Graphical Models

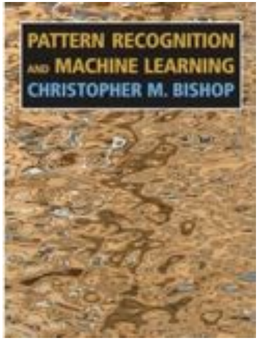
Bayesian Networks and Markov Random Fields

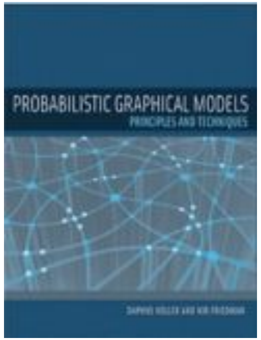
Zhiwei Xu
zhiwei.xu@anu.edu.au

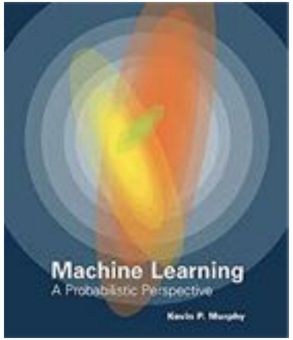
ENGN8501 Guest Lectures
27 September, 2021

THE AUSTRALIAN NATIONAL UNIVERSITY

Reference Reading





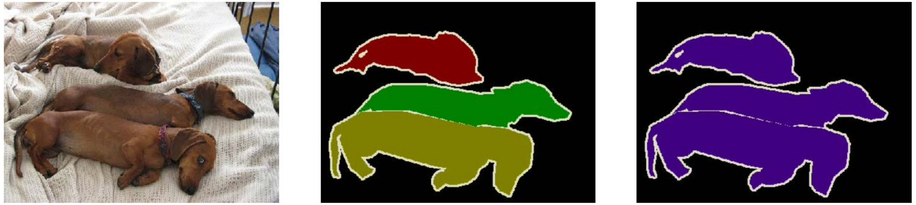


- Pattern Recognition and Machine Learning, *chapter 8*
- Graphical Models and Belief Propagation, *2010 Chapter 7*
- Markov Random Fields and Images, *1998*
- Application of the Mean Field Methods to MRF Optimization Computer Vision, *2012*
- Loopy Belief Propagation in Image-Based Rendering, *2007*
- Markov Random Fields and Gibbs Sampling for Image Denoising, *2018*
- http://www.cs.toronto.edu/~fleet/courses/2503/fall11/Handouts/mrf.pdf

