separates* A *and* B

- Specifying conditional independencies using the neighborhood N(i) is enough (V finite)

Undirected graphs: Markov Networks

The law of the random variable $X = (X_1 \dots X_n)$
is a graphical model according to the non-directed graph G

if for all i :

$$X_i \perp \{X_j, j \notin N(i) \cup \{i\}\} \mid \{X_j, j \in N(i)\}$$

$$\text{We can write } \mathcal{L}(X) \sim G$$

Connection with directed acyclic graphs

The Moral graph gets the parents married

The moral graph Gm associated to a directed acyclic graph G is obtained by:

- Setting an edge between each parent of each nodes
- Replacing arrows by edges

We have:

$$\mathcal{L}(X) \sim \mathrm{G} \implies \mathcal{L}(X) \sim \mathrm{Gm}$$

Hammersley-Clifford theorem

In practice (**computation**), we use the connection between conditional independencies (Markov properties) and **factorization** property

- Boltzmann-Gibbs representation

$$\Psi_c(x_c) = \exp(-E(x_c))$$

- **P is a positive MRF (satisfies Markov properties) is equivalent to P is a Gibbs distribution**

$$P(x) = \frac{1}{Z} \exp(-E(x))$$

- Energy function

$$E(x) = \sum_c E_c(x_c)$$

Example: pairwise Markov Random Fields

- Cliques: pairs, singletons

$$E(x) = \sum_i \{\Psi_i(x_i) + \frac{1}{2} \sum_{j \in N(i)} \Psi_{ij}(x_i, x_j)\}$$

- Famous ones:
  - **Ising** model: **binary** variables on a graph G with pairwise interactions

$$P(x;\theta) = \frac{1}{Z} \exp(\sum_i \theta_i x_i + \sum_{i \sim j} \theta_{ij} x_i x_j)$$

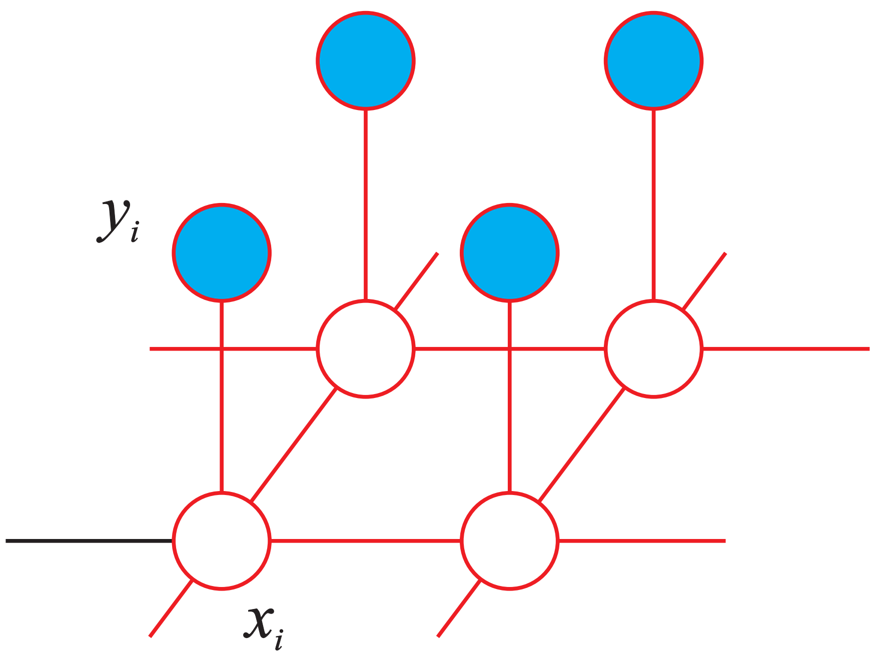
  - **Potts** model: **K-ary** variables

Interaction parameters+ external field parameters

Example: graph representation of a Pairwise MRF

- Typical application: image region labelling

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{ij}(x_i, x_j)$$



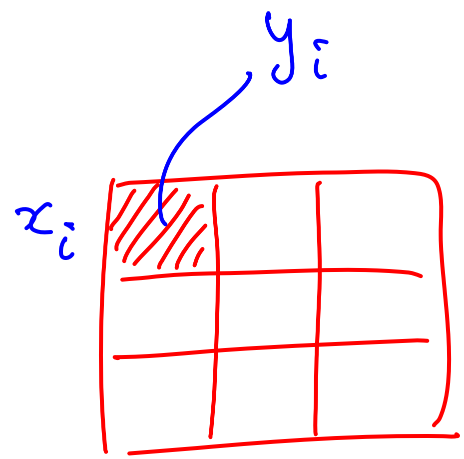
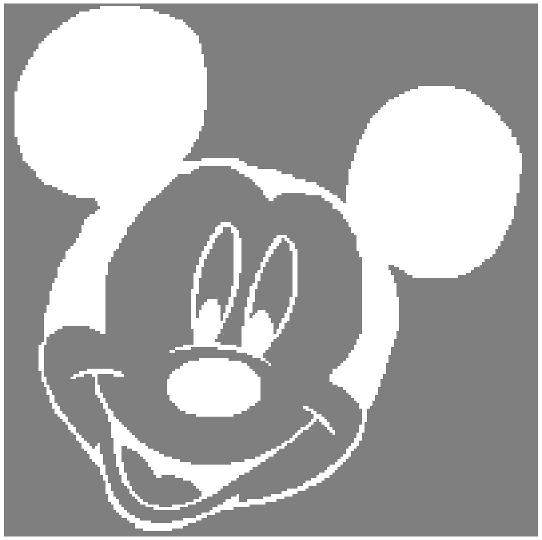


