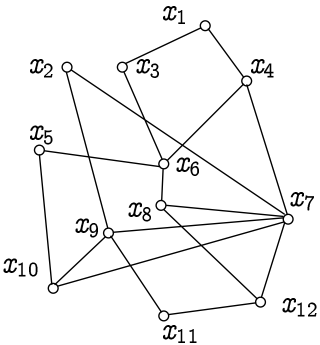
Stat 375: Inference in Graphical Models
Lectures 5-6

Andrea Montanari

Stanford University

April 16, 2012

Undirected Pairwise Graphical Model

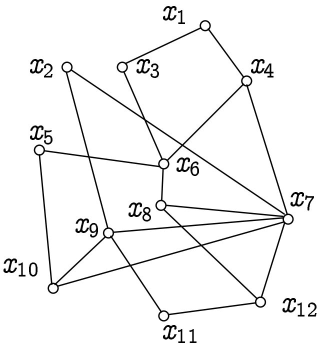


$G = (V, E)$, $V = [n]$, $x = (x_1, \dots, x_n)$, $x_i \in \mathcal{X}$, $|\mathcal{X}| < \infty$

$$\mu(x) = \frac{1}{Z} \prod_{(ij) \in E} \psi_{ij}(x_i, x_j) \, .$$

Computing marginals of μ

Undirected Pairwise Graphical Model



$G = (V, E)$, $V = [n]$, $x = (x_1, \dots, x_n)$, $x_i \in \mathcal{X}$, $|\mathcal{X}| < \infty$

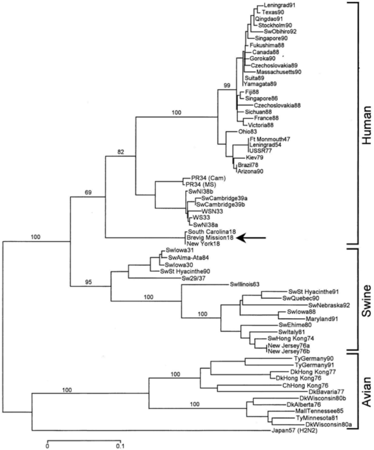
$$\mu(x) = \frac{1}{Z} \prod_{(ij) \in E} \psi_{ij}(x_i, x_j)\,.$$

Computing marginals of μ

Outline

A motivating example

Influenza



[A.H. Reid, T.G. Fanning, J.V. Hultin, and J.K. Taubenberger, Proc. Natl. Acad. Sci. 96 (1999) 1651-1656]

Challenges in phylogeny

Phylogeny reconstruction: Given DNA sequences at vertices (only at leaves), infer the underlying tree $T = (V, E)$.

Phylogeny evaluation: Given a tree $T = (V, E)$ evaluate the probability of observed DNA sequences at vertices (only at leaves).

Challenges in phylogeny

Phylogeny reconstruction: Given DNA sequences at vertices (only at leaves), infer the underlying tree $T = (V, E)$.

Phylogeny evaluation: Given a tree $T = (V, E)$ evaluate the probability of observed DNA sequences at vertices (only at leaves).

Challenges in phylogeny

Phylogeny reconstruction: Given DNA sequences at vertices (only at leaves), infer the underlying tree $T = (V, E)$.

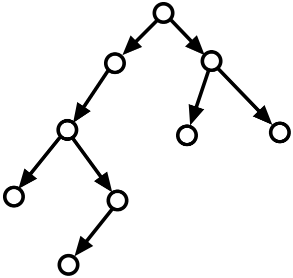
Phylogeny evaluation: Given a tree $T = (V, E)$ evaluate the probability of observed DNA sequences at vertices (only at leaves).

Challenges in phylogeny

Phylogeny reconstruction: Given DNA sequences at vertices (only at leaves), infer the underlying tree $T = (V, E)$.

Phylogeny evaluation: Given a tree $T = (V, E)$ evaluate the probability of observed DNA sequences at vertices (only at leaves).

A Markov model

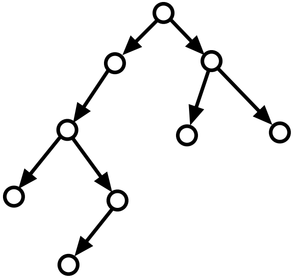


$T = (V, D)$ directed graph, observed $x = (x_i)_{i \in V} \in \mathcal{X}^V$

$$\mu_T(x) = q_o(x_o) \prod_{(i,j) \in D} q_{i,j}(x_i, x_j)\,,$$

$$q_{i,j}(x_i, x_j) = \text{Probability that the descendent is } x_j \text{ if ancestor is } x_i.$$

