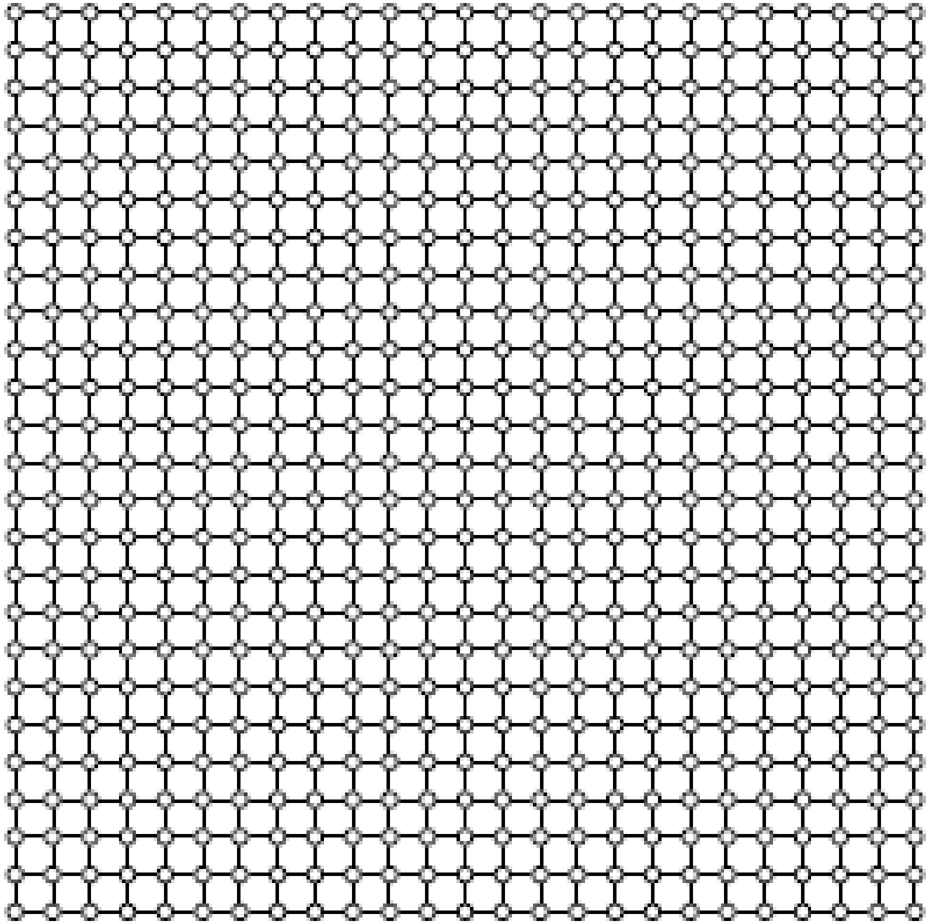
Probabilistic Graphical Models

CVFX

2015.04.23

再看一次範例: MRF



Joint probability

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

state

noisy image

state-state compatibility function

neighboring state nodes

image-state compatibility function

local observations

MAP inference in graphical models

Maximum *a posteriori* :
Find the assignment $\hat{\mathbf{x}} \in \{0,1\}^N$ such that $P(\mathbf{x} = \hat{\mathbf{x}} | \mathbf{y})$ is max

$$p(\mathbf{x}|\mathbf{y} = \bar{\mathbf{y}}) = \frac{1}{Z} \prod_{\{i,j\}} \psi_{ij}(x_i, x_j) \prod_i \phi_{ii}(x_i, \bar{y}_i)$$

Energy functions

- we need to choose energy functions for the cliques
  - a suitable energy function should express the relations among the nodes of a cliques
  - E.g., for image de-noising

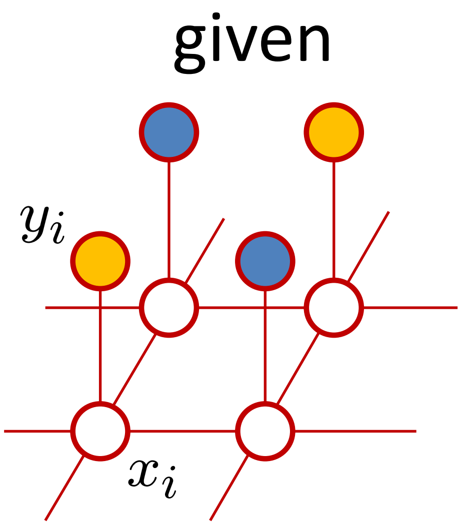
$$E(\mathbf{x}) = \beta \sum_{\{i,j\}} |x_i - x_j| + \eta \sum_i |x_i - \bar{y}_i|$$

$$\Downarrow$$

$$p(\mathbf{x}|\bar{\mathbf{y}}) = \frac{1}{Z} \exp\{-E(\mathbf{x})\}$$

minimizing energy = maximizing probability

Binary pixel labeling as energy minimization



$$E(\mathbf{x}) = \beta \sum_{\{i,j\}} |x_i - x_j| + \eta \sum_i |x_i - \bar{y}_i|$$

$$x_i = \{0, 1\}$$

Find assignment $\mathbf{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$ such that $E(\mathbf{x})$ is "minimized"