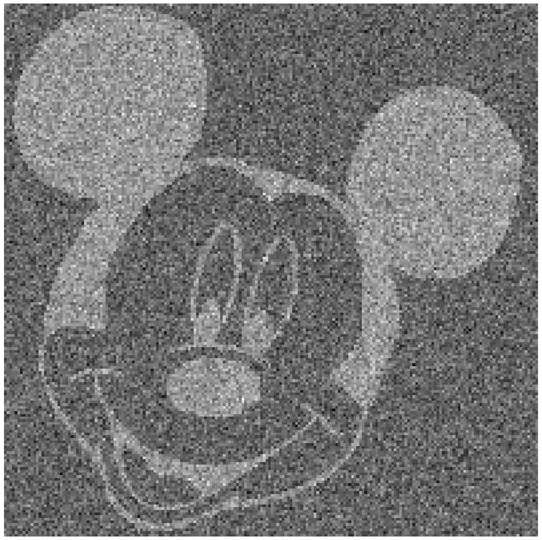
Illustration: image segmentation

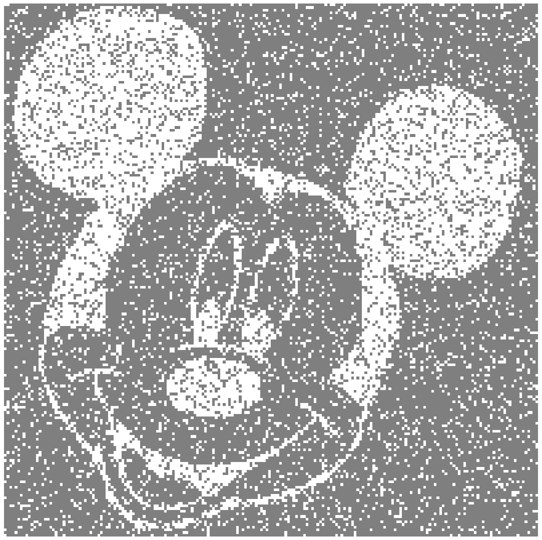
site/vertex i: pixel,

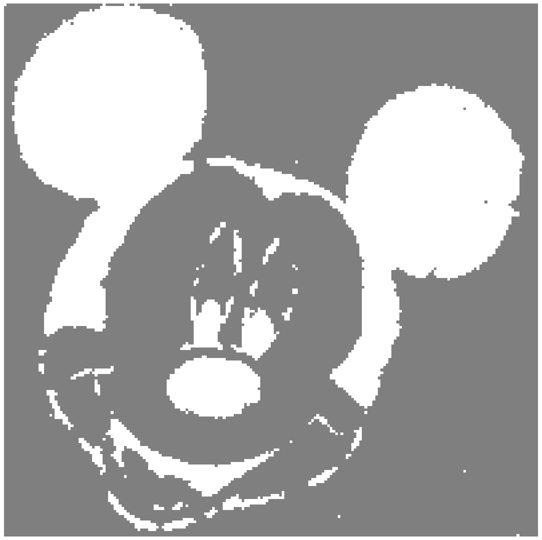
y_i : observed grey level,

x_i: label/0 or 1/ binary variable









Challenging computational problems

- Frequently, it is of interest to compute various quantities associated with an undirected graphical model:
  - The log normalization constant log Z
  - Local marginal distributions (p(xi)) or other local statistics
  - Modes and most probable configurations

- Often grow rapidly with graph size and max clique size
- Example: Computing the normalization constant for binary random variables

$$Z = \sum_{x \in \{0,1\}^n} \prod_{c \in C} \psi_c(x_c)$$

Complexity scales exponentially as 2^n

Inference and learning

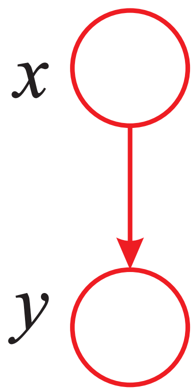
Inference in Graphical models

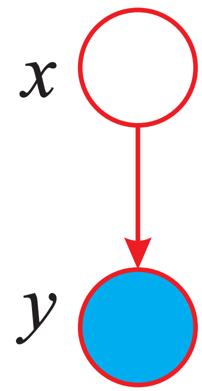
- **Exploit the graphical** structure to find efficient algorithm for inference and to make the structure of these algorithms clear (eg **propagation of local messages** around the graph)

- Exact inference
- Approximate inference

Inference

- Simple example: Bayes' theorem





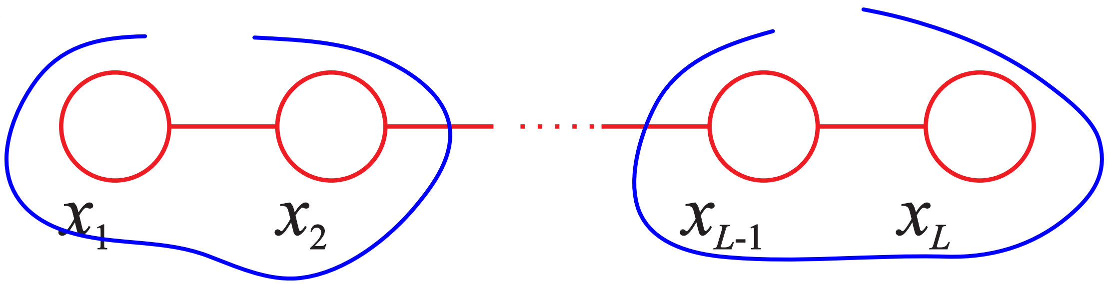
$$p(x, y) = p(x)p(y|x)$$

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

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

Message Passing: compute marginals

- Example



- Find marginal for a particular node

$$p(x_i) = \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_L} p(x_1, \ldots, x_L)$$

  – for M-state nodes, cost is $O(M^L)$
  – exponential in length of chain
  – but, we can exploit the graphical structure (conditional independencies)

Message Passing

- Joint distribution

$$p(x_1, \ldots, x_L) = \frac{1}{Z}\psi(x_1, x_2) \ldots \psi(x_{L-1}, x_L)$$

- Exchange sums and products: ab+ ac = a(b+c)

$$m_\alpha(x_i) \quad \text{before } x_i$$

$$p(x_i) \;=\; \underbrace{\frac{1}{Z} \cdots \sum_{x_2} \psi(x_2, x_3) \left[\sum_{x_1} \psi(x_1, x_2)\right]}_{m_\alpha(x_i)}$$

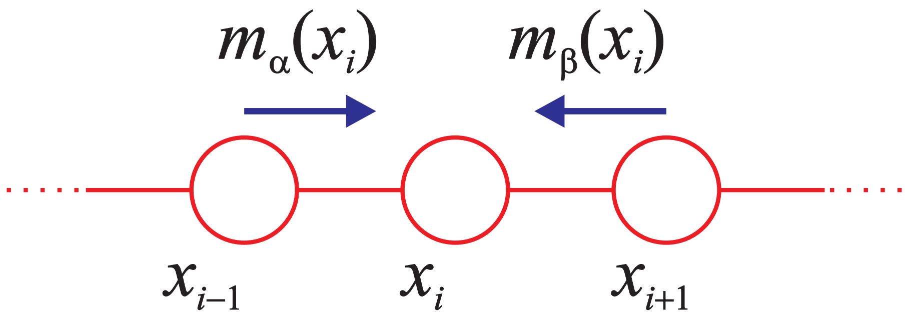
$$\underbrace{\cdots \sum_{x_{L-1}} \psi(x_{L-2}, x_{L-1}) \left[\sum_{x_L} \psi(x_{L-1}, x_L)\right]}_{m_\beta(x_i)}$$