A Markov model

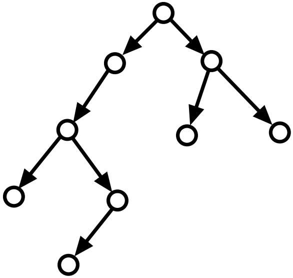


$T = (V, D)$ directed graph, observed $x = (x_i)_{i \in V} \in \mathcal{X}^V$

$$\mu_T(x) = q_o(x_o) \prod_{(i,j) \in D} q_{i,j}(x_i, x_j)\,,$$

$$q_{i,j}(x_i, x_j) = \text{Probability that the descendent is } x_j \text{ if ancestor is } x_i.$$

A Markov model

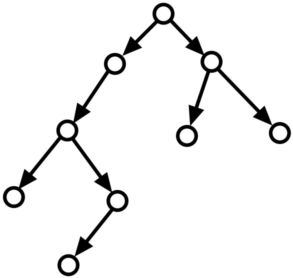


$T = (V, D)$ directed graph, observed $x = (x_i)_{i \in V} \in \mathcal{X}^V$

$$\mu_T(x) = q_o(x_o) \prod_{(i,j) \in D} q_{i,j}(x_i, x_j)\,,$$

$$q_{i,j}(x_i, x_j) = \text{Probability that the descendent is } x_j \text{ if ancestor is } x_i.$$

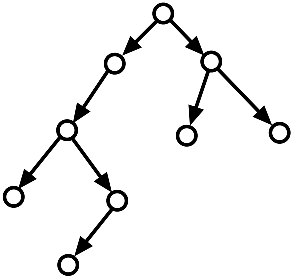
Simplified model: $\mathcal{X} = \{+1, -1\}$



$$\begin{aligned} q_o(x_o) &= \frac{1}{2}, \\ q(x_i, x_j) &= \begin{cases} 1-q & \text{if } x_j = x_i, \\ q & \text{if } x_i \neq x_j. \end{cases} \end{aligned}$$

$$q(x_i, x_j) \propto e^{\theta x_i x_j}.$$

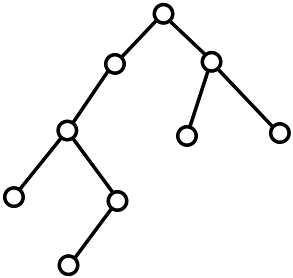
Simplified model: $\mathcal{X} = \{+1, -1\}$



$$q_o(x_o) = \frac{1}{2},$$

$$q(x_i, x_j) = \begin{cases} 1 - q & \text{if } x_j = x_i, \\ q & \text{if } x_i \neq x_j. \end{cases} \qquad q(x_i, x_j) \propto e^{\theta x_i x_j}.$$

Simplified model: $\mathcal{X} = \{+1, -1\}$

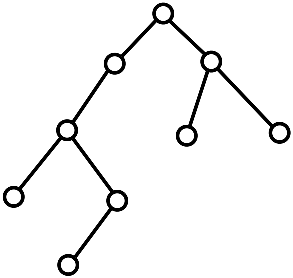


$T = (V, E)$ directed graph, observed $x = (x_i)_{i \in V} \in \mathcal{X}^V$

$$\mu_T(x) = \frac{1}{Z_\theta(T)} \prod_{(i,j) \in E} e^{\theta x_i x_j} \, .$$

Problem: For given T, compute $\mu_T(x)$. Difficult part: $Z_\theta(T)$.

Simplified model: $\mathcal{X} = \{+1, -1\}$



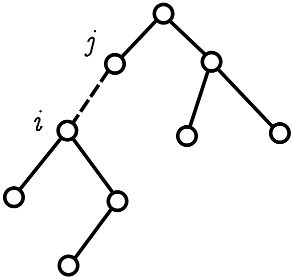
$T = (V, E)$ directed graph, observed $x = (x_i)_{i \in V} \in \mathcal{X}^V$

$$\mu_T(x) = \frac{1}{Z_\theta(T)} \prod_{(i,j) \in E} e^{\theta x_i x_j} \, .$$

Problem: For given T, compute $\mu_T(x)$. Difficult part: $Z_\theta(T)$.

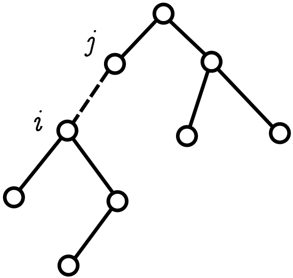
Recursion on trees

Subtree



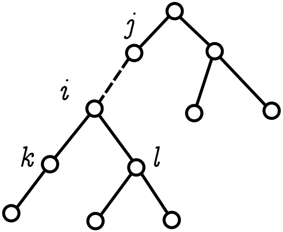
$$
\begin{aligned}
T_{i\to j} = (V_{i\to j}, E_{i\to j}) \quad &= \quad \text{Subtree rooted at } i \text{ and excluding } j, \\
\mu_{i\to j}(x_{V_{i\to j}}) \quad &= \quad \frac{1}{Z(T_{i\to j})} \prod_{(u,v)\in E_{i\to j}} e^{\theta x_u x_v}, \\
\nu_{i\to j}(x_i) \quad &= \quad \sum_{x_{V_{i\to j}}\setminus i} \mu_{i\to j}(x_{V_{i\to j}})
\end{aligned}
$$