MAP vs. energy minimization

prior likelihood

$$p(\mathbf{x}|\mathbf{y}=\bar{\mathbf{y}}) = \frac{1}{Z}\prod_{\{i,j\}} \psi_{ij}(x_i, x_j) \prod_i \phi_{ii}(x_i, \bar{y}_i)$$

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

smoothness terms data terms

$$E(\mathbf{x}) = \sum_{\{i,j\}} V_{ij}(x_i, x_j) + \sum_i D_i(x_i)$$

Inference

- conditional probability query
- MAP

- exact inference
  - variable elimination
  - message passing for trees

- approximate inference

Inference

A set of factors Φ defines an unnormalized function $P_\Phi(X) = \prod_{\phi \in \Phi} \phi$.

Conditional probability queries

- evidence: $\mathbf{E} = \mathbf{e}$
- query: a subset of variables $\mathbf{Y}$
- task: compute $P_\Phi(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \frac{P_\Phi(\mathbf{Y},\mathbf{e})}{P_\Phi(\mathbf{e})}$

NP-hardness

The following problem is NP-hard:

- given a graphical model P_Φ, a variable X, and a value $x \in Val(X)$, compute $P_\Phi(X = x)$.

Sum-product

$P_\Phi(\mathbf{Y}, \mathbf{E} = \mathbf{e}) = \sum_{\{X_1,\ldots,X_n\} - \mathbf{Y} - \mathbf{E}} \frac{1}{Z} \prod_k \phi'_k(D'_k)$ (reduced factors)

Algorithms: conditional probability

Exact

- push summations into factor product
  - variable elimination, dynamic programming

General

- message passing over a graph
  - belief propagation
  - variational approximation
- random sampling
  - Markov Chain Monte Carlo (MCMC)
  - importance sampling

Inference

A set of factors Φ defines an unnormalized function $P_\Phi(X) = \prod_{\phi \in \Phi} \phi$.

MAP (maximum a posteriori)

- evidence: $\mathbf{E} = \mathbf{e}$
- query: all other variables $\mathbf{Y} = \{X_1, \ldots, X_n\} - \mathbf{E}$
- task: compute $\mathrm{MAP}(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \arg\max_{\mathbf{y}} P_\Phi(\mathbf{Y} = \mathbf{y}|\mathbf{E} = \mathbf{e})$

NP-hardness

The following problem is NP-hard:

- given a graphical model P_Φ and a number τ, decide whether there exists an assignment $\mathbf{x}$ to $\mathbf{X}$ such that $P_\Phi(\mathbf{x}) > \tau$.

Max-product

$P_\Phi(\mathbf{Y} = \mathbf{y}|\mathbf{E} = \mathbf{e}) \propto P_\Phi(\mathbf{Y}, \mathbf{E} = \mathbf{e})$
$P_\Phi(\mathbf{Y}, \mathbf{E} = \mathbf{e}) = \frac{1}{Z} \prod_k \phi'_k(D'_k) \propto \prod_k \phi'_k(D'_k)$ (reduced factors)
$\arg\max_{\mathbf{y}} P_\Phi(\mathbf{Y} = \mathbf{y}|\mathbf{E} = \mathbf{e}) = \arg\max_{\mathbf{y}} \prod_k \phi'_k(D'_k)$

Algorithms: MAP

Exact

- push maximization into factor product
  - variable elimination

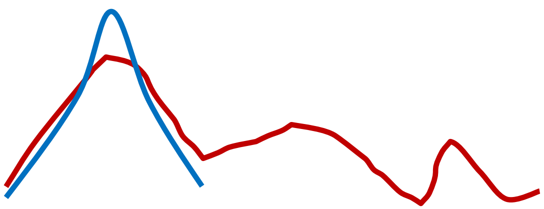
General

- message passing over a graph
  - max-product belief propagation
- using methods for integer programming
- for some networks: graph-cut methods
- combinatorial search

Inference as optimization

Optimization framework

- Define a surrogate class of 'easy' distributions $\mathcal{Q}$, and search for a particular instance Q within that class which is the 'best' approximation to the target distribution P_Φ. Queries can then be done by inference on Q rather than on P_Φ.
- Approximate P_Φ with Q: choose Q to be close to P_Φ.
  - how to measure the distance between two distributions: relative entropy (KL-divergence)
  - how to optimize the distance



Relative entropy

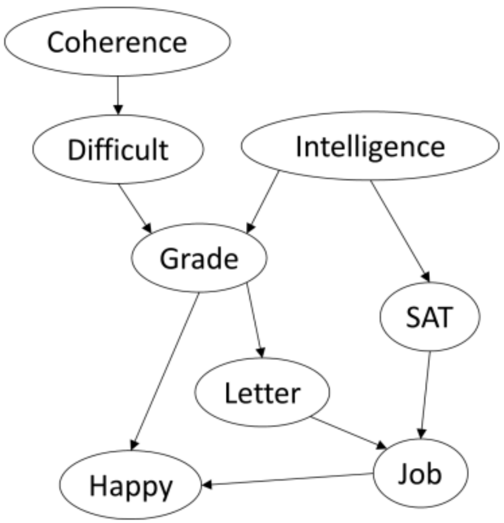
$$KL(P_1 \| P_2) = \mathbb{E}_{P_1}\left[\log \frac{P_1(X)}{P_2(X)}\right].$$

It is asymmetric, always nonnegative, and equal to 0 if and only if $P_1 = P_2$.