Contents

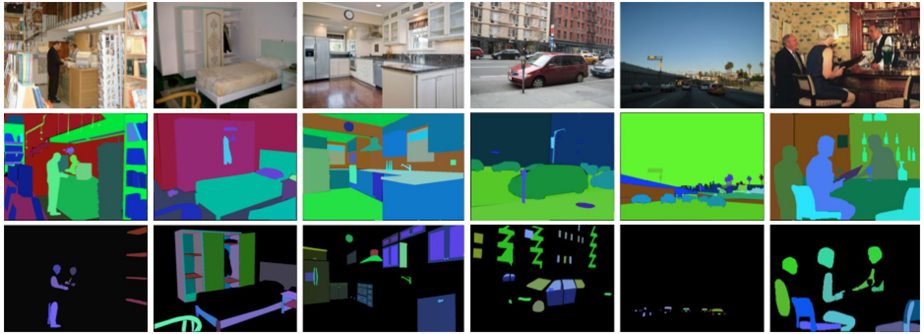
Graphical Models

Overview

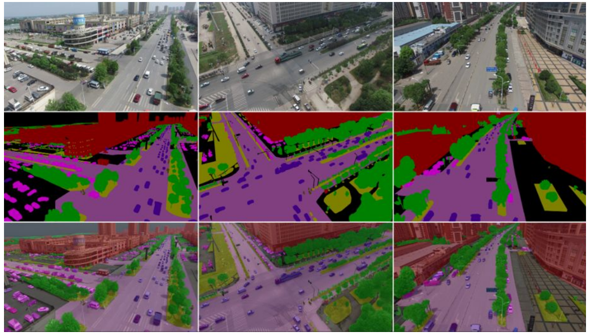


(a) PASCAL VOC 2012

(b) cityscapes



(c) ADE20K



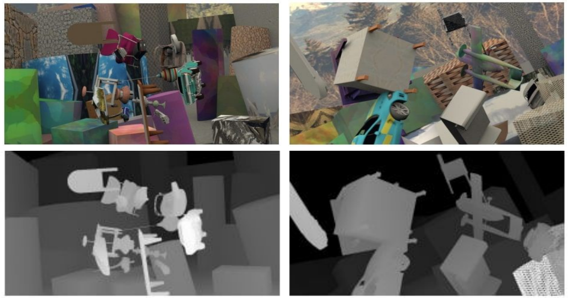
(d) UAVid

Fig. 1 Semantic and instance segmentation

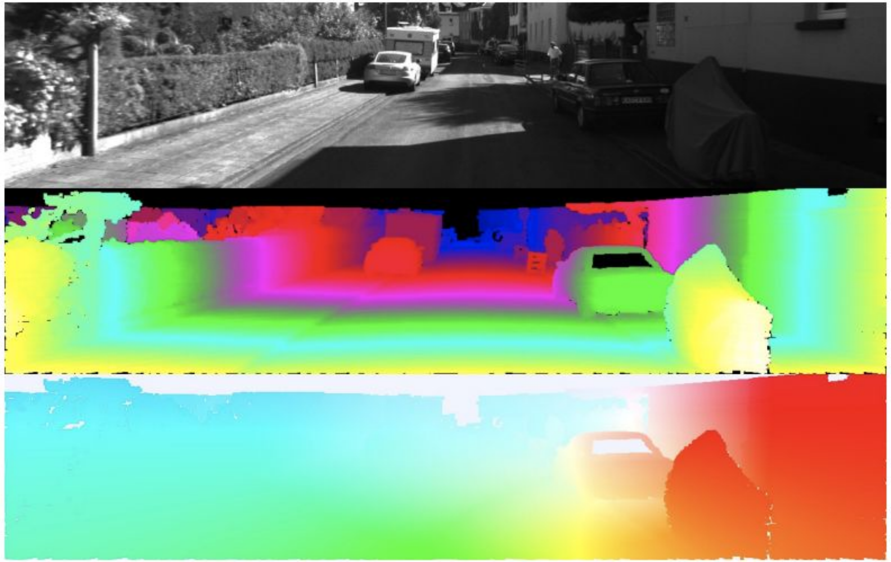
Graphical Models

Overview

(a) Middlebury



(b) SceneFlow

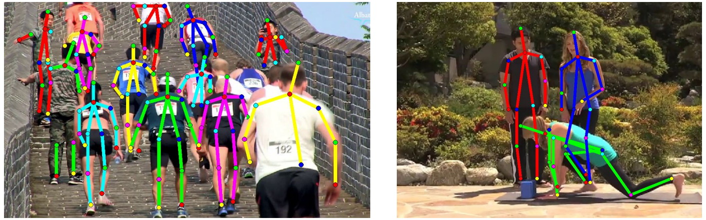


(c) KITTI 2015

Fig. 2 Stereo vision and optical flow

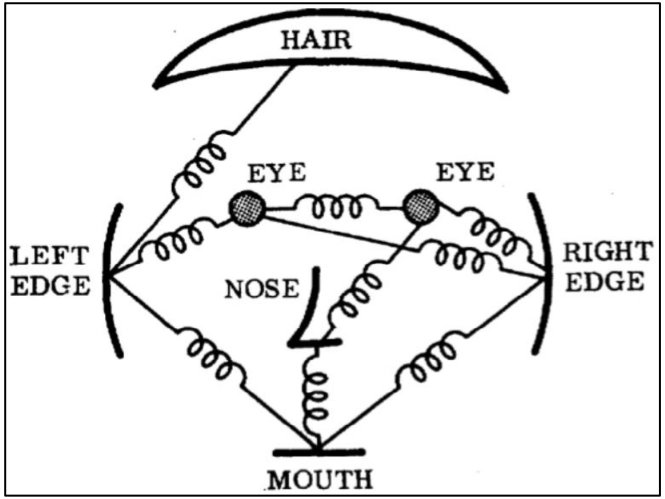
Graphical Models

Overview



(a) MPII human pose

(b) VGG human pose



(c) human face parts

Fig. 3 Human pose and facial expression estimation

Graphical Models

Random variables and probability

- *Random variables*: defined on a set Ω that, a variable X taking a value x in Ω has a certain probability $P(X=x)$, viewed as a random phenomenon.

- *Independent random variables*: given two random variables X_1, X_2 defined on space $\Omega_1 \times \Omega_2$ independent if the joint probability satisfies

$$P(X_1, X_2) = P(X_1)\,P(X_2)$$

Graphical Models

Random variables and probability

- *Joint random variables*: defined on a Cartesian product space $\Omega = \Omega_1 \times \ldots \times \Omega_n$ that, a random variable $\mathbf{X}$ containing n elements takes a value $\mathbf{x} = \left(x_1, \ldots, x_n\right)$ for a random variable X_i in $\mathbf{X}$ with value $x_i = \lambda \in \Omega_i$ has a marginal probability

$$P\left(X_i = \lambda\right) = \sum_{\mathbf{x} \in \Omega, x_i = \lambda} P(\mathbf{x})$$

Graphical Models

Random variables and probability

- *Conditional random variables*: given two independent random variables X_1, X_2 that are conditioned on a random variable X_3, the conditional probability is

$$P(X_1, X_2 | X_3) = P(X_1 | X_3)\, P(X_2 | X_3)$$

Graphical Models

What is a graph in graph theory?

- Vertex (node): a vertex $v \in V$ is incident with a head of an edge $e \in E$; two vertices v_1, v_2 are adjacent if they are incident with the same edge.
- Edge (arc): a directed edge is an ordered pair of vertices while an undirected edge is simply a connection between vertices.
- Graph: a graph consists of vertices and edges with a certain structure. A directed graph is with all edges that are directed; an undirected graph has no edges that are incident.

Graphical Models

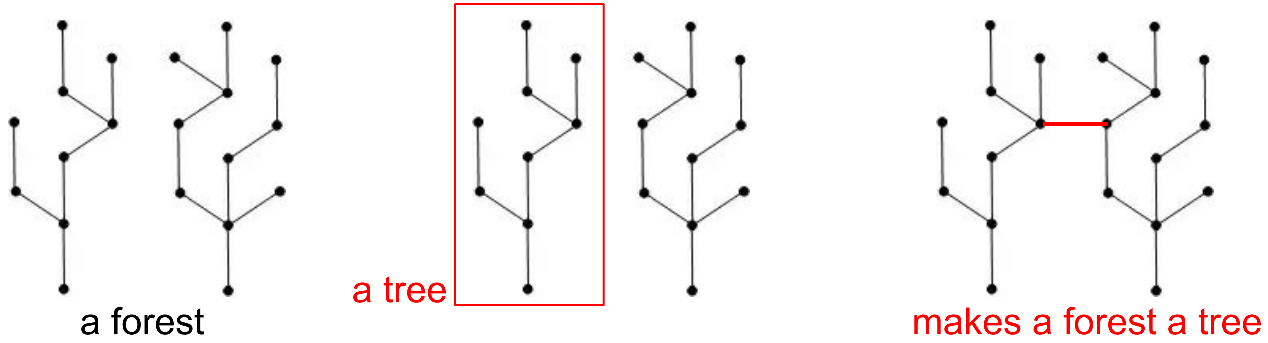
What is a graph in graph theory?