Subtree



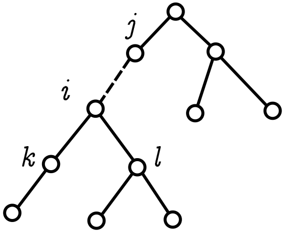
$$
\begin{aligned}
T_{i\to j} = (V_{i\to j}, E_{i\to j}) &= \text{Subtree rooted at } i \text{ and excluding } j,\\
\mu_{i\to j}(x_{V_{i\to j}}) &= \frac{1}{Z(T_{i\to j})} \prod_{(u,v)\in E_{i\to j}} e^{\theta x_u x_v},\\
\nu_{i\to j}(x_i) &= \sum_{x_{V_{i\to j}}\setminus i} \mu_{i\to j}(x_{V_{i\to j}})
\end{aligned}
$$

Merging trees



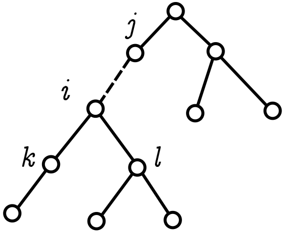
$$\mu_{i\to j}(x_{V_{i\to j}}) = \frac{1}{Z(T_{i\to j})}\, e^{\theta x_i x_k}\; e^{\theta x_i x_l} \Big\{ \prod_{(u,v)\in E_{k\to i}} e^{\theta x_i x_j} \Big\} \Big\{ \prod_{(u,v)\in E_{l\to i}} e^{\theta x_u x_v} \Big\}$$

Merging trees



$$\mu_{i\to j}(x_{V_{i\to j}}) \cong e^{\theta x_i x_k}\, e^{\theta x_i x_l} \Big\{ \prod_{(u,v)\in E_{k\to i}} e^{\theta x_i x_j} \Big\} \Big\{ \prod_{(u,v)\in E_{l\to i}} e^{\theta x_u x_v} \Big\}$$

Merging trees

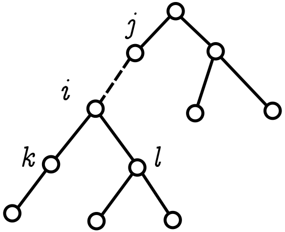


$$\mu_{i\to j}(x_{V_{i\to j}}) \cong e^{\theta x_i x_k}\, e^{\theta x_i x_l} \mu_{k\to i}(x_{V_{k\to i}}) \mu_{l\to i}(x_{V_{l\to i}})$$

$$\sum_{x_{v_{i\to j}\setminus i}} \mu_{i\to j}(x_{V_{i\to j}}) \cong \sum_{x_{V_{k\to i}}} e^{\theta x_i x_k} \mu_{k\to i}(x_{V_{k\to i}}) \sum_{x_{V_{l\to i}}} e^{\theta x_i x_l} \mu_{l\to i}(x_{V_{l\to i}}),$$

$$\nu_{i\to j}(x_i) \cong \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\} \Big\{ \sum_{x_l} e^{\theta x_i x_l} \nu_{l\to i}(x_l) \Big\}$$

Merging trees

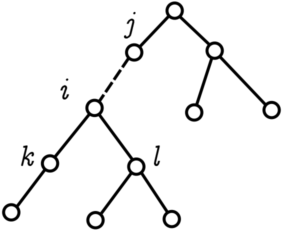


$$\mu_{i\to j}(x_{V_{i\to j}}) \cong e^{\theta x_i x_k}\, e^{\theta x_i x_l} \mu_{k\to i}(x_{V_{k\to i}})\mu_{l\to i}(x_{V_{l\to i}})$$

$$\sum_{x_{v_{i\to j}\setminus i}} \mu_{i\to j}(x_{V_{i\to j}}) \cong \sum_{x_{V_{k\to i}}} e^{\theta x_i x_k} \mu_{k\to i}(x_{V_{k\to i}}) \sum_{x_{V_{l\to i}}} e^{\theta x_i x_l} \mu_{l\to i}(x_{V_{l\to i}}),$$

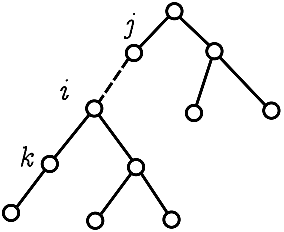
$$\nu_{i\to j}(x_i) \cong \Big\{\sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k)\Big\}\Big\{\sum_{x_l} e^{\theta x_i x_l} \nu_{l\to i}(x_l)\Big\}$$