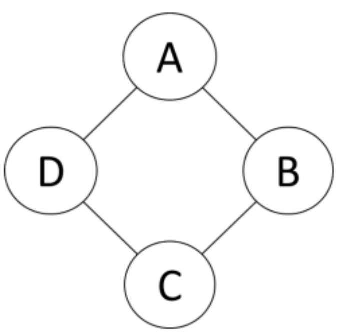
Exact inference: variable elimination

Two examples

- variable elimination in Bayesian networks

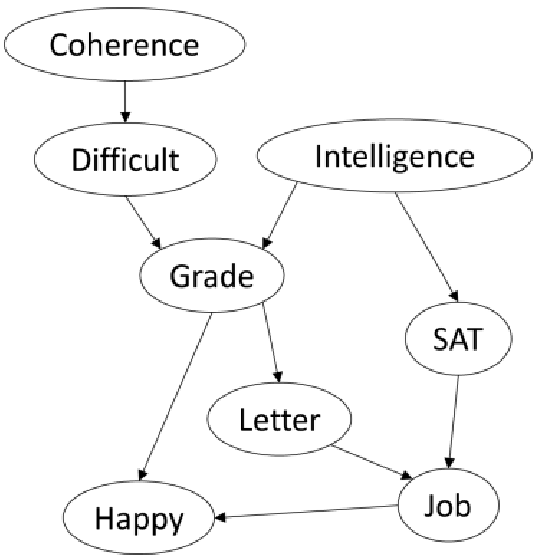


- variable elimination in Markov networks



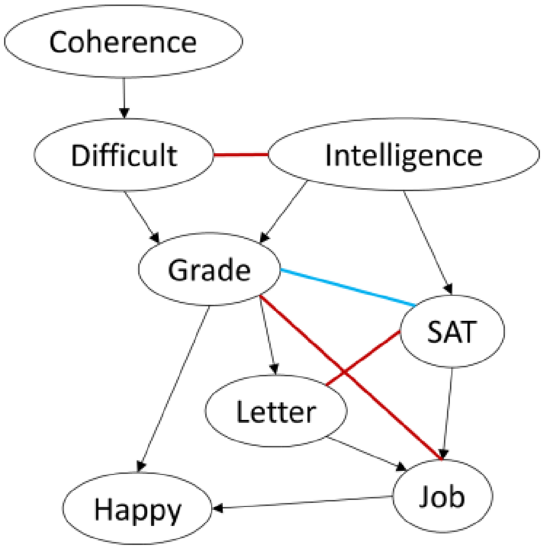
Variable elimination in BNs

- goal: $P(J)$
- eliminate: C, D, I, H, G, S, L



$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\,\phi_L(L,G)\,\phi_S(S,I)$$
$$\phi_G(G,I,D)\,\phi_H(H,G,J)\,\phi_I(I)\,\phi_D(C,D)\,\phi_C(C)$$

Step by step

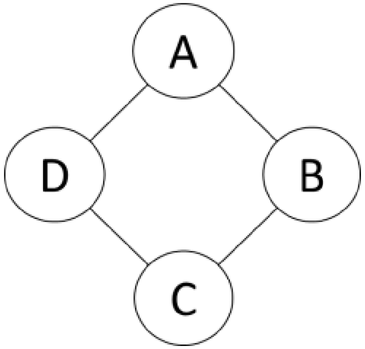


moralization

- C: $\tau_1(D) = \sum_C \phi_D(C, D)\, \phi_C(C)$
- D: $\tau_2(G, I) = \sum_D \phi_G(G, I, D)\, \tau_1(D)$
- I: $\tau_3(S, G) = \sum_I \phi_S(S, I)\, \phi_I(I)\, \tau_2(G, I)$
- H: $\tau_4(G, J) = \sum_H \phi_H(H, G, J)$
- G: $\tau_5(L, S, J) = \sum_G \phi_L(L, G)\, \tau_4(G, J)\, \tau_3(S, G)$
- L, S: $\tau_6(J) = \sum_{L,S} \phi_J(J, L, S)\, \tau_5(J, L, S)$

Variable elimination in MNs

- goal: $P(D)$
- eliminate: A, B, C



$$\sum_{A,B,C} \phi_1(A,B)\,\phi_2(B,C)\,\phi_3(C,D)\,\phi_4(A,D)$$

- A: $\tau_1(B,D) = \sum_A \phi_1(A,B)\,\phi_4(A,D)$
- B: $\tau_2(C,D) = \sum_B \phi_2(B,C)\,\tau_1(B,D)$
- C: $\tau_3(D) = \sum_C \phi_3(C,D)\,\tau_2(C,D) = \widetilde{P}(D) \propto P(D)$
- At the end of elimination, renormalize $\tau_3(D)$ to get $P(D)$.