$$m_\beta(x_i) \quad \text{after } x_i$$

Message Passing

- Express as product of messages

$$p(x_i) = \frac{1}{Z} m_\alpha(x_i) m_\beta(x_i)$$



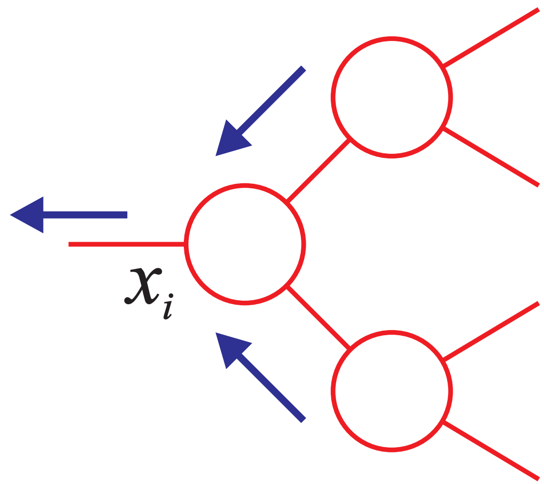
- Recursive evaluation of messages: Linear in L

$$m_\alpha(x_i) = \sum_{x_{i-1}} \psi(x_{i-1}, x_i) m_\alpha(x_{i-1})$$
$$m_\beta(x_i) = \sum_{x_{i+1}} \psi(x_i, x_{i+1}) m_\beta(x_{i+1})$$

- Find Z by normalizing $p(x_i)$

Belief Propagation

- Extension to general tree-structured graphs
- At each node:
  - form product of *incoming* messages and local evidence
  - marginalize to give *outgoing* message
  - one message in each direction across every link



- Fails if there are loops

Junction Tree Algorithm

- An efficient exact algorithm for a general graph
  - applies to both directed and undirected graphs
  - compile original graph into a tree of cliques
  - then perform message passing on this tree
- Problem:
  - cost is exponential in size of largest clique
  - many vision models have intractably large cliques

Loopy Belief Propagation

- Apply belief propagation directly to general graph
  - possible because message passing rules are local
  - need to keep iterating
  - might not converge
- State-of-the-art performance in some applications

Max-product Algorithm: most probable x

- Goal: find

$$\mathbf{x}^{\mathsf{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{x})$$

  – define

$$\phi(x_i) = \max_{x_1} \cdots \max_{x_{i-1}} \max_{x_{i+1}} \cdots \max_{x_L} p(x_1, \ldots, x_L)$$

  – then

$$x_i^{\mathsf{MAP}} = \arg\max_{x_i} \phi(x_i)$$

- Message passing algorithm with “sum” replaced by “max”
- Example:
  – Viterbi algorithm for HMMs

Inference and learning

In general: Hidden or latent X (underlying scene) and Observed Y (image)

- Inference: computing P(x|y) (“posterior”)
- Learning: computing P(y) (likelihood) usually $P_\theta(y)$
  (θ : parameter estimation based on ML)

Likelihood of the data y $L(\theta) = P_\theta(y)$

Maximum (log) likelihood

$$\theta_{ML} = \arg\max_\theta \log L(\theta)$$

Example: classification with context

- The labeling problem

★ n objects/individuals $(i \in V = \{1, \ldots, n\})$
★ K labels $(k \in \mathcal{A} = \{1, \ldots, K\})$
★ $n * \ldots$ observations $(y = (y_1, y_2, \ldots))$

assign a label to each object consistently with y:

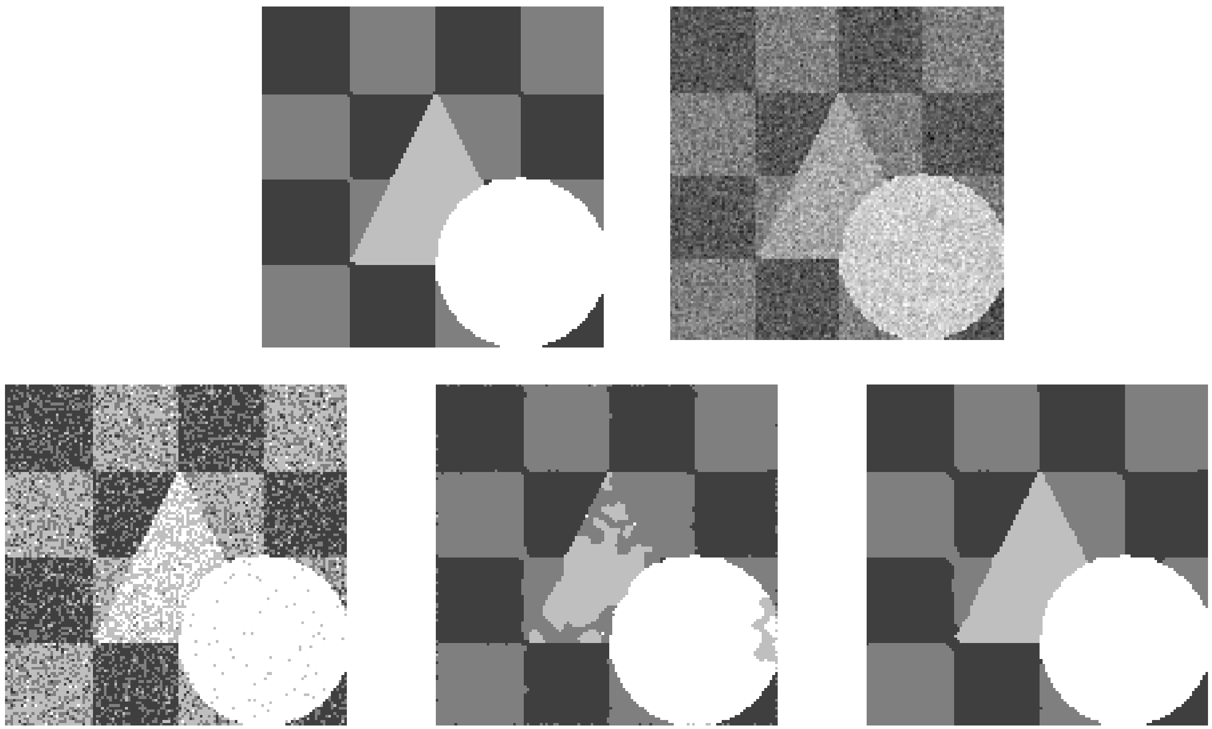
$$\mathbf{x} : V \to \mathcal{A}$$

$$x = (x_1, \ldots, x_n \in \mathcal{A}^n)$$

(assignement, colouring (graph), configuration (random fields)

Contextual constraints: distance, similarity, compatibility, etc.