Merging trees



$$
\begin{aligned}
\mu_{i\to j}(x_{V_{i\to j}}) &\cong e^{\theta x_i x_k}\, e^{\theta x_i x_l} \mu_{k\to i}(x_{V_{k\to i}})\mu_{l\to i}(x_{V_{l\to i}}) \\
\sum_{x_{v_{i\to j}\setminus i}} \mu_{i\to j}(x_{V_{i\to j}}) &\cong \sum_{x_{V_{k\to i}}} e^{\theta x_i x_k} \mu_{k\to i}(x_{V_{k\to i}}) \sum_{x_{V_{l\to i}}} e^{\theta x_i x_l} \mu_{l\to i}(x_{V_{l\to i}}), \\
\nu_{i\to j}(x_i) &\cong \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\} \Big\{ \sum_{x_l} e^{\theta x_i x_l} \nu_{l\to i}(x_l) \Big\}
\end{aligned}
$$

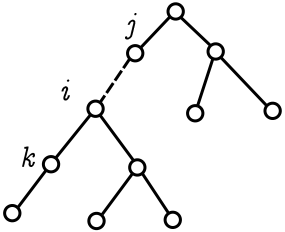
Merging trees



$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\}$$

$$\nu_i(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\} = \mu_T(x_i)$$

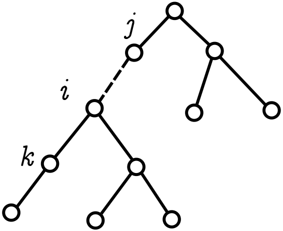
Merging trees



$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\}$$

$$\nu_i(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k} e^{\theta x_i x_k} \nu_{k\to i}(x_k) \Big\} = \mu_T(x_i)$$

What about a general model ?

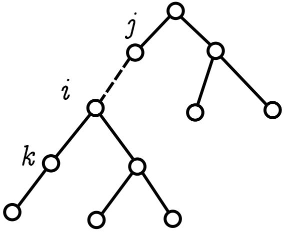


$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j}\Big\{\sum_{x_k\in\mathcal{X}}\psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\}$$

$$\nu_i(x_i) \cong \prod_{k\in\partial i}\Big\{\sum_{x_k\in\mathcal{X}}\psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\} = \mu_T(x_i)$$

Computes the marginals in $|\mathcal{X}|^2 n$ operations.

What about a general model ?



$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

$$\nu_i(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\} = \mu_T(x_i)$$

Computes the marginals in $|\mathcal{X}|^2 n$ operations.

Belief propagation

Notation

$$\vec{E} \equiv \text{Directed edges},$$

$$\nu \equiv \{\nu_{i\to j}(\,\cdot\,)\}_{(i,j)\in\vec{E}}\} \in \mathsf{M}(\mathcal{X})^{\vec{E}}, \text{ messages}$$

$$\mathsf{F} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{\vec{E}}$$

$$\nu \mapsto \mathsf{F}(\nu), \quad \mathsf{F}(\nu)_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i, x_k)\nu_{k\to i}(x_k) \Big\}$$

$$\mathsf{F}^{\mathsf{v}} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{V}$$

$$\nu \mapsto \mathsf{F}^{\mathsf{v}}(\nu), \quad \mathsf{F}^{\mathsf{v}}(\nu)_{i}(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i, x_k)\nu_{k\to i}(x_k) \Big\}$$

Notation

$$\vec{E} \equiv \text{Directed edges,}$$

$$\nu \equiv \{\nu_{i\to j}(\,\cdot\,)\}_{(i,j)\in\vec{E}}\} \in \mathsf{M}(\mathcal{X})^{\vec{E}}, \text{ messages}$$

$$\mathsf{F} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{\vec{E}}$$

$$\nu \mapsto \mathsf{F}(\nu), \quad \mathsf{F}(\nu)_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

$$\mathsf{F}^{\mathsf{v}} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{V}$$

$$\nu \mapsto \mathsf{F}^{\mathsf{v}}(\nu), \quad \mathsf{F}^{\mathsf{v}}(\nu)_i(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

Notation

$$\vec{E} \equiv \text{Directed edges,}$$

$$\nu \equiv \{\nu_{i\to j}(\,\cdot\,)\}_{(i,j)\in\vec{E}}\} \in \mathsf{M}(\mathcal{X})^{\vec{E}}, \text{ messages}$$

$$\mathsf{F} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{\vec{E}}$$

$$\nu \mapsto \mathsf{F}(\nu), \quad \mathsf{F}(\nu)_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\}$$

$$\mathsf{F}^{\mathsf{V}} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{V}$$

$$\nu \mapsto \mathsf{F}^{\mathsf{V}}(\nu), \quad \mathsf{F}^{\mathsf{V}}(\nu)_{i}(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\}$$