Variable elimination: summary

VE algorithm

1. Reduce all factors by evidence, get a set of factors Φ;
2. For each non-query variable Z, eliminate Z from Φ;
3. Multiply all remaining factors;
4. Renormalize to get a distribution.

VE properties

- simple algorithm, works for both BNs and MNs
- can be done in any order, subject to when Z is eliminated
- complexity of VE is linear in
  - size of the model
  - size of the largest factor generated
- size of factor is exponential in its scope
  - **depends heavily on elimination order**
  - **finding the optimal elimination ordering is NP-hard**

Belief propagation

Local message passing for *trees*

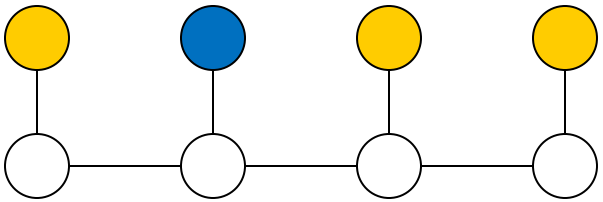
- sum-product algorithm
  - find marginals

- max-product algorithm
  - find a setting of the variables that has the larges probability

- exact inference in trees

- converge in finite time

Sum-product algorithm



Input: Graph, $\psi_{ij}(x_i, x_j)$, $\phi_{ii}(x_i, y_i)$

$m_{ij}(x_j)$: message that x_i sends to x_j

$b_i(x_i)$: belief at node x_i

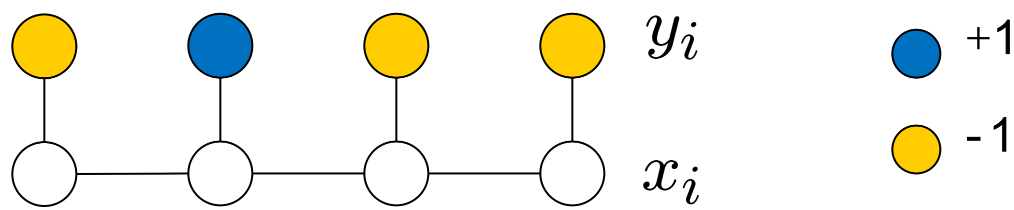
Iterate :

$$m_{ij}(x_j) \leftarrow \alpha \sum_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{x_k \in \mathcal{N}(x_i) \setminus x_j} m_{ki}(x_i)$$

Finally:

$$b_i(x_i) \leftarrow \alpha\phi_i(x_i) \prod_{x_j \in \mathcal{N}(x_i)} m_{ji}(x_i)$$

Example



$$\psi_{ij}(x_i, x_i) = ke^{0.6 x_i x_j}$$

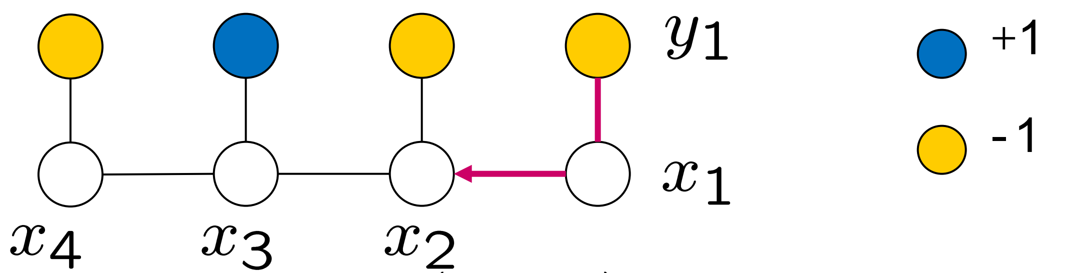
$$\phi_{ii}(x_i, y_i) = ke^{x_i y_i}$$

$$\psi_{ij}(x_i, x_j) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}$$

$$\phi_{ii}(x_i, +1) : \begin{array}{cc} \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix} & \begin{array}{c} +1 \\ -1 \end{array} \end{array}$$

$$\phi_{ii}(x_i, -1) : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

Example



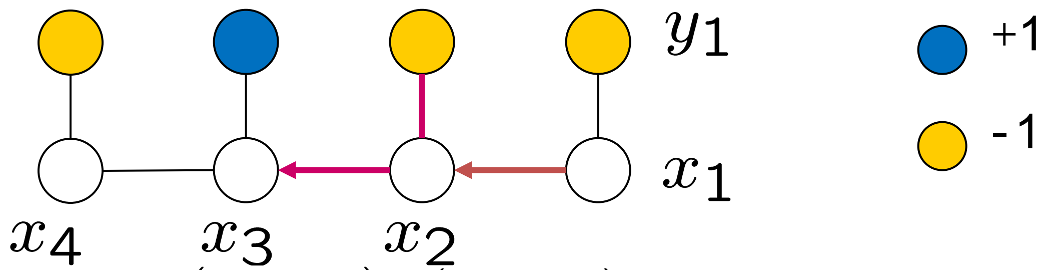
$$\begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$\psi_{12}(x_1, x_2) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_1(x_1, -1) : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$m_{12}(x_2) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} .* \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \right\} = \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix}$$

$$\boxed{m_{12}(x_2) \leftarrow \alpha \sum_{x_1} \psi_{12}(x_1, x_2) \phi_1(x_1) \prod_{x_k \in \mathcal{N}(x_1) \setminus x_2} m_{k1}(x_1)}$$

Example



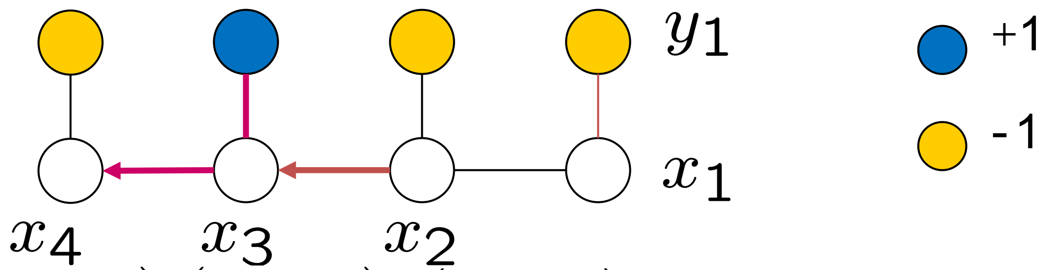
$$\begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$\psi_{23}(x_2, x_3) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_2(x_2, -1) : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$m_{23}(x_3) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} .* \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \right\} = \begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix}$$

$$m_{23}(x_3) \leftarrow \alpha \sum_{x_2} \psi_{23}(x_2, x_3) \phi_2(x_2) \prod_{x_k \in \mathcal{N}(x_2) \backslash x_3} m_{k2}(x_2)$$

Example



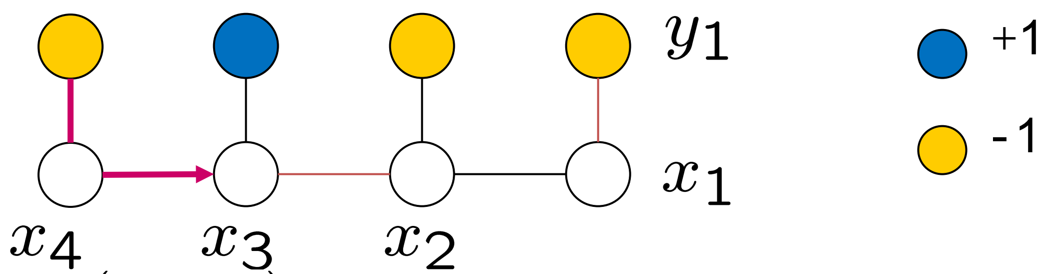
$$\begin{pmatrix} 0.62 \\ 0.38 \end{pmatrix} \begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$\psi_{34}(x_3, x_4) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_3(x_3, +1) : \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix}$$

$$m_{34}(x_4) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix} .* \begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix} \right\} = \begin{pmatrix} 0.62 \\ 0.38 \end{pmatrix}$$

$$\boxed{m_{34}(x_4) \leftarrow \alpha \sum_{x_3} \psi_{34}(x_3, x_4) \phi_3(x_3) \prod_{x_k \in \mathcal{N}(x_3) \backslash x_4} m_{k3}(x_3)}$$

Example



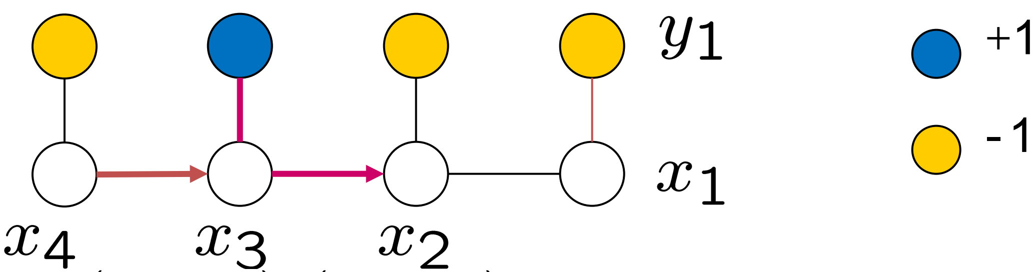
$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix}$$

$$\psi_{43}(x_4, x_3) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_4(x_4, -1) : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$m_{43}(x_3) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} .* \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \right\} = \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix}$$

$$\boxed{m_{43}(x_3) \leftarrow \alpha \sum_{x_4} \psi_{43}(x_4, x_3) \phi_4(x_4) \prod_{x_k \in \mathcal{N}(x_4) \backslash x_3} m_{k4}(x_4)}$$

Example



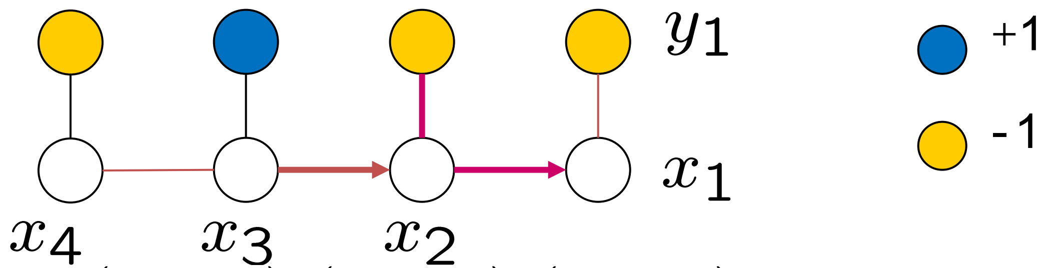
$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}\begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix}\begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix}$$

$$\psi_{32}(x_3, x_2) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_3(x_3, +1) : \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix}$$

$$m_{32}(x_2) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix} .* \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \right\} = \begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix}$$

$$\boxed{m_{32}(x_2) \leftarrow \alpha \sum_{x_3} \psi_{32}(x_3, x_2) \phi_3(x_3) \prod_{x_k \in \mathcal{N}(x_3) \setminus x_2} m_{k3}(x_3)}$$

Example



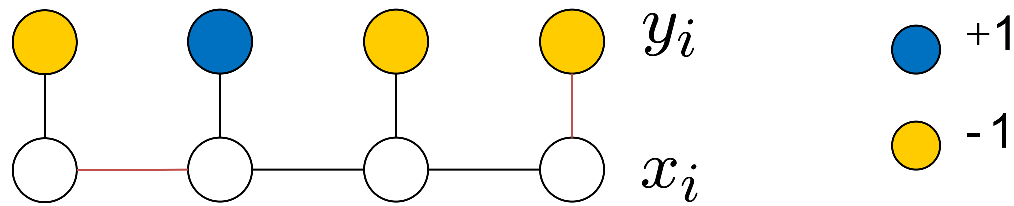
$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix} \begin{pmatrix} 0.33 \\ 0.67 \end{pmatrix}$$

$$\psi_{21}(x_2, x_1) : \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix} \qquad \phi_2(x_2, -1) : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$m_{21}(x_1) : \alpha \begin{pmatrix} 0.77 & 0.23 \\ 0.23 & 0.77 \end{pmatrix}^T \left\{ \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} .* \begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix} \right\} = \begin{pmatrix} 0.33 \\ 0.67 \end{pmatrix}$$

$$\boxed{m_{21}(x_1) \leftarrow \alpha \sum_{x_2} \psi_{21}(x_2, x_1) \phi_2(x_2) \prod_{x_k \in \mathcal{N}(x_2) \backslash x_1} m_{k2}(x_2)}$$

Example



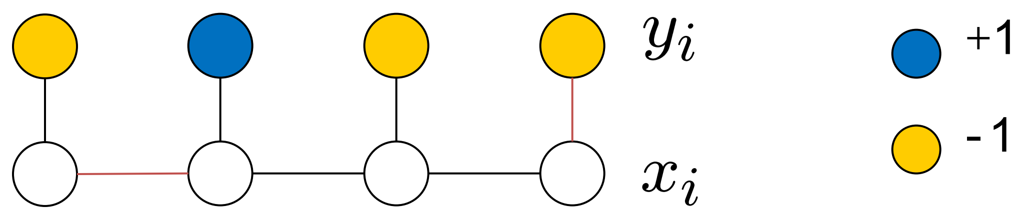
$$m_{i-1,i}(x_i): \begin{pmatrix} 0.62 \\ 0.38 \end{pmatrix} \begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$m_{i+1,i}(x_i): \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix} \begin{pmatrix} 0.33 \\ 0.67 \end{pmatrix}$$

$$\phi_4 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} \quad \phi_3 : \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix} \quad \phi_2 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} \quad \phi_1 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$\boxed{b_i(x_i) \leftarrow \alpha \phi_i(x_i) \prod_{x_j \in \mathcal{N}(x_i)} m_{ji}(x_i)}$$

Example



$$\begin{pmatrix} 0.62 \\ 0.38 \end{pmatrix} \begin{pmatrix} 0.26 \\ 0.74 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \longleftarrow m_{i-1,i}(x_i)$$

$$m_{i+1,i}(x_i) \longrightarrow \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \begin{pmatrix} 0.29 \\ 0.71 \end{pmatrix} \begin{pmatrix} 0.63 \\ 0.37 \end{pmatrix} \begin{pmatrix} 0.33 \\ 0.67 \end{pmatrix}$$

$$\phi_4 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} \quad \phi_3 : \begin{pmatrix} 0.88 \\ 0.12 \end{pmatrix} \quad \phi_2 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix} \quad \phi_1 : \begin{pmatrix} 0.12 \\ 0.88 \end{pmatrix}$$

$$b_4 : \begin{pmatrix} 0.18 \\ 0.82 \end{pmatrix} \quad b_3 : \begin{pmatrix} 0.51 \\ 0.49 \end{pmatrix} \quad b_2 : \begin{pmatrix} 0.09 \\ 0.92 \end{pmatrix} \quad b_1 : \begin{pmatrix} 0.06 \\ 0.95 \end{pmatrix}$$

Max-product algorithm

- find a setting of the variables that has the larges probability
  - Maximum *a posteriori* (MAP) probabilities

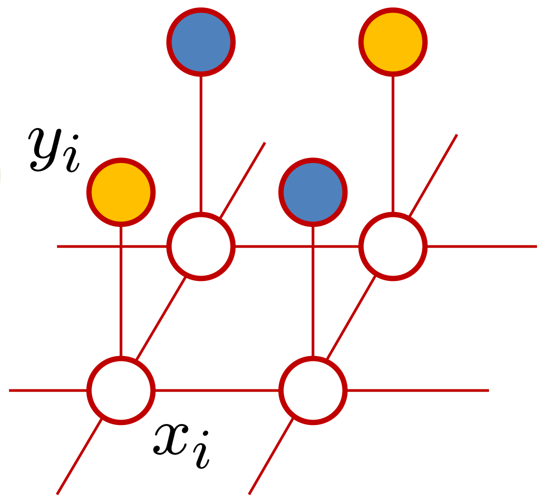
$$m_{ij}(x_j) \leftarrow \alpha \max_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{x_k \in \mathcal{N}(x_i) \setminus x_j} m_{ki}(x_i)$$

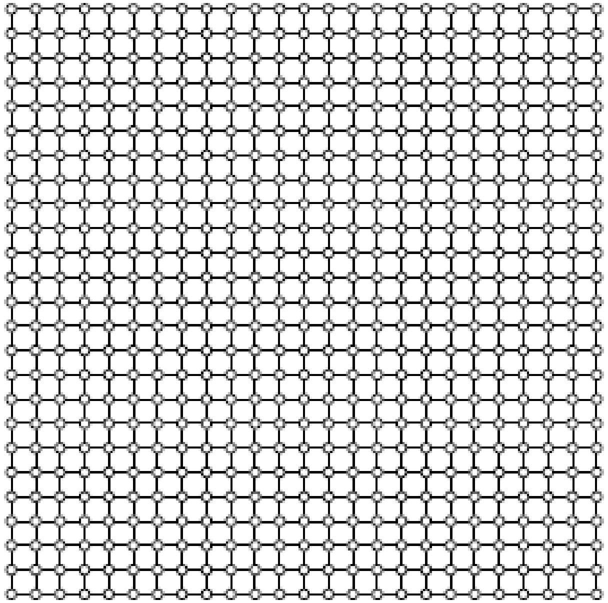
Local message passing for trees

- sum-product algorithm
  - find marginals
- max-product algorithm
  - find a setting of the variables that has the larges probability
- exact inference in trees
- converge in finite time

PRML, Chris Bishop

MRF of image is not a tree





Convert an arbitrary graph into a tree

- NP-hard problem

- junction-tree algorithm
  - clique trees

Approximate inference

- sampling methods
  – Monte Carlo methods
- variational approaches

- loopy belief propagation
  – ignore the existence of loops and run the algorithm as if the graph is a tree
  – the algorithm may never converge
  – however, in practice it is generally found to converge within a reasonable time for most applications

Software

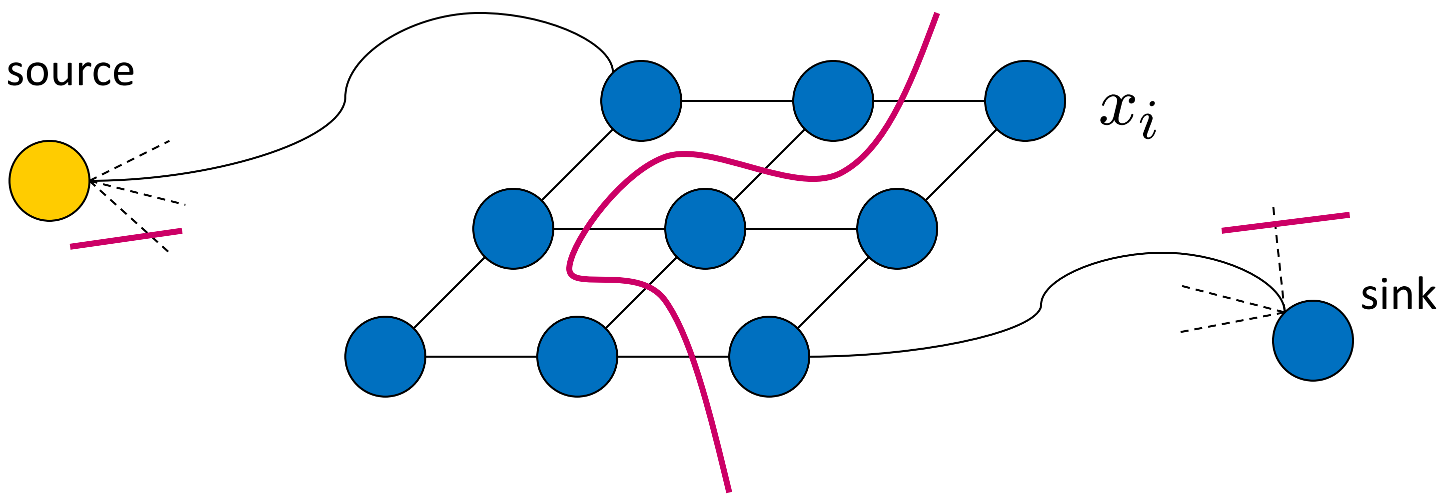
- tree-reweighted message passing and belief propagation
  - http://research.microsoft.com/en-us/downloads/dad6c31e-2c04-471f-b724-ded18bf70fe3/
  - http://vision.middlebury.edu/MRF/code/

- Bayes Net toolbox
  - http://code.google.com/p/bnt/

Energy minimization as a min-cut problem

$$E(\mathbf{x}) = \sum_{\{i,j\}} V_{ij}(x_i, x_j) + \sum_i D_i(x_i)$$

energy terms ← → costs on the edges



Graph cuts

- Binary labeling problems on MRFs can be solved via energy minimization

- If the energy function satisfies the regularity requirement, we can construct a graph such that finding the min-cut is equivalent to minimizing the energy
  - max-flow/min-cut algorithms are fast
  - global optimum for binary labeling
    - Multiple labels?

Multiple labels

- Multiple labels
  - Alpha expansion (or expansion move)
  - Alpha-beta swap (or swap move)
  - *Fast Approximate Energy Minimization via Graph Cuts*
    - Yuri Boykov, Olga Veksler, and Ramin Zabih
    - ICCV '99

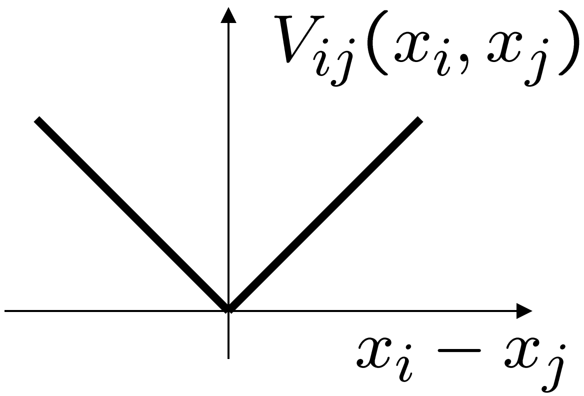
Software

- Min-Cut/max-flow algorithms for energy minimization in computer vision
  - http://pub.ist.ac.at/~vnk/software.html
  - http://vision.middlebury.edu/MRF/code/
- Matlab wrapper for graph cuts
  - http://vision.csd.uwo.ca/code/

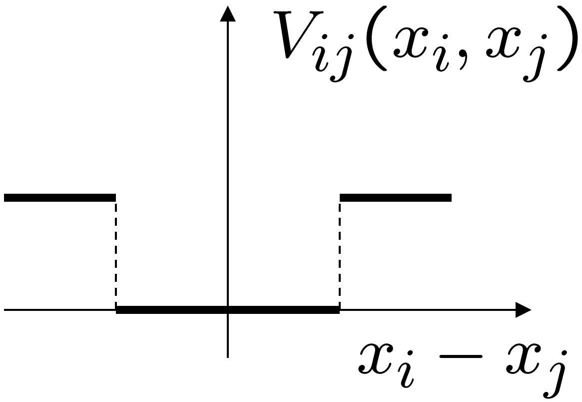
Summary

- inference 在做甚麼?
- 有哪些演算法?

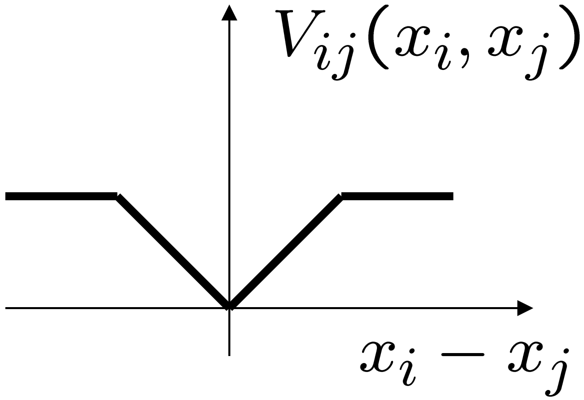
Major types of smoothness priors



everywhere smooth prior



piecewise constant prior



piecewise smooth prior

The partition function

The partition function Z:

$$\ln Z = F[P_\Phi, Q] + KL(Q \| P_\Phi),$$

where $F[P_\Phi, Q]$ is the energy functional
$F[P_\Phi, Q] = \sum_{\phi \in \Phi} \mathbb{E}_Q[\ln \phi] + H_Q(X)$.
Mimimizing the relative entropy $KL(Q \| P_\Phi)$ is equivalent to maximizing the energy functional $F[P_\Phi, Q]$ of which the second term is referred to as the Helmholtz free energy.

- **Computing the partition function is often the hardest part of inference.**
- $KL(Q \| P_\Phi) > 0$, $\ln Z > F[P_\Phi, Q]$: **the energy functional is the lower bound of the logarithm of the partition function** Z**. If we have a good approximation** $KL(Q \| P_\Phi)$**, we can get a good lower bound aprroximation to** Z**.**