- Image analysis, segmentation, etc.
- Biometrics: spatially related observations
- Documents analysis: hyperlinks between documents



No context   Too much context   Good compromise

Connection Cost/Energy and probability

★ **assignment cost** $x : V \longrightarrow \mathcal{A}$

$c(i,k)$ [likelihood of k at site i] or $c_y(i,k)$ [data term]

★ **Neighborhood cost**:

i and j *nearby* $\Rightarrow$ x_i and x_j *similar/compatible*

$\rightarrow$ graph $G = (V, E)$: if $(i,j) \in E$

$\rightarrow$ cost $w_{ij} \times d_{ij}(x_i, x_j)$ $\quad [\Psi_{ij}(x_i, x_j)]$

Total cost: $$E(x) = \sum_{i \in S} c(i, x_i) + \sum_{(i,j) \in E} w_{ij} d_{ij}(x_i, x_j)$$

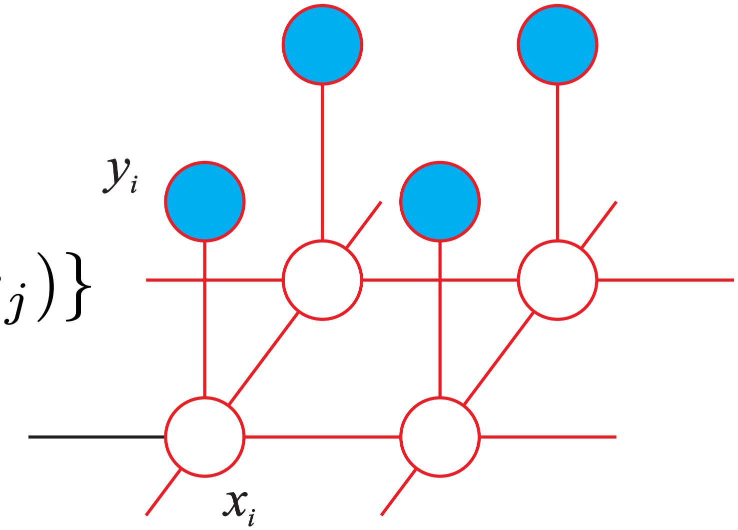
- **Goal: find x that maximizes E**
- Discrete optimization, **NP-hard, find approximations, satisfying assignments**

Optimal configuration for Pairwise MRF with energy E

Energy and MAP rule

- Corresponding graphical model: Pairwise MRF

$$E(x) = \sum_i \{\Psi_i(x_i) + \frac{1}{2} \sum_{j \in N(i)} \Psi_{ij}(x_i, x_j)\}$$



- Maximum A Posteriori (MAP) principle:

$$\hat{x} = \arg\max_{x \in \mathcal{A}^n} P(x|y)$$

Hidden MRF: accounting for observations

- Observations, eg. Measures $Y = \{Y_i, i \in S\}$
- Hidden data, eg. Labels, X discrete MRF $P(x) = \frac{1}{Z}\exp(-E(x))$

- **Data term,** $P(y|x) = \exp(-E(y|x))$

Conditional MRF (posterior): $P(x|y) = \frac{1}{Z_y}\exp(-E_y(x))$

$$E_y(x) = E(x) + E(y|x)$$

E(x): Regularizing term (prior, context)

E(y | x): Data term

MAP solution $\hat{x} = \arg\min_{x \in \mathcal{L}^n} E_y(x)$

Approximate solutions

- Deterministic approaches: relaxation, variational methods (mean field, etc.)
- Stochastic approaches: Gibbs sampling, simulation methods (MC)

- Classification approaches: hard clustering, ICM, K-means
- Parameter estimation approaches: soft clustering, EM

Approximate Inference

For general graphical models (not tree-structured)

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \exp(\Psi_c(x_c))$$

All basic computations are intractable, combinatorial for large G

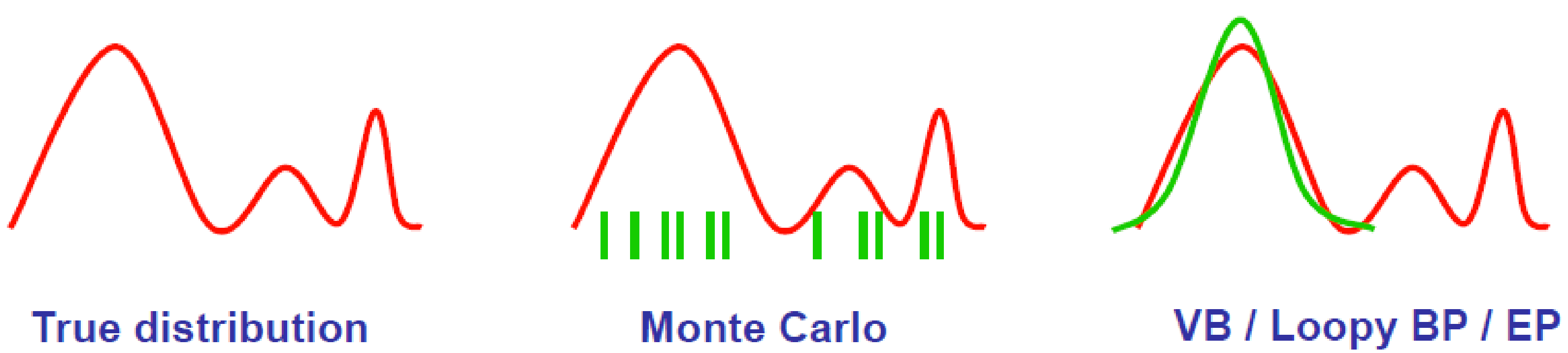
Likelihood and partition function $$Z = \sum_{x \in \mathcal{X}^N} \prod_{c \in \mathcal{C}} \exp(\Psi_c(x_c))$$

Marginals and conditionals $$p(x_j) = \frac{1}{Z} \sum_{x_i, i \neq j} \prod_{c \in \mathcal{C}} \exp(\Psi_c(x_c))$$