- Path: a (directed) path is a sequence of vertices, where consecutive vertices are neighbours in the graph. A graph is cyclic if its endpoints are equal; an acyclic graph has no cycles in the path.

Graphical Models

What is a graph in graph theory?

- Forest, tree, DAG: a forest is an undirected, acyclic, and disconnected graph. A tree can be a connected forest. DAG is a directed acyclic graph.



Graphical Models

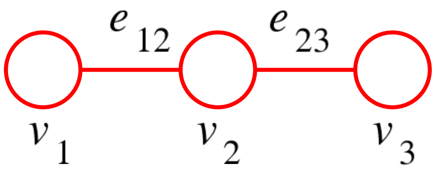
What is a graph in graph theory?

a graph, G

vertices (nodes), $V = \{v_1, \ldots, v_n\}$

edges (arcs), $E = \{e_{ij}\}, \ \forall\, i, j \in V$

$$G = (V, E)\ , \ E \subset V \times V$$



Graphical Models

Distinguish graph types (directed, undirected, cyclic, acyclic)

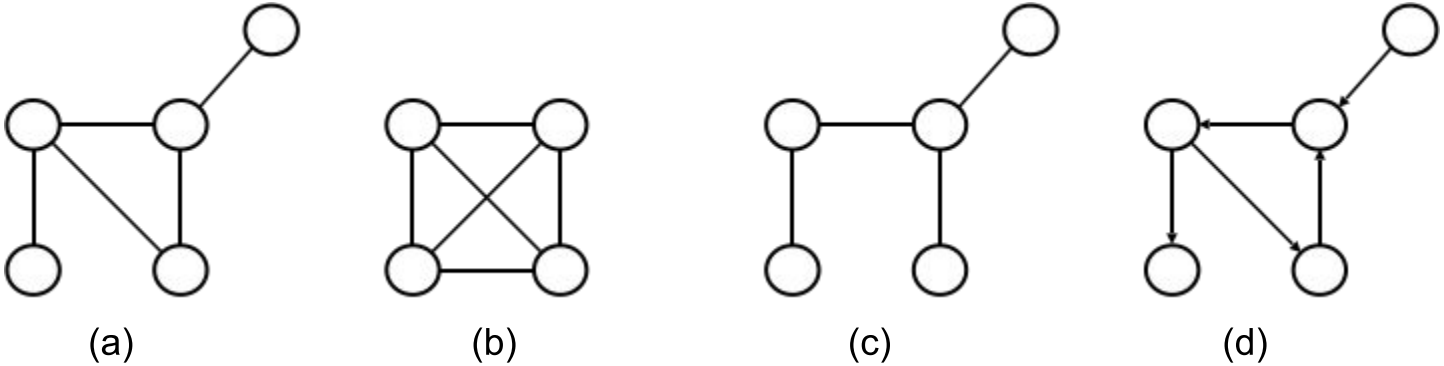


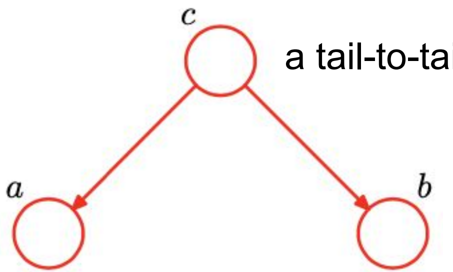
Fig. 4 Graph types

Bayesian Networks

Bayesian Networks

Conditional independence

- Left: random variables a, b are not independent when variable c is unobserved
- Right: random variables a, b are independent conditioned on variable c observed

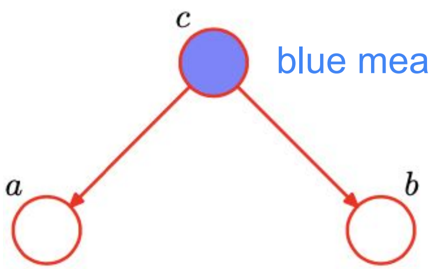


a tail-to-tail node

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

cannot be factorized into $p(a)\,p(b)$

$$\Longrightarrow \quad a \not\!\perp\!\!\!\perp b \mid \emptyset$$



blue means observed

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

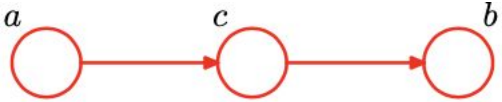
$$\Longrightarrow \quad a \perp\!\!\!\perp b \mid c$$

Bayesian Networks

Conditional independence

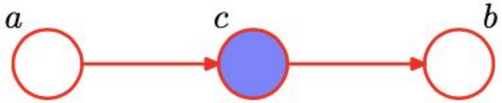
- Left: random variables a, b are not independent when variable c is unobserved
- Right: random variables a, b are independent conditioned on variable c observed

a head-to-tail node



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

$$\Rightarrow \quad a \not\perp\!\!\!\perp b \mid \emptyset$$



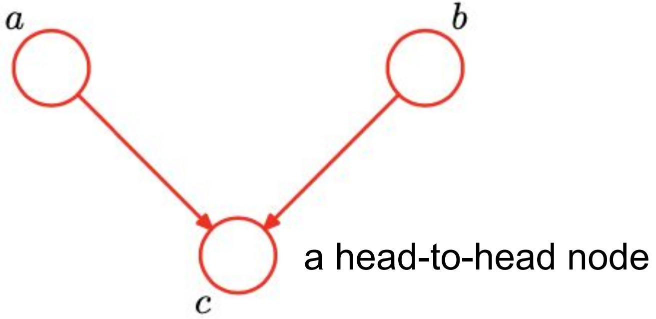
$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