Notation

$$\vec{E} \equiv \text{Directed edges,}$$

$$\nu \equiv \{\nu_{i\to j}(\,\cdot\,)\}_{(i,j)\in\vec{E}}\} \in \mathsf{M}(\mathcal{X})^{\vec{E}}, \text{ messages}$$

$$\mathsf{F} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{\vec{E}}$$

$$\nu \mapsto \mathsf{F}(\nu), \quad \mathsf{F}(\nu)_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\}$$

$$\mathsf{F}^{\mathsf{v}} \quad \mathsf{M}(\mathcal{X})^{\vec{E}} \to \mathsf{M}(\mathcal{X})^{V}$$

$$\nu \mapsto \mathsf{F}^{\mathsf{v}}(\nu), \quad \mathsf{F}^{\mathsf{v}}(\nu)_{i}(x_i) \cong \prod_{k\in\partial i} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\Big\}$$

Belief propagation (sequential version)

Belief propagation(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu_{i\to k}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: Recursively over $(i,j) \in \vec{E}$ compute (from leaves):
3: $\quad \nu_{i\to j} = \mathsf{F}(\nu)_{i\to j}$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad \nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Belief propagation (parallel version)

Messages: $\nu_{i\to j}^{(t)}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu_{i\to k}^{(0)}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max} \equiv \text{diam}(T)\}$
3: $\quad \nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad \nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes all the marginals in $|\mathcal{X}|^2 n \cdot \text{diam}(T)$ operations.

Belief propagation (parallel version)

Messages: $\nu_{i\to j}^{(t)}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}$)
1: Initialize $\nu_{i\to k}^{(0)}(x_i) = 1/\lvert\mathcal{X}\rvert$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max} \equiv \mathrm{diam}(T)\}$
3: $\nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes all the marginals in $\lvert\mathcal{X}\rvert^2 n \cdot \mathrm{diam}(T)$ operations.

Belief propagation (parallel version)

Messages: $\nu^{(t)}_{i\to j}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu^{(0)}_{i\to k}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max} \equiv \mathrm{diam}(T)\}$
3: $\quad\nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad\nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes all the marginals in $|\mathcal{X}|^2 n \cdot \mathrm{diam}(T)$ operations.

Belief propagation (parallel version)

Messages: $\nu^{(t)}_{i\to j}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

Belief propagation(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu^{(0)}_{i\to k}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max} \equiv \text{diam}(T)\}$
3: $\quad\nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad\nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes all the marginals in $|\mathcal{X}|^2 n \cdot \text{diam}(T)$ operations.

Belief propagation (loopy version)

Messages: $\nu^{(t)}_{i\to j}(\,\cdot\,)$, $t \in \{0,1,2,\dots\}$

BELIEF PROPAGATION(Graph $T=(V,E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu^{(0)}_{i\to k}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0,1,2,\dots,t_{\max}\}$
3: $\quad \nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad \nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes ?????? in $|\mathcal{X}|^2 n \cdot t_{\max}$ operations.

Belief propagation (loopy version)

Messages: $\nu^{(t)}_{i\to j}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Graph $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu^{(0)}_{i\to k}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max}\}$
3: $\quad \nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad \nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes ?????? in $|\mathcal{X}|^2 n \cdot t_{\max}$ operations.

Belief propagation (loopy version)

Messages: $\nu_{i\to j}^{(t)}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Graph $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu_{i\to k}^{(0)}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\max}\}$
3: $\quad\nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad\nu_i = \mathsf{F}^{\mathsf{v}}(\nu)_i$;

Computes ?????? in $|\mathcal{X}|^2 n \cdot t_{\max}$ operations.

Belief propagation (loopy version)

Messages: $\nu_{i\to j}^{(t)}(\,\cdot\,)$, $t \in \{0, 1, 2, \dots\}$

BELIEF PROPAGATION(Graph $T = (V, E)$, $\psi = \{\psi_{ij}\}$)

1: Initialize $\nu_{i\to k}^{(0)}(x_i) = 1/|\mathcal{X}|$ for all i leaves;
2: For $t \in \{0, 1, 2, \dots, t_{\rm max}\}$
3: $\quad \nu^{(t+1)} = \mathsf{F}(\nu^{(t)})$;
4: For each $i \in V$ output the estimated marginal;
5: $\quad \nu_i = \mathsf{F}^{\rm v}(\nu)_i$;

Computes ?????? in $|\mathcal{X}|^2 n \cdot t_{\rm max}$ operations.

Folklore about Loopy BP

Generally it does not converge, and if it does, the output is incorrect.
This does not stop people from using it!

- Works better when $\psi_{i,j}(x_i, x_j) = \psi_{ij,1}(x_i)\psi_{ij,2}(x_j) + \text{small}(x_i, x_j)$.
- Works better when G has few short loops.
- Works better when $\psi_{ij}(x_i, x_j)$ is attractive.
- Nonconvex variational principle.

Folklore about Loopy BP

Generally it does not converge, and if it does, the output is incorrect.
This does not stop people from using it!