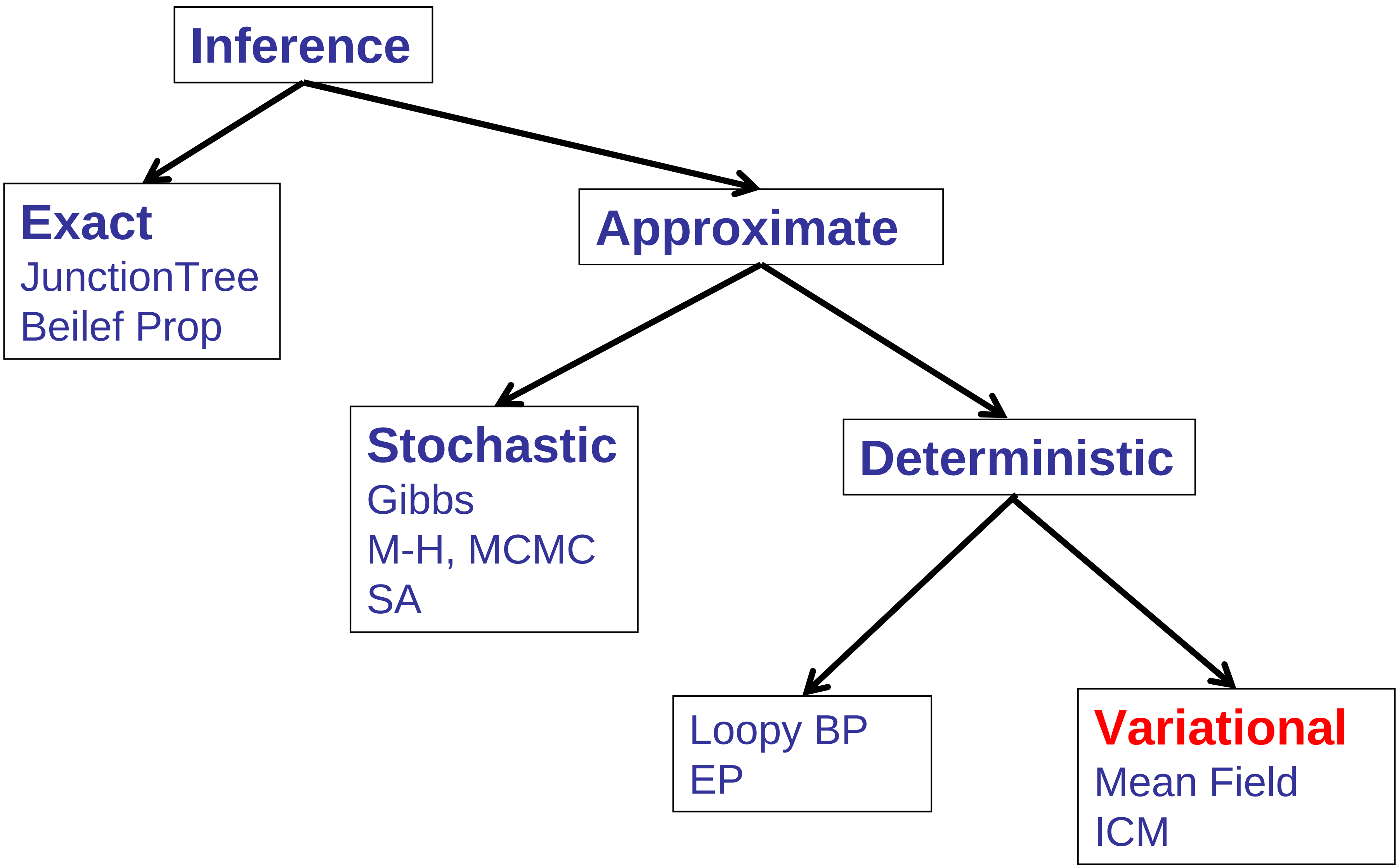
Modes $$x^* = \arg \max_{x \in \mathcal{X}^N} \prod_{c \in \mathcal{C}} \exp(\Psi_c(x_c))$$

Approximate Inference

- Stochastic (Sampling)
  - Metropolis-Hastings, Gibbs, (Markov Chain) Monte Carlo, etc
  - Computationally expensive, but is “exact” (in the limit)
- Deterministic (Optimization)
  - Mean Field (MF), Loopy Belief Propagation (LBP)
  - Variational Bayes (VB), Expectation Propagation (EP)
  - Computationally cheaper, but is not exact (gives bounds)



Taxonomy of inference methods



General View of Variational Inference

- Consider arbitrary distribution $q(x)$ over the latent variables
- The following decomposition always holds

$$\log p(y|\theta) = F(q,\theta) + KL(q,p)$$

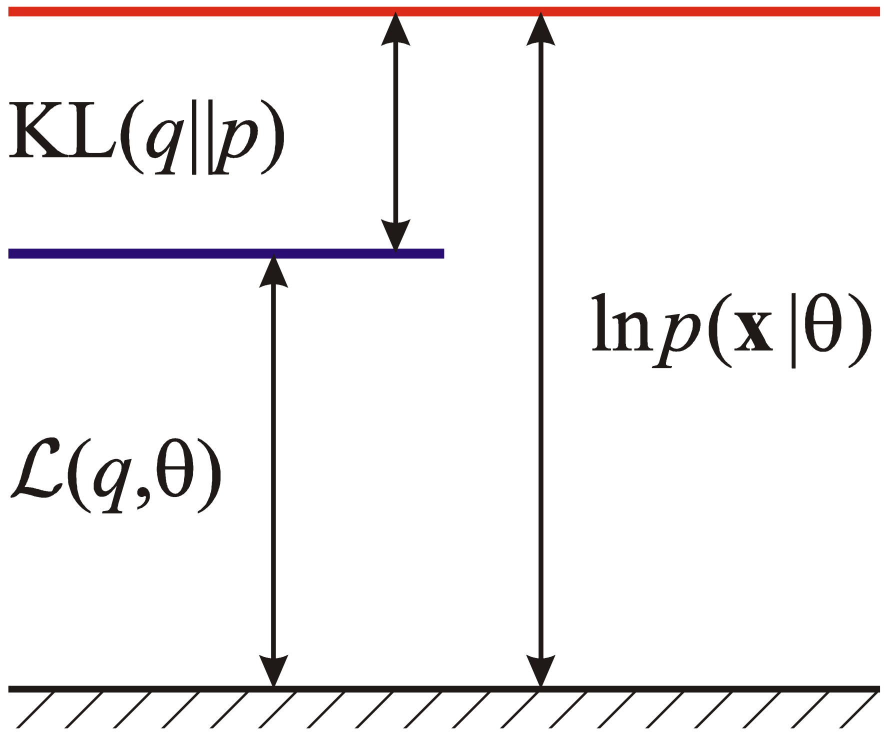
where

$$F(q,\theta) = \sum_x q(x)\ \log \frac{p(x,y|\theta)}{q(x)}$$

$$KL(q,p) = -\sum_x q(x)\ \log \frac{p(x|y,\theta)}{q(x)}$$

Decomposition

Maximizing over $q(x)$ would give the true posterior distribution – but this is intractable by definition



Factorized Approximation

- Goal: choose a family of distributions which are:
  - sufficiently flexible to give good posterior approximation
  - sufficiently simple to remain tractable
- Here we consider factorized distributions

$$q(x) = \prod_i q_i(x_i)$$

- *No further assumptions are required!*
- Optimal solution for one factor, keeping the remained fixed

$$q_j^*(x_j) \propto \exp(\mathbb{E}_{q^*_{\setminus j}}[\log p(y,x)]) \qquad q^*_{\setminus j} = \prod_{i \neq j} q_i^*$$