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

Bayesian Networks

Conditional independence

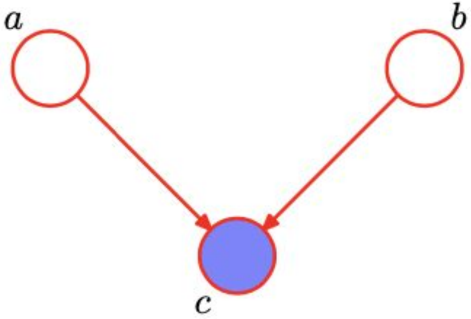
- Left: random variables a, b are independent conditioned on variable c unobserved
- Right: random variables a, b are not independent when variable c observed



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

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

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



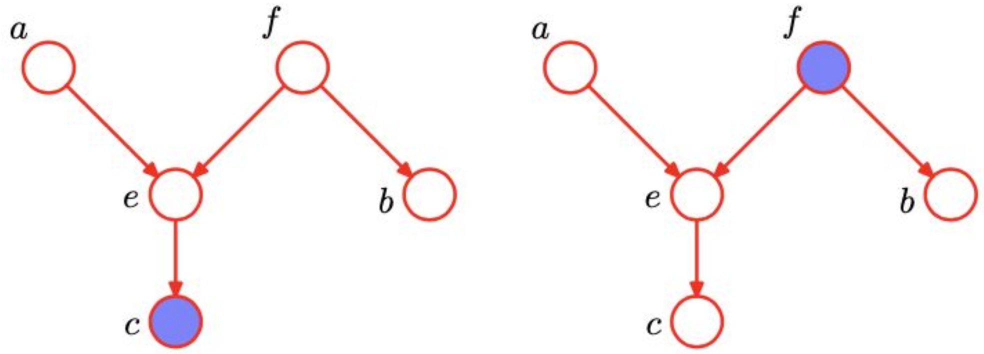
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

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

Bayesian Networks

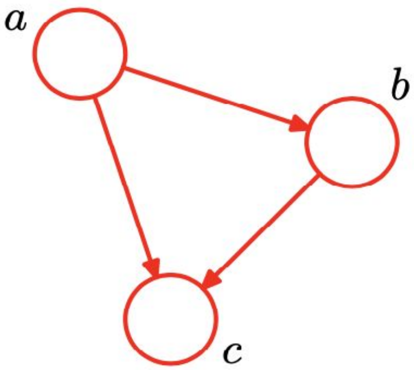
D-separation: a, b are independent and conditioned on in a DAG where any of the paths from a to b is *blocked* when either or both of the following two are satisfied.

- the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the condition set c observed
- the arrows meet head-to-head at the node, and neither the node, nor any of its descendants is in the condition set c observed

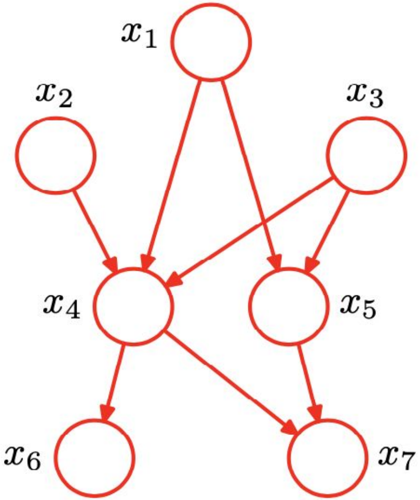


Bayesian Networks

Directed graphical models



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



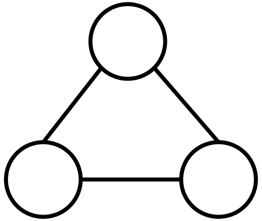
$$p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)$$
$$p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Fig. 5 Joint probability distribution on a DAG, dependent and independent random variables

