today: sum-product alg.

clique tree:

1

2

3

4

5

running graph.

Eliminate:

+ keep track of all added edges

→ get a triangulated graph

→ graph with no cycle

of size 4 or more that

cannot be broken by a chord

not all ordering are good:

big clique

convention:

treedepth(tree) = 1

treedepth(

min \text{ size of biggest clique} - 1)

= \min \text{ elimination orderng}

treedepth of graph

bad news:

a) NP hard to compute treewidth (or find best ordering)

b) NP hard to do inference in general OGM

(need approximate methods)

e.g., tree width of a grid

\leq \sqrt{\text{dim}}

good news:

\begin{itemize}
  \item inference in linear time for trees (treewidth=1)
  \item (HMM, markov chain)
  \item efficient for "small tree width graph"
  \item graph alg.
\end{itemize}
Inference in a tree:

\[
\mathbb{P}(x_4) = \frac{\mathcal{N}_t(x_4) \sum \mathcal{N}_j(x_j) \mathcal{N}_{x_4} x_3}{\mathcal{Z}}
\]

\[
= \mathcal{N}_{2 \rightarrow 3}(x_3) \mathcal{N}_{3 \rightarrow 4}(x_4)
\]

Make a directed tree by using \( x_4 \) as root.

Sum-product alg (for trees)

get all marginals cheaply by storing & re-using messages
(dynamical programming)

Goal is to have \( \forall i, j \in E \), want to have

\[
M_{i \rightarrow j}(x_j)
\]

\[
M_{j \rightarrow i}(x_i)
\]

Rule: I can only send message to neighbor \( j \) \( M_{i \rightarrow j}(x_j) \)

When it has received all messages from other neighbors

\[
M_{i \rightarrow j}(x_j) = \sum_{x_0} \mathcal{N}_i(x_i) \mathcal{N}_{ij}(x_i, x_j) \bigg/ \bigg( \sum_{x_i} \prod_{k \neq i} \mathcal{N}_k(x_k) \bigg)
\]

Teaching Page 2
condtions: 
\[ \langle z, \epsilon \rangle \rightarrow \rho(z, \epsilon) \]

Loop bp: approximate equality

-\[ \langle z, \epsilon \rangle = \rho(z, \epsilon) \]

\[ m_i(z) = \frac{1}{n} \sum_{\epsilon} \rho(z, \epsilon) \]

2) at every step in parallel: compute \( m_i(z) \)

3) untangle \( m_i(z) \) to uniform distribution \( T(\cdot, \cdot) \)

end: \[ \rho(x, y) = \sum_{z, \epsilon} m_i(z) \rho(z, \epsilon) \]

node's distribute/collect schedule:

- evenly distribute messages across tree

- every step, messages are correctly computed
Keep it fixed during sum-product

(formal rule): redefine 

\[ P_i^j(x_j) = \frac{1}{Z_j} \sum \frac{P^i(x_i) \cdot \text{shift}(x_i, x_j)}{Z_j} \]

Kronecker-delta \( \delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \)

\( m_{j \rightarrow i}(x_i) = \frac{1}{Z_j} \sum \frac{P^i_j(x_j) \cdot \text{shift}(x_j, x_i)}{Z_j} \)

\( \rho(x_i | \overline{\mathbf{x}}_E) = \frac{P(x_i, \overline{\mathbf{x}}_E)}{\sum_{x_i} P(x_i, \overline{\mathbf{x}}_E)} \)

\[ \max \text{-} product? \]

Only properly used: distributivity of + with \( \circ \)

\((\mathbb{R}, +, \circ)\) is \underline{semi-ring} (don't need inverse to addition)

Can do "sum-product" on other semi-rings

\((\mathbb{R}_+, \max, \circ)\)

\( \max (a \cdot b, a \cdot c) \)

\( \circ = a \cdot \max (b, c) \)

\((\mathbb{R}, \max, \circ)\)

"max-product"

\[ \max_{x_i \in \mathbb{X}_i} \prod_{j \in \mathbb{X}_j} \max_T \left( \prod_{i \in \mathbb{X}_i} P^i_j(x_j) \right) \]

\[ m_{i \rightarrow j}(x_j) = \max_{x_i} \left[ \prod_{i \in \mathbb{X}_i} P^i_j(x_i, x_j) \prod_{k \in \mathbb{X}_k \setminus \{i \}} m_{k \rightarrow i}(x_k) \right] \]

\( m_{i \rightarrow j}(x_j) \) for arg-max, store argument as \( \text{argument of } x_j \)

Max-product aka Viterbi algorithm \( \rightarrow \) decoding argmax \( \rho(x_{1:n}) \)

\( \otimes \) implementation comment \( \prod_{i \in \mathbb{X}_i} \max_T (x_i) \)
\[
\mathcal{K} \triangleq \frac{1}{\text{small it.}} \exp \left( \sum_{i} \log m_{x_{i}}(x_{i}) \right)
\]
\[
\sum_{x} f(x) = \sum_{x} \exp \left( \log f(x) \right) = \exp (\log f_{\text{max}}) \left[ 1 + \sum_{x \neq x_{\text{max}}} \exp \left( \log \frac{f(x)}{f_{\text{max}}} \right) \right]
\]

**Proposition:**

\( \rho \in \mathcal{F}(\text{Tree}) \) with non-zero marginals \( \Rightarrow \rho(x) = \prod_{i} \prod_{y \in \text{set}} \rho(x_{i}, y_{i}) \rho(x_{i}) \rho(y_{i}) \)

**Proof:**

\( \rho \in \mathcal{F}(\text{undirected tree}) \iff \rho \in \mathcal{F}(\text{any orientation}) \)

\( \Rightarrow \rho(x) = \prod_{i} \rho(x_{i}, y_{i}) \rho(x_{i}) \) for some orientation of the

say \( x_n \) is root

\[
= \prod_{i} \rho(x_{i}, y_{i}) \rho(x_{i})
\]

\[
= \prod_{i} \rho(x_{i}) \prod_{y \in \text{set}} \rho(x_{i}, y_{i}) \rho(x_{i})
\]

\[
= \prod_{i} \rho(x_{i}) \prod_{y \in \text{set}} \rho(x_{i}, y_{i}) \rho(x_{i})
\]

as before, for any set of factors \( \prod_{j} \rho_{j}(x_{i}, y_{i}) \)

\[
\mathcal{S}_{i} \ni \sum_{x_{i}} \leq \sum_{x_{i}} \rho_{j}(x_{i}, y_{i}) = \rho_{j}(x_{i})
\]

"local consistency property" \( \leq \sum_{x_{i}} \rho_{j}(x_{i}, y_{i}) = \rho_{j}(x_{i}) \)

\( \leq \sum_{x_{i}} \rho_{j}(x_{i}) = 1 \)

Define a joint \( \rho(x) = \prod_{i} \rho(x_{i}) \prod_{y \in \text{set}} \rho(x_{i}, y_{i}) \rho(x_{i}) \)

with correct marginals i.e. \( \rho(x_{i}) = \rho_{i}(x_{i}) \) etc...

junction tree alg:
generalization of sum-product to clique trees.

running interaction property

triangulated graph / decomposable graph

running

Graph Eliminate