- Coupled solutions so initialize then cyclically update

Factorized approximation

In practice, we compute $q_j^*(x_j) \propto \exp(\mathbb{E}_{q^*_{\setminus j}}[\log p(x|y)])$

ommiting terms that does not depend on x

and hope to recognize a standard distribution …. or normalize

Ex. Hidden Markov Field

$$p(y|x) = \prod_i p(y_i|x_i)$$

$$p(x) \text{ is a MRF so that } p(x_j|x_{\setminus j}) = p(x_j|x_{N(j)})$$

$$\Longrightarrow \; p(x|y) \propto p(y|x)\; p(x) \propto \prod_i p(y_i|x_i)\; p(x_j|x_{\setminus j}) p(x_{\setminus j})$$

$$\Longrightarrow \boxed{q_j^*(x_j) \propto \exp(\mathbb{E}_{q^*_{\setminus j}}[\log p(y_j|x_j) + \log p(x_j|X_{\setminus j})]))}$$

ommiting terms that does not depend on x_j

Example: Discrete Hidden MRF

$$p(x) = \frac{1}{Z}\exp(E(x)) \text{ with } \quad x_i \in \{1\ldots K\} \text{ and } E(x) = \sum_{i\sim j}\Psi_{ij}(x_i, x_j)$$

$$\Rightarrow p(x_j | x_{\setminus j}) \propto \exp(\sum_{i\in N(j)} \Psi_{ij}(x_i, x_j))$$

$$and$$

$$p(y|x) = \prod_i p(y_i|x_i) \text{ with } \quad p(y_i|x_i = k) = f_{\theta_k}(y_i)$$

$$\rightarrow \quad q_j^*(x_j) \propto p(y_j|x_j)\ \exp(\mathbb{E}_{q^*_{N(j)}}[\sum_{i\in N(j)} \Psi_{ij}(x_i, x_j)])$$

$$q_j^*(x_j) \propto p(y_j|x_j)\ \exp(\sum_{i\in N(j)} \mathbb{E}_{q_i^*}[\Psi_{ij}(x_i, x_j)])$$

Illustration: Ising model, binary MRF

$$\Psi(x_i, x_j) = \theta_{ij}\ x_i x_j \qquad x_i \in \{-1, 1\}$$

Remark: $\Psi(x_i, x_j) = \theta_{ij}(2x_i - 1)(2x_j - 1)$ if $x_i \in \{0, 1\}$

$$\mathbb{E}_{q_i^*}[\Psi(x_i, x_j)] = \theta_{ij}\ x_j \mathbb{E}_{q_i^*}[x_i] = \theta_{ij}\ x_j\ (q_i^*(x_i = 1) - q_i^*(x_i = -1))$$

$$q_j^*(x_j = 1) = \frac{1}{1 + \frac{p(y_j | x_j = -1)}{p(y_j | x_j = 1)} \exp(-2 \sum_{i \in N(j)} \theta_{ij}\ (q_i^*(x_i = 1) - q_i^*(x_i = -1)))}$$

$$q_j^*(x_j = -1) = 1 - q_j^*(x_j = 1)$$

Fixed point equation or iterative updating

Iterated Conditional Modes (ICM) for HMRF

[Besag 70s]

For each j in turn

$$x_j^* = \arg\max_x \; p(y_j | x_j = x) \, p(x_j = x | x^*_{N(j)})$$

$$x_j^* = \arg\max_x \; p(y_j | x_j = x) \; \exp(x \sum_{i \in N(j)} \theta_{ij} x_i^*)$$

A *modal* version of variational *mean* field

Gibbs sampler for HMRF

[Geman & Geman 80s]

A stochastic version of ICM or a simulated version of variational Mean Field

For each j in turn $$x_j^* \sim p(y_j|x_j)\; p(x_j|x^*_{N(j)})$$

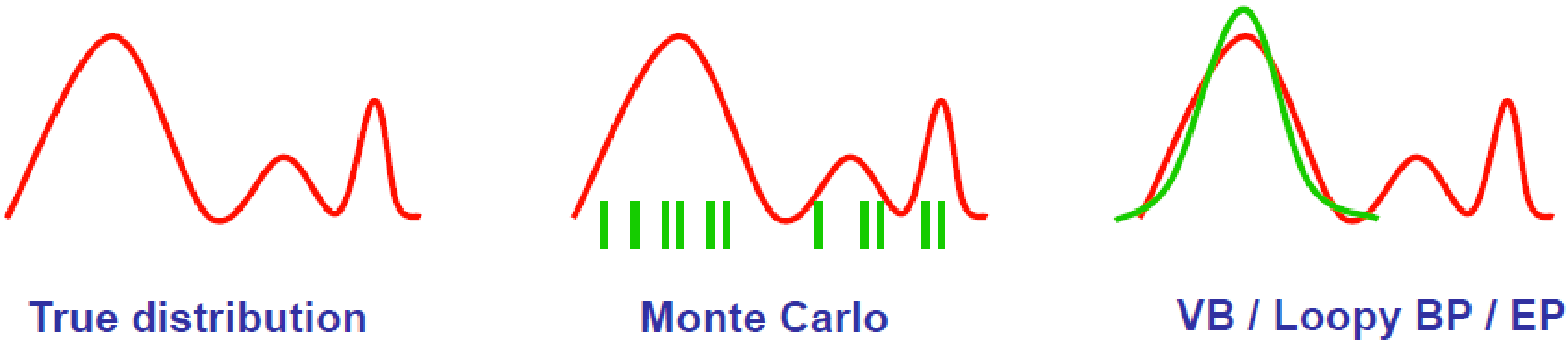
$$x_j^* \sim p(y_j|x_j)\; \exp(x_j \sum_{i\in N(j)} \theta_{ij} x_i^*)$$ (Ising)

$$\text{Sample } \; u \sim Uniform(0,1)$$

$$x_j^* = 1 \text{ if } \quad u \leq \frac{1}{1 + \frac{p(y_j|x_j=-1)}{p(y_j|x_j=1)}\; \exp(-2 \sum_{i\in N(j)} \theta_{ij} x_i^*)}$$

$$x_j^* = -1 \text{ otherwise}$$

Sampling vs Variational approximations



1) MCMC (eg Gibbs sampler)