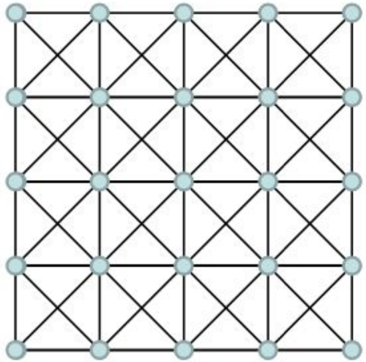
Markov Random Fields

Markov Random Fields

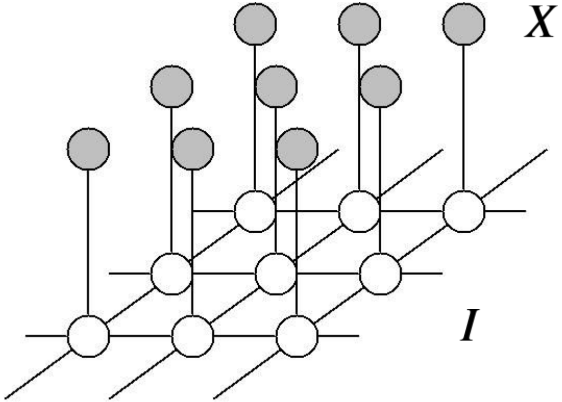
Undirected graphical models



(a) undirected graph



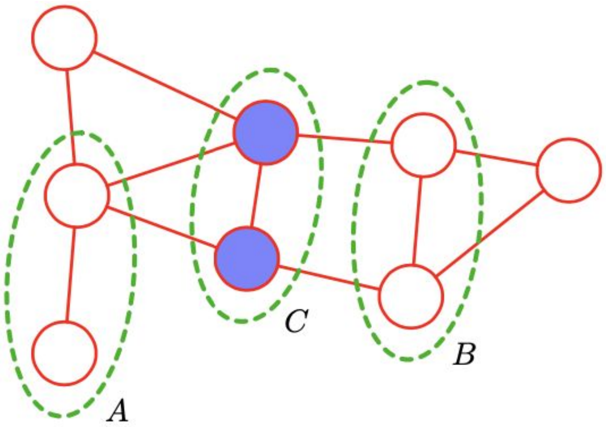
(b) a 8-connected MRF



(c) labeling ***X*** based on image intensity ***I***

Markov Random Fields

The status of a node/clique only depends on its neighbours



potential function

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

$$\boxed{Z} = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

partition function

- Clique sizes: 1, 2, 3
- Maximal clique size: 3
- Cliques *A* and *B* are independent and conditioned on *C*

Markov Random Fields

Energy minimization versus probability maximization

Energy function: $E(\mathbf{x}, \theta) = \sum_C \theta_C(\mathbf{x}_C)$

Joint distribution: $P(\mathbf{x}, \theta) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$, $Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$

Relation: $\psi_C(\mathbf{x}_C) = \exp(-\theta_C(\mathbf{x}_C))$

Minimize an energy function (min-sum) is equivalent to maximize its joint probability (sum-product or max-sum).

Markov Random Fields

Pairwise energy function

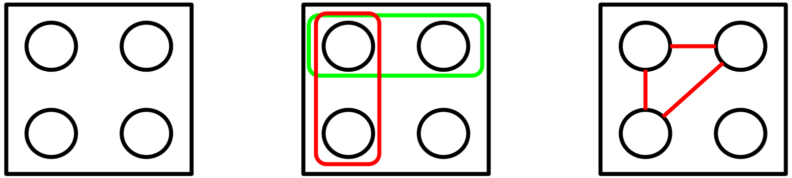
$$E(\mathbf{x},\theta) = \sum_{\mathbf{c}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})$$

Discrete: $E(\mathbf{x},\theta) = \boxed{\sum_{i\in\nu}\theta_i(x_i)} + \boxed{\sum_{(i,j)\in\varepsilon}\theta_{ij}(x_i,x_j)} + \boxed{\sum_{|c|\geq 3}\theta_c(\mathbf{x}_c)}$

unary pairwise higher-order

Real-valued:

$$E(\mathbf{q},\theta) = \sum_{i\in\nu}\sum_{\lambda\in L}\theta_i(\lambda)\,q_i(\lambda) + \sum_{(i,j)\in\varepsilon}\sum_{\lambda,\mu\in L}\theta_{ij}(x_i,x_j)\,q_i(\lambda)\,q_j(\mu) + \sum_{|c|\geq 3}\sum_{\boldsymbol{\varphi}\in L^{|c|}}\theta_c(\boldsymbol{\varphi})\prod_{i\in v_c} q_i(\boldsymbol{\varphi})$$



Markov Random Fields

Conditional Random Fields (CRFs, conditional distribution)

CRF is a special case of MRFs

$$P(\mathbf{X}\boxed{|\mathbf{I}}) = \frac{1}{Z(\mathbf{I})} \exp(-\textstyle\sum_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(\mathbf{X}_c|\mathbf{I})) \qquad E(\mathbf{x}|\mathbf{I}) = \textstyle\sum_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(\mathbf{x}_c|\mathbf{I})$$

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \qquad \psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\textstyle\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)}$$

$$k(\mathbf{f}_i, \mathbf{f}_j) = \underbrace{w^{(1)} \exp\left(-\frac{\boxed{|p_i - p_j|}^2}{2\theta_\alpha^2} - \frac{\boxed{|I_i - I_j|}^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + \underbrace{w^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}$$

more details: *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*, NIPS 2011

Inference in Graph Models

Inference in Graph Models

Revisit joint probability distribution, for inference on a chain

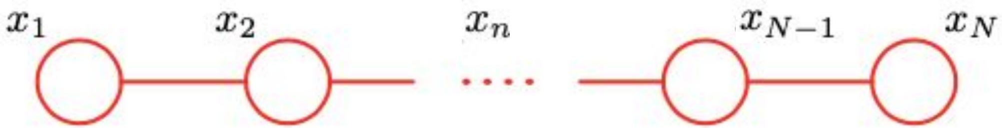
potential function over clique

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

partition function

Inference in Graph Models

Revisit marginal probability distribution, for inference on a chain



$$p(x_n) = \frac{1}{Z}$$

$$\underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right]\right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right] \cdots \right]}_{\mu_\beta(x_n)}$$

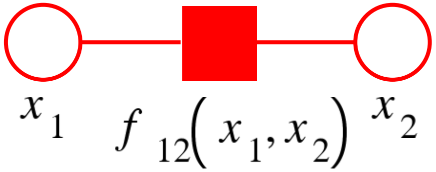
fix the node, and flow from the two ends to this node

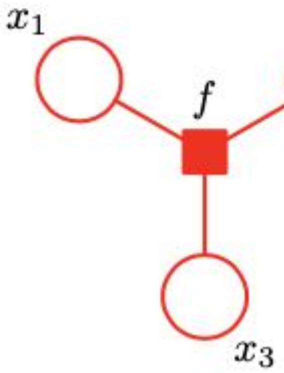
Inference in Graph Models

A factor graph is *bipartite* due to the two node types: variable and factor.