- Works better when $\psi_{i,j}(x_i, x_j) = \psi_{ij,1}(x_i)\psi_{ij,2}(x_j) + \text{small}(x_i, x_j)$.
- Works better when G has few short loops.
- Works better when $\psi_{ij}(x_i, x_j)$ is attractive.
- Nonconvex variational principle.

Folklore about Loopy BP

Generally it does not converge, and if it does, the output is incorrect.
This does not stop people from using it!

- Works better when $\psi_{i,j}(x_i, x_j) = \psi_{ij,1}(x_i)\psi_{ij,2}(x_j) + \text{small}(x_i, x_j)$.
- Works better when G has few short loops.
- Works better when $\psi_{ij}(x_i, x_j)$ is attractive.
- Nonconvex variational principle.

Folklore about Loopy BP

Generally it does not converge, and if it does, the output is incorrect.
This does not stop people from using it!

- Works better when $\psi_{i,j}(x_i, x_j) = \psi_{ij,1}(x_i)\psi_{ij,2}(x_j) + \text{small}(x_i, x_j)$.
- Works better when G has few short loops.
- Works better when $\psi_{ij}(x_i, x_j)$ is attractive.
- Nonconvex variational principle.

Folklore about Loopy BP

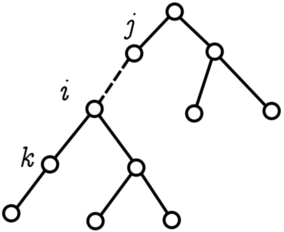
Generally it does not converge, and if it does, the output is incorrect.
This does not stop people from using it!

- Works better when $\psi_{i,j}(x_i, x_j) = \psi_{ij,1}(x_i)\psi_{ij,2}(x_j) + \text{small}(x_i, x_j)$.
- Works better when G has few short loops.
- Works better when $\psi_{ij}(x_i, x_j)$ is attractive.
- Nonconvex variational principle.

A couple of exercises

Exercise #1: Partition function on trees

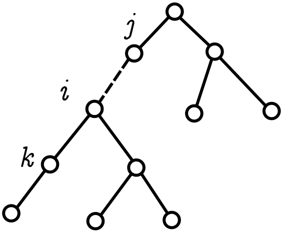
Exercise #1: Partition function on trees



$$Z(T_{i\to j}) = \prod_{k\in\partial i\setminus j} Z(T_{k\to i}) \sum_{x_i\in\mathcal{X}} \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

Computes the partition function in $|\mathcal{X}|^2 n$ operations.

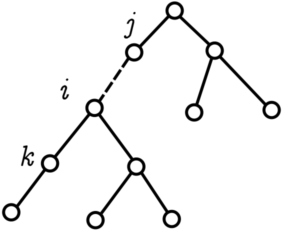
Exercise #1: Partition function on trees



$$Z(T_{i\to j}) = \prod_{k\in\partial i\setminus j} Z(T_{k\to i}) \sum_{x_i\in\mathcal{X}} \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

Computes the partition function in $|\mathcal{X}|^2 n$ operations.

Exercise #1: Partition function on trees



$$Z(T_{i\to j}) = \prod_{k\in\partial i\setminus j} Z(T_{k\to i}) \sum_{x_i\in\mathcal{X}} \prod_{k\in\partial i\setminus j} \Big\{ \sum_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \Big\}$$

Computes the partition function in $|\mathcal{X}|^2 n$ operations.

Exercise #1: Example

Suppose you observe

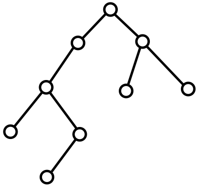
$$x = (+1, +1, +1, +1, +1, +1, +1, +1, +1)$$

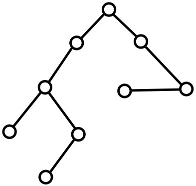
Exercise #1: Example

Suppose you observe

$$x = (+1, +1, +1, +1, +1, +1, +1, +1, +1)$$

and you know this comes from either of





$$\mu(x) = \frac{1}{Z(T)} \prod_{(i,j)\in E} e^{\theta x_i x_j} \prod_{i \in V} e^{\gamma x_i}$$

Which one has highest likelihood?

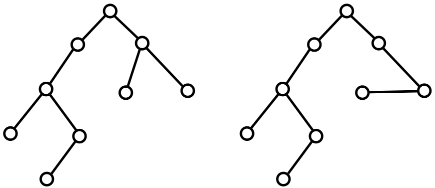
$\arg\min\{Z(T_1), Z(T_2)\}$

Exercise #1: Example

Suppose you observe

$$x = (+1, +1, +1, +1, +1, +1, +1, +1, +1)$$

and you know this comes from either of

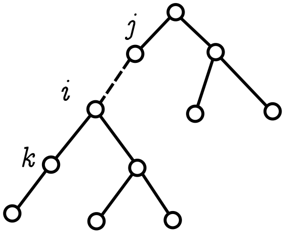


$$\mu(x) = \frac{1}{Z(T)} \prod_{(i,j)\in E} e^{\theta\, x_i x_j} \prod_{i \in V} e^{\gamma\, x_i}$$