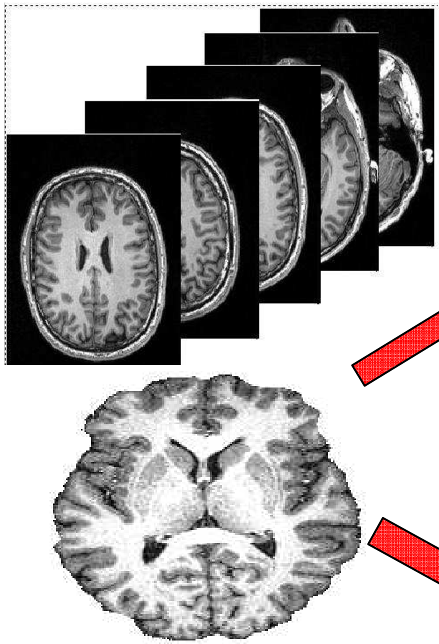
- **Theoretical properties**
- High computational cost
- Complicated convergence monitoring
- Model selection & general noise model: not straightforward

2) Variational (eg VEM)

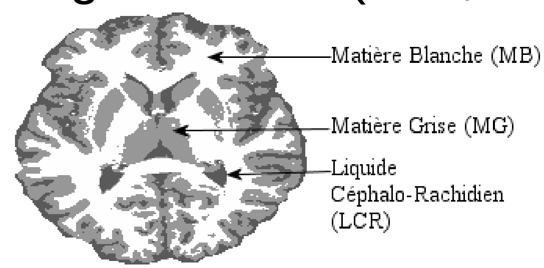
- **Fast and flexible**
- Lack of theoretical properties
- Global covariance structure cannot be estimated

Example 1: MRI Brain scan segmentation

Assign each voxel to a class (label) (among K classes) [Forbes et al 2011]

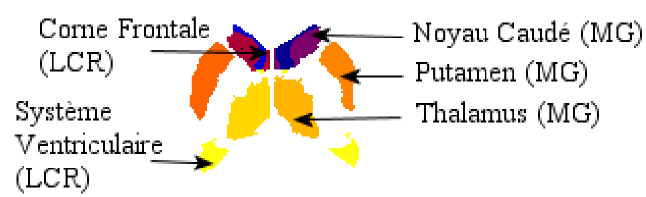


Tissue segmentation (WM, GM, CSF)



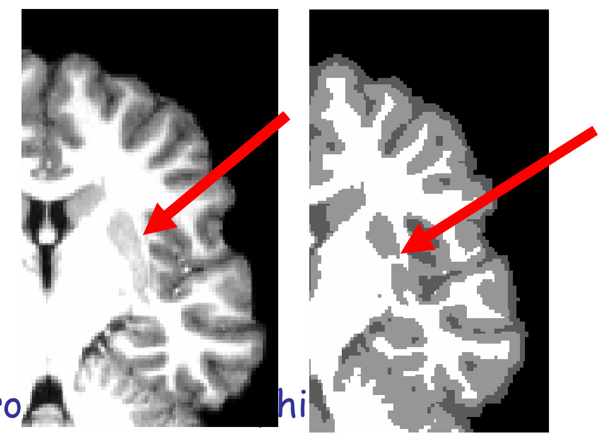
➔ Cortex 3D reconstruction

Structure segmentation

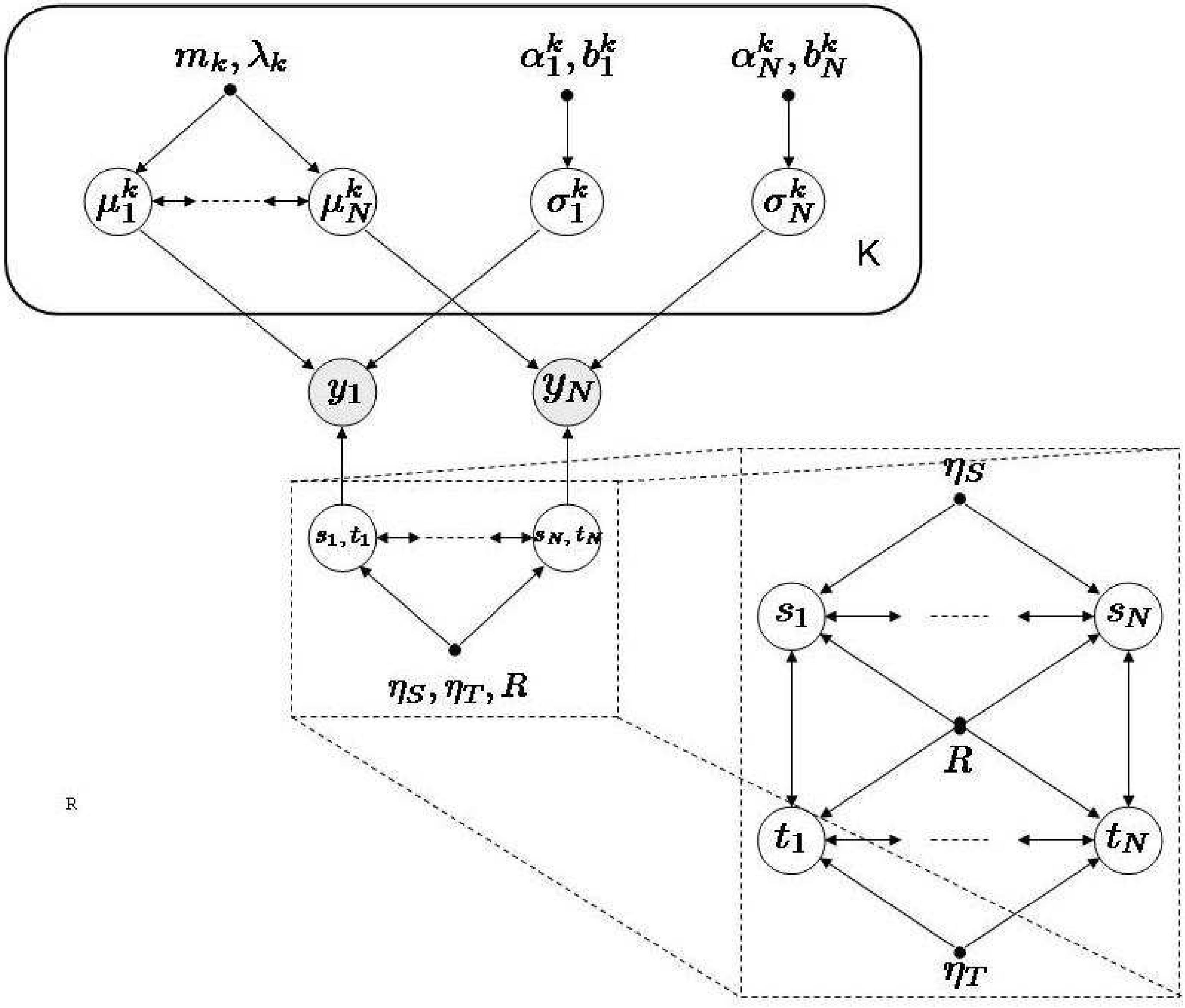


➔Useful for :

