Exact Inference

Qinfeng (Javen) Shi

24 April 2017

Table of Contents I

Marginal and MAP Queries

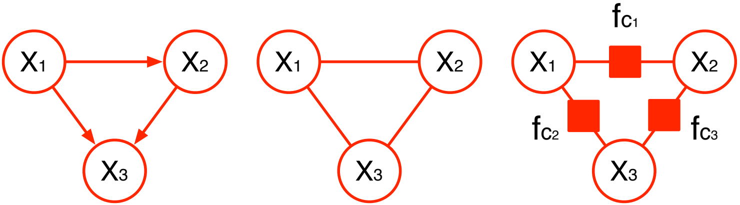
Given joint distribution $P(Y, E)$, where

- Y, query random variable(s), unknown
- E, evidence random variable(s), observed *i.e.* E = e.

Two types of queries:

- Marginal queries (a.k.a. probability queries)
  task is to compute $P(Y|E = e)$
- MAP queries (a.k.a. most probable explanation)
  task is to find $y^* = \mathrm{argmax}_{y \in Val(Y)} P(Y|E = e)$

Marginal and MAP Inference



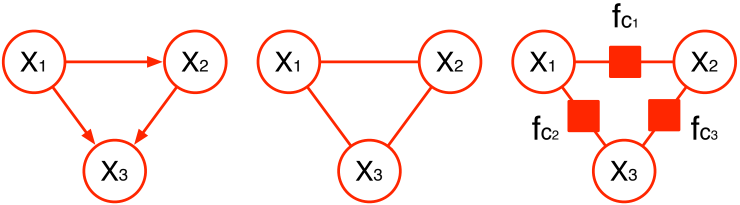
(a) Directed graph (b) Undirected graph (c) Factor graph

Marginal inference: $P(x_i) = \sum_{x_j : j \neq i} P(x_1, x_2, x_3)$

MAP inference: $(x_1^*, x_2^*, x_3^*) = \operatorname{argmax}_{x_1, x_2, x_3} P(x_1, x_2, x_3)$

Warning: $x_i^* \neq \operatorname{argmax}_{x_i} P(x_i)$ in general

Marginal and MAP Inference



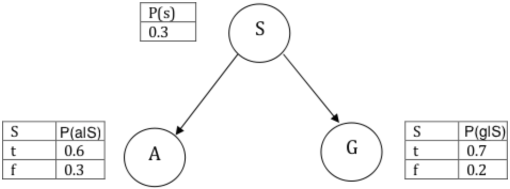
(d) Directed graph (e) Undirected graph (f) Factor graph

Extends to seeing the evidence E,

Marginal inference: $P(x_i|E) = \sum_{x_j : j \neq i} P(x_1, x_2, x_3|E)$

MAP inference: $(x_1^*, x_2^*, x_3^*) = \operatorname{argmax}_{x_1, x_2, x_3} P(x_1, x_2, x_3|E)$

Example of 4WD



- $P(\neg g, a|s)$? (*i.e.* $P(G = \neg g, A = a|S = s)$)
- $P(S)$?
- $\text{argmax}_{G,A,S} P(G, A, S)$?

Marginals

When do we need marginals? Marginals are used to compute

- query for probabilities like in W4D example.
- normalisation constant
  $Z = \sum_{x_i} q(x_i) = \sum_{x_j} q(x_j) \quad \forall i, j = 1, \ldots.$
  log loss in Conditional Random Fields (CRFs) is
  $-\log P(x_1, \ldots, x_n) = \log(Z) + \ldots.$
  Here $q(x_i)$ is a belief (not necessarily a probability) in marginal inference.
- expectations like $\mathbb{E}_{P(x_i)}[\phi(x_i)]$ and $\mathbb{E}_{P(x_i,x_j)}[\phi(x_i, x_j)]$, where
  $\psi(x_i) = \langle \phi(x_i), w \rangle$ and $\psi(x_i, x_j) = \langle \phi(x_i, x_j), w \rangle$
  Gradient of CRFs risk contains above expectations.

MAP

When do we need MAP?

- find the most likely configuration for $(x_i)_{i \in \mathcal{V}}$ in testing.
- find the most violated constraint generated by $(x_i^\dagger)_{i \in \mathcal{V}}$ in training (*i.e.* learning), *e.g.* by cutting plane method (used in SVM-Struct) or by Bundle method for Risk Minimisation (Teo JMLR2010).

How to infer?

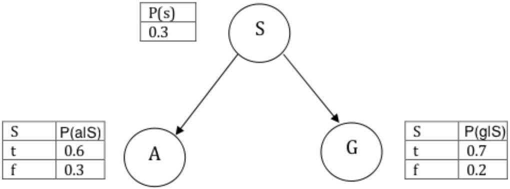
How to infer by hand for Bayesian Networks? (previous lecture).

Problems: hand-tiring for many variables, and it's only for Bayesian Networks.

How to infer for other graphical models and how to do it in a computer program?

Variable elimination

Variable elimination: infer by eliminating variables (works for both marginal and MAP inference)



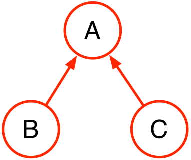
$$
\begin{aligned}
P(A) &= \sum_{S,G} P(A,S,G) \\
&= \sum_{S,G} P(S)P(A|S)P(G|S) \\
&