Which one has highest likelihood? $\arg\min\{Z(T_1), Z(T_2)\}$

Exercise #2: Sampling from on the tree

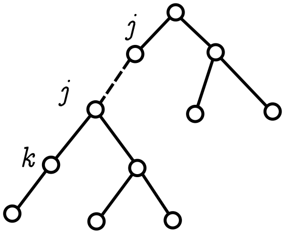
Exercise #2: Sampling on the tree



SAMPLING(Tree $T = (V, E)$, $\psi = \{\psi_{ij}\}_{(ij)\in E}$)

1: Choose a root $o \in V$;
2: Sample $X_o \sim \mu_o(\,\cdot\,)$;
2: Recursively over $i \in V$ (from root to leaves):
3: Compute $\mu_{i|\pi(i)}(x_i|x_{\pi(i)})$;
4: Sample $X_i \sim \mu_{i|\pi(i)}(\,\cdot\,|x_{\pi(i)})$;

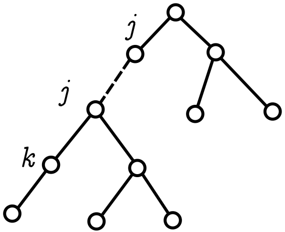
Exercise #2: Sampling on the tree



$$\mu_T(x_{V_{i\to j}}|x_j) \;\cong\; \psi_{ij}(x_i x_j)\,\mu_{T_{i\to j}}(x_{V_{i\to j}})\,,$$

$$\mu_T(x_i|x_j) \;\cong\; \psi_{ij}(x_i x_j)\,\nu_{i\to j}(x_i)\,.$$

Exercise #2: Sampling on the tree

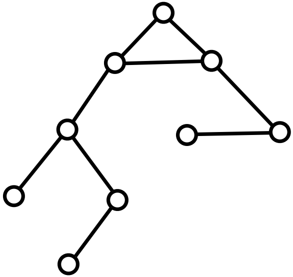


$$\mu_T(x_{V_{i\to j}}|x_j) \cong \psi_{ij}(x_i x_j)\,\mu_{T_{i\to j}}(x_{V_{i\to j}})\,,$$

$$\mu_T(x_i|x_j) \cong \psi_{ij}(x_i x_j)\,\nu_{i\to j}(x_i)\,.$$

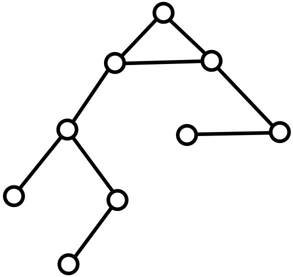
Tree decomposition

Idea



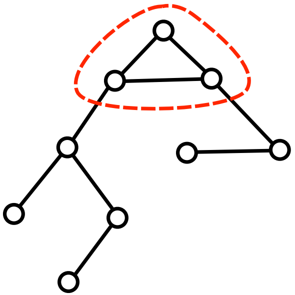
OK, this is not a tree but...

Idea



OK, this is not a tree but. . .

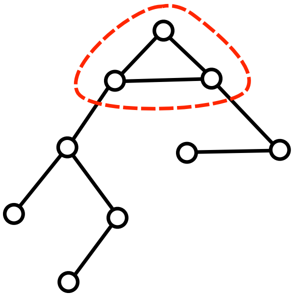
Idea



Create an equivalent tree graph.

Price: enlarging the alphabet to $\mathcal{X}^3$.

Idea



Create an equivalent tree graph.

Price: enlarging the alphabet to $\mathcal{X}^3$.

How you do this in general?

- Tree decomposition.

- Equivalent graphical model.

- Alphabet enlargement $\mathcal{X} \to \mathcal{X}^k$.

- Treewidth$(G) \equiv$ Minimum such k.

Problem: In general Treewidth$(G) = \Theta(n)$.

How you do this in general?

- Tree decomposition.
- Equivalent graphical model.
- Alphabet enlargement $\mathcal{X} \to \mathcal{X}^k$.
- Treewidth$(G) \equiv$ Minimum such k.

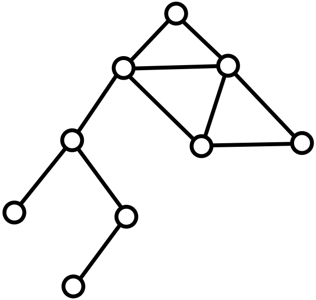
Problem: In general Treewidth$(G) = \Theta(n)$.

Tree decomposition of $G = (V, E)$

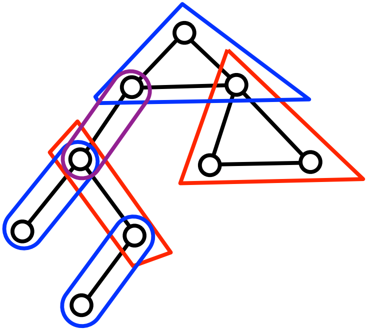
A tree $T = (V_T, E_T)$ and a mapping $V : V_T \to$SUBSETS(V) s.t.:

- For each $i \in V$ there exists at least one $u \in V_T$ with $i \in V(u)$.
- For each $(i, j) \in E$ there exists at least one $u \in V_T$ with $i, j \in V(u)$.
- If $i \in V(u_1)$ and $i \in V(u_2)$, then $i \in V(w)$ for any w on the path between u_1 and u_2 in T.

For instance

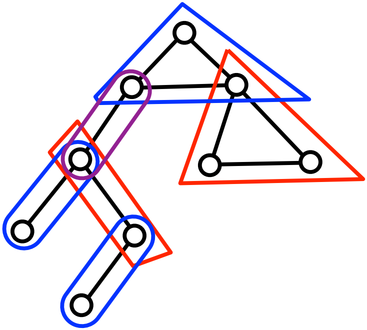


For instance



You see the tree, right?

For instance



You see the tree, right?

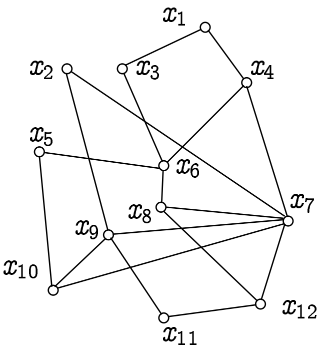
General methods to prove convergence

- Monotonicity.

- Contraction.

The max-product algorithm

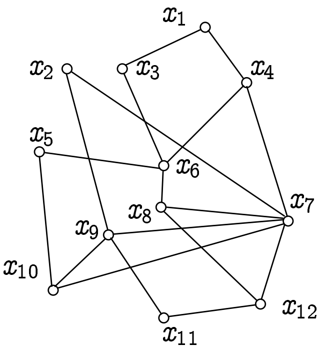
Optimization – Mode computation



$G = (V, E)$, $V = [n]$, $x = (x_1, \dots, x_n)$, $x_i \in \mathcal{X}$, $|\mathcal{X}| < \infty$
Compute

$$\arg\max_{x \in \mathcal{X}^V} \mu(x) = \arg\max_{x \in \mathcal{X}^V} \prod_{(ij) \in E} \psi_{ij}(x_i, x_j) \, .$$

Equivalently



$G = (V, E)$, $V = [n]$, $x = (x_1, \dots, x_n)$, $x_i \in \mathcal{X}$, $|\mathcal{X}| < \infty$
Compute

$$\arg\max_{x \in \mathcal{X}^V} \sum_{(ij) \in E} \theta_{ij}(x_i, x_j)\,.$$

Everything goes through: Max-Marginals

$$\mu_v(x_v) \cong \max\Big\{ \prod_{(ij)\in E} \psi_{ij}(x'_i, x'_j) \;:\; x' \in \mathcal{X}^V, x'_v = x_v \Big\}\,.$$

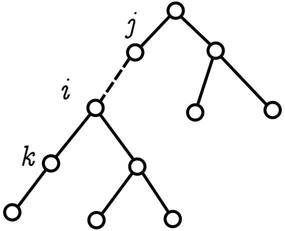
Normalization $\max_{x_v} \mu_v(x_v) = 1$

Everything goes through: Max-Marginals

$$\mu_v(x_v) \cong \max\Big\{ \prod_{(ij)\in E} \psi_{ij}(x'_i, x'_j) \;:\; x' \in \mathcal{X}^V, x'_v = x_v \Big\}\,.$$

Normalization $\max_{x_v} \mu_v(x_v) = 1$

Everything goes through: Max-Product algorithm

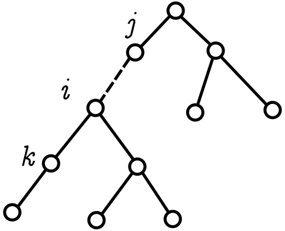


$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \left\{ \max_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \right\}$$

$$\nu_i(x_i) \cong \prod_{k\in\partial i} \left\{ \max_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k) \right\}$$

Computes the max-marginals in $|\mathcal{X}|^2 n$ operations.
$\equiv$ Dynamic programming.

Everything goes through: Max-Product algorithm



$$\nu_{i\to j}(x_i) \cong \prod_{k\in\partial i\setminus j} \left\{ \max_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\right\}$$

$$\nu_{i}(x_i) \cong \prod_{k\in\partial i} \left\{ \max_{x_k\in\mathcal{X}} \psi_{ik}(x_i,x_k)\nu_{k\to i}(x_k)\right\}$$

Computes the max-marginals in $|\mathcal{X}|^2 n$ operations.
$\equiv$ Dynamic programming.