- Distinguishing Cortex GM from Nuclei GM
- volumetric studies
- …

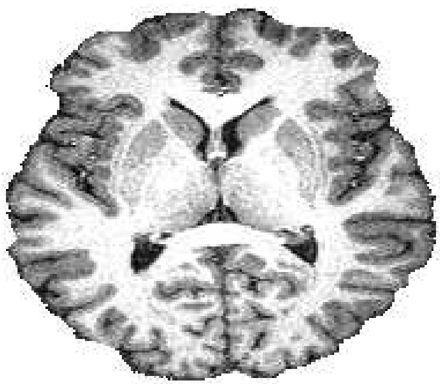


Graphical model representation

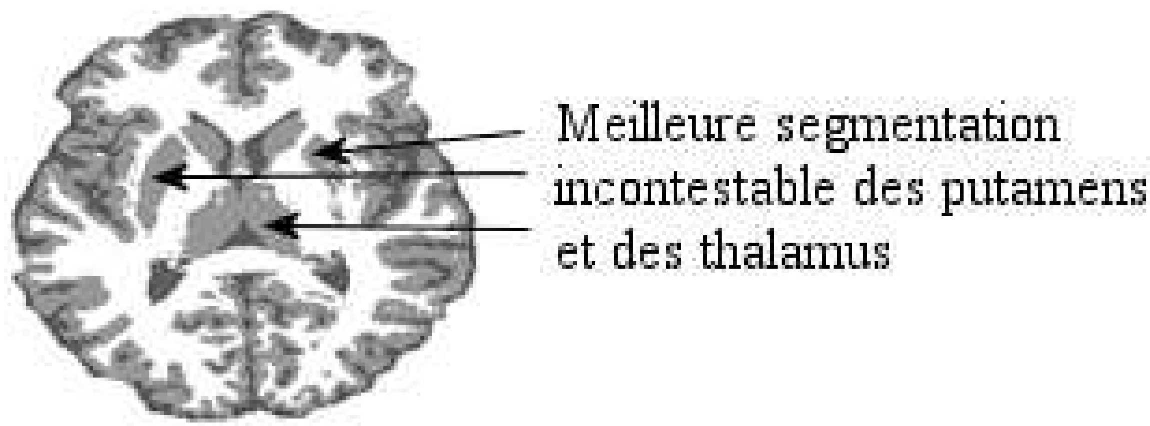


Cooperative segmentation of tissues and structures

observations



No anatomical information

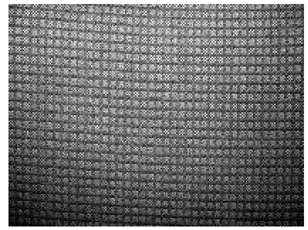


Cooperative method

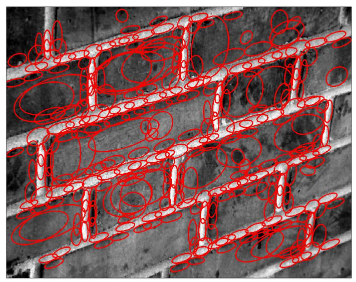
Example 2: texture recognition

[Blanchet & Forbes 2008]

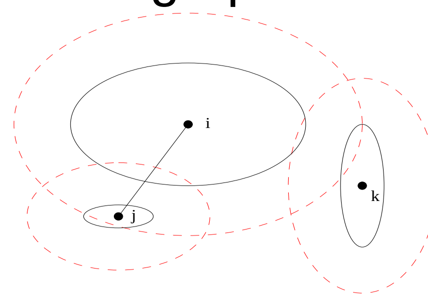
- Learning step: model estimation

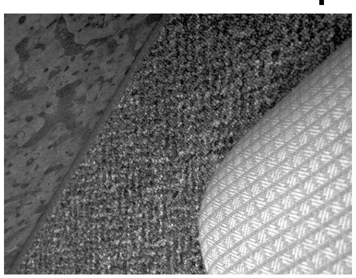
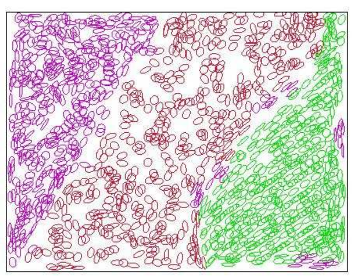

- Interest points

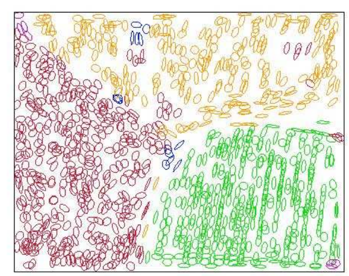


neighborhood graph

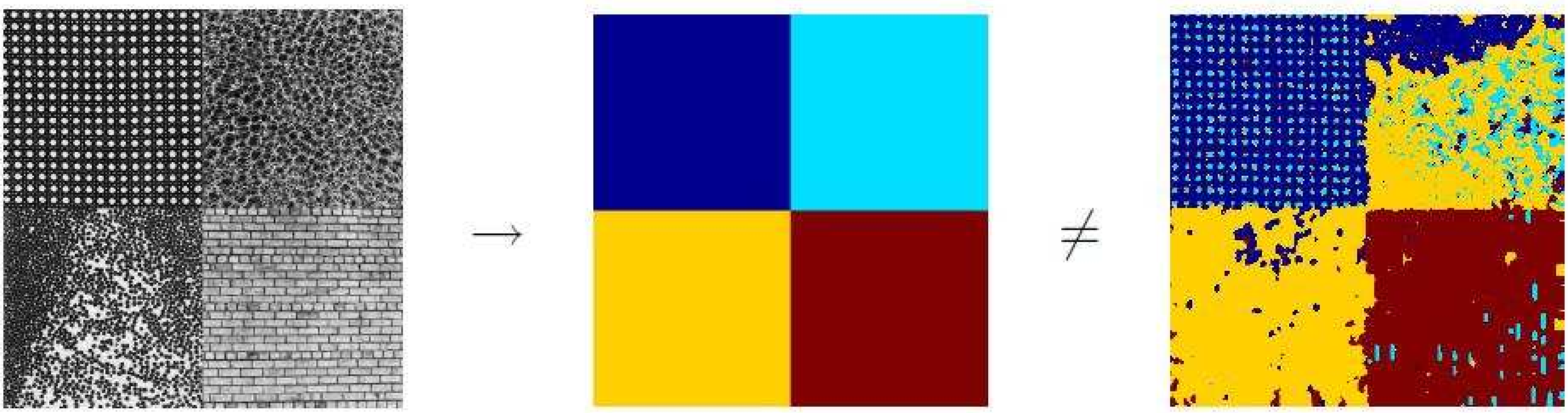


- Test step: classification



Example 2: texture recognition



Thank you for your attention