Graph *factorization* over all subsets $\mathbf{x}_s$ of variables.

joint: $p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s) \equiv p(\mathbf{x}) = \prod_{s \in \mathrm{ne}(x)} F_s(x, X_s)$ marginal: $p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x})$

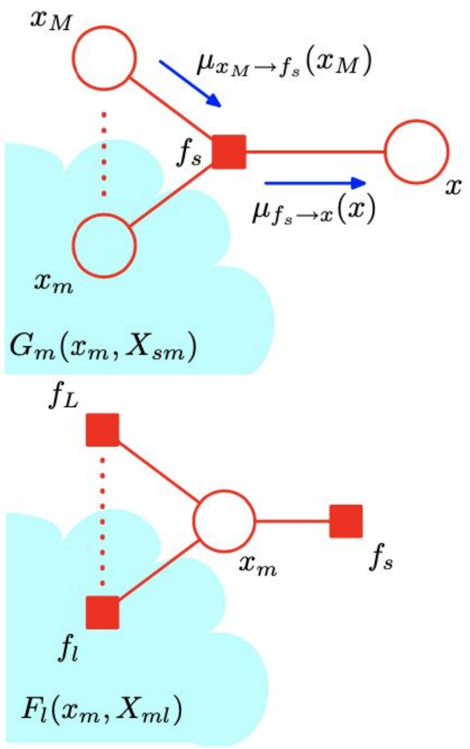




$f_{12}(x_1, x_2)$ easily understood as unary term of x_1 by unary term of x_2 by pairwise term of (x_1, x_2)

Inference in Graph Models

Sum-product algorithm (marginal distribution)



$$p(x) = \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)$$

$$\begin{aligned}\mu_{f_s \to x}(x) &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \left[\sum_{X_{xm}} G_m(x_m, X_{sm}) \right] \\ &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)\end{aligned}$$

$$\begin{aligned}\mu_{x_m \to f_s}(x_m) &= \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} \left[\sum_{X_{ml}} F_l(x_m, X_{ml}) \right] \\ &= \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)\end{aligned}$$

Inference in Graph Models

Max-sum algorithm (maximum a posterior)

$$p(x^{max}) = \max_x p(x)$$

Recall: $$\ln\left(\prod_i x_i\right) = \ln\left(\prod_i \exp(\ln(x_i))\right) = \sum_i \ln(x_i)$$

rename variables

$$\boxed{p(x) = \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)} \Longrightarrow \boxed{q(x) = \sum_{s \in \mathrm{ne}(x)} \ln(\mu_{f_s \to x}(x)) \boxed{\equiv} \sum_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)}$$

Inference in Graph Models

Max-sum algorithm (maximum a posterior)

$$\mu_{f_s \to x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

$$\mu_{f \to x}(x) = \max_{x_1, \ldots, x_M} \left[\ln f(x, x_1, \ldots, x_M) + \sum_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f}(x_m) \right]$$

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

$$\mu_{x \to f}(x) = \sum_{l \in \mathrm{ne}(x) \setminus f} \mu_{f_l \to x}(x)$$

Inference in Graph Models

Min-sum algorithm (equivalent to maximum a posterior)

Recall the relation between joint distribution and energy function

$$P(\mathrm{x}) = \frac{1}{Z}\exp(-E(\mathrm{x})), \; Z = \sum_x \exp(-E(\mathrm{x}))$$

maximize: $\ln P(\mathrm{x}) = -E(\mathrm{x}) + \boxed{\ln(Z)}$ constant

minimize: $E(\mathrm{x})$

Inference in Graph Models

Min-sum algorithm (equivalent to maximum a posterior)

sum-product: $\boxed{P(\mathbf{x}) = \sum_{\mathbf{x}} \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)}$

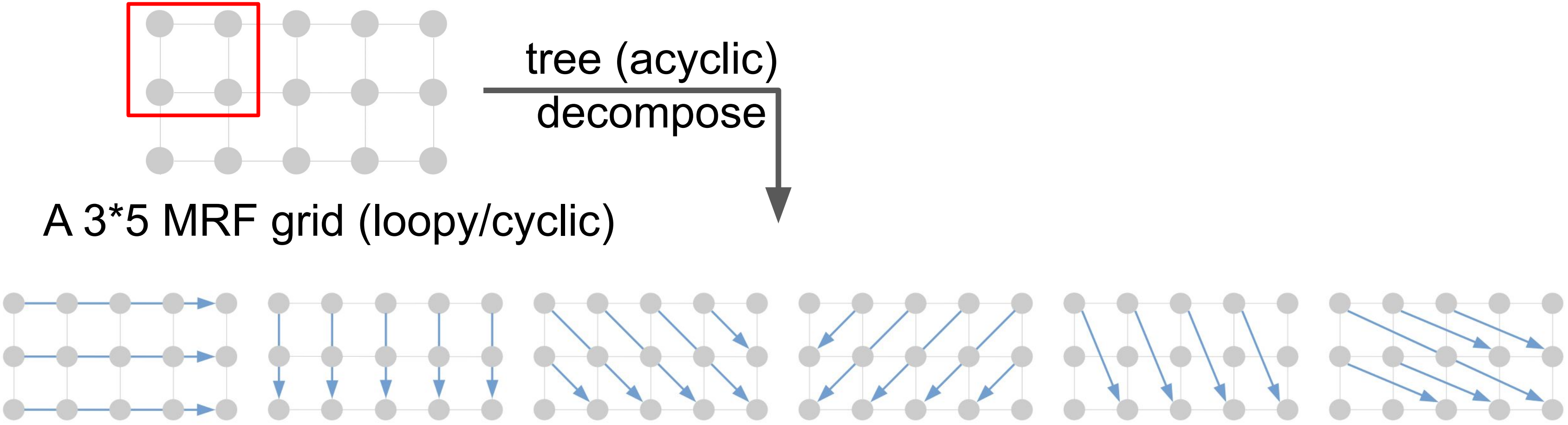
rename variables

$$\Longrightarrow \quad \boxed{E(\mathbf{x}) = \min_{\mathbf{x}} \sum_{s \in \mathrm{ne}(x)} \ln\left(\mu_{f_s \to x}(x)\right) \boxed{\equiv} \min_{\mathbf{x}} \sum_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)}$$

Now, $\mu_{f_s \to x}(x)$ updates as max-sum but contains all min() operations instead of max().

Inference in Graph Models

- Loopy belief propagation
- Mean-field method
- Semi-global matching method
- Tree reweighted message passing method

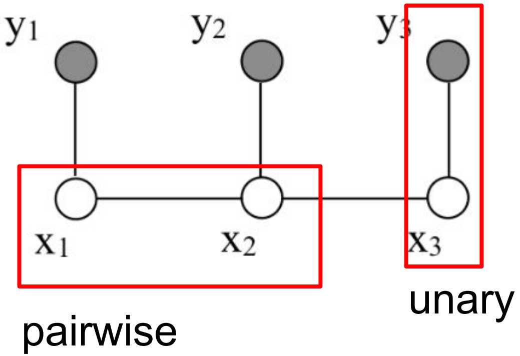


(a) (b) (c) (d) (e) (f)

Inference in Graph Models

Belief propagation

Recall:



Joint probability (Y is unobserved)

$$P(x_1, x_2, x_3, y_1, y_2, y_3) = \phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\psi_1(y_1, x_1)\psi_2(y_2, x_2)\psi_3(y_3, x_3)$$

Conditional probability (Y is observed)

$$P(x_1, x_2, x_3 | y_1, y_2, y_3) = \frac{1}{P(\vec{y})}\phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\psi_1(y_1, x_1)\psi_2(y_2, x_2)\psi_3(y_3, x_3)$$

Marginal probability (Y is observed)

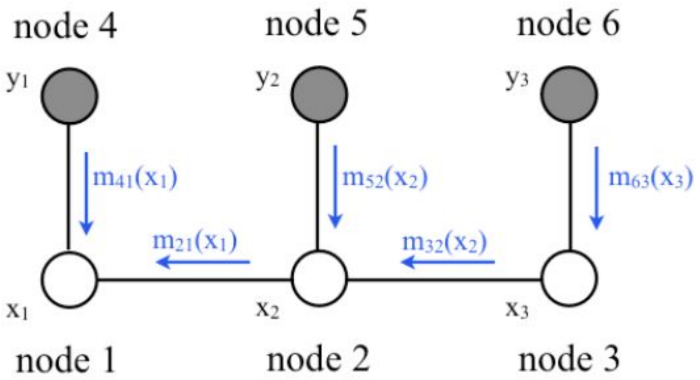
$$P(x_1 | \vec{y}) = \frac{1}{P(\vec{y})}\sum_{x_2}\sum_{x_3}\phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\psi_1(y_1, x_1)\psi_2(y_2, x_2)\psi_3(y_3, x_3)$$

Inference in Graph Models

Belief propagation

Now, message flowing between nodes:

1. Directed
2. Sequential
3. Normalized for probability



$$m_{32}(x_2) = \sum_{x_3} \psi_{23}(x_2, x_3) m_{63}(x_3)$$

$$m_{21}(x_1) = \sum_{x_2} \psi_{12}(x_1, x_2) m_{52}(x_2) m_{32}(x_2)$$

$$P(x_1|\vec{y}) = \frac{1}{P(\vec{y})} m_{41}(x_1) m_{21}(x_1)$$

generalized $\Longrightarrow$

$$m_{ji}(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in \eta(j) \backslash i} m_{kj}(x_j)$$

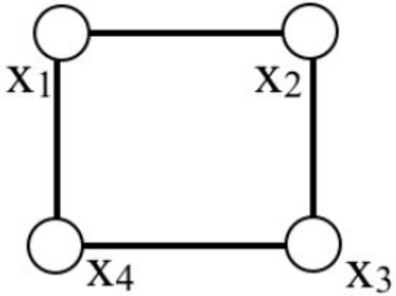
$$P_i(x_i) = \prod_{j \in \eta(i)} m_{ji}(x_i)$$

Inference in Graph Models

Loopy belief propagation

Procedure

1. Initiate all messages as 1
2. Update messages by approximate marginal probability
3. Repeat the above until messages unchanged



Note:

1. Local optimization with fixed points as minima of Bethe free energy, in contrast to the exact solution when on tree or chain

$$P_1(x_1) \quad = \quad \sum_{x_2}\sum_{x_3}\sum_{x_4} \phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\phi_{34}(x_3, x_4)\phi_{41}(x_4, x_1)$$

Inference in Graph Models

Mean-field method, probability perspective (an example from dense CRF)

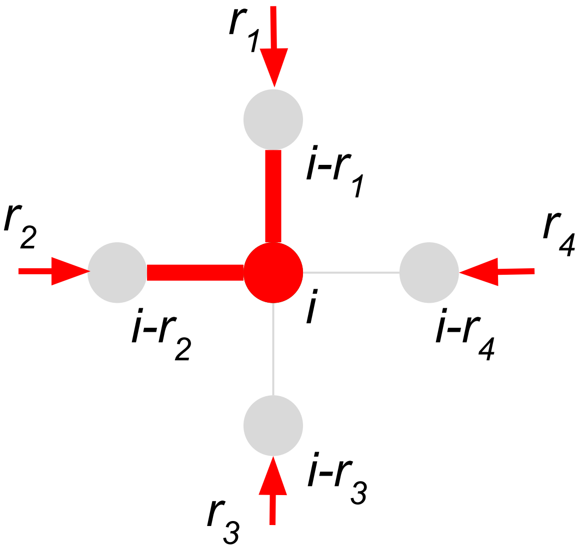
$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\left\{ -\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l') \right\}$$

Initialize Q ▷ $Q_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$
while not converged **do** ▷ See Section 6 for convergence analysis
  $\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m ▷ **Message passing** from all X_j to all X_i
  $\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$ ▷ **Compatibility transform**
  $Q_i(x_i) \leftarrow \exp\{-\psi_u(x_i) - \hat{Q}_i(x_i)\}$ ▷ **Local update**
  normalize $Q_i(x_i)$
end while

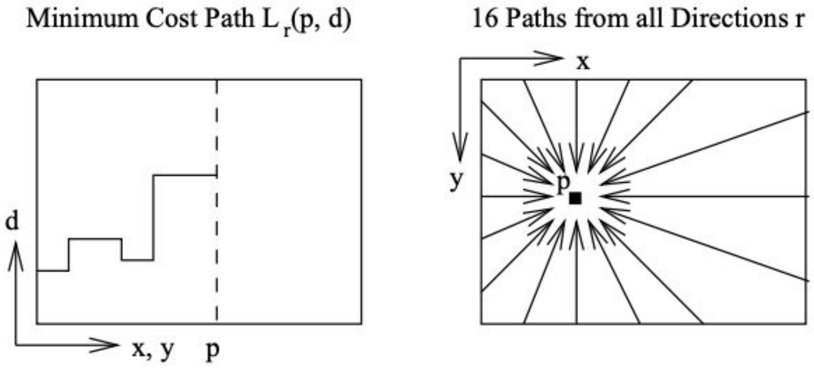
more details: *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*, NIPS 2011

Inference in Graph Models

Semi-global matching method, energy perspective



$m^{r\_1}+m^{r\_2}+m^{r\_3}+m^{r\_4}=c$

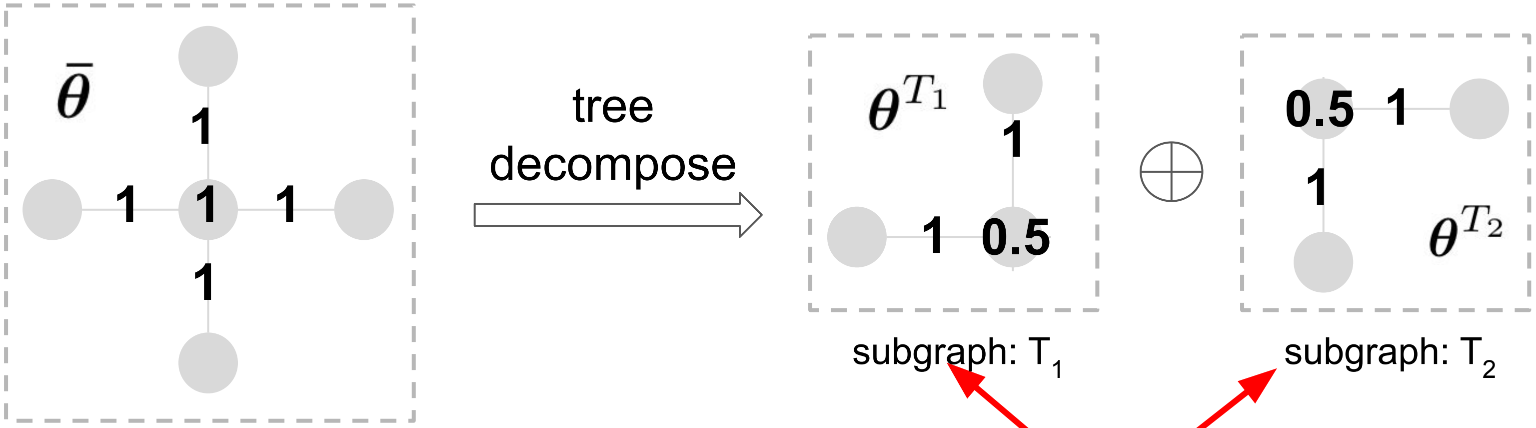


$$
\begin{aligned}
L_{\mathbf{r}}(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d), \\
& L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d-1) + P_1, \\
& L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d+1) + P_1, \\
& \min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, k)
\end{aligned}
$$

more details: *Stereo Processing by Semi-Global Matching and Mutual Information*, TPAMI 2007

Inference in Graph Models

Tree reweighted message passing method, energy perspective



$$\max_{\boxed{\boldsymbol{\theta}}\in\mathcal{A},\sum_T \rho^T\theta^T=\bar{\theta}} \boxed{\sum_T \rho^T \boxed{\min_{\mathbf{x}\in\mathcal{X}} \langle\theta^T, \phi(\mathbf{x})\rangle}}$$

Energy minimization on each subgraph

Reparameterization on a larger set

A lower bound on the condition of $\sum_T \rho^T\theta^T = \bar{\theta}$

more details: *Convergent Tree-reweighted Message Passing for Energy Minimization*, PAMI, 2006

Example

Image rendering using LBP

- Refer to paper “Loopy Belief Propagation in Image-Based Rendering”
- Additional with higher-order prior, see “Efficient Belief Propagation for Vision Using Linear Constraint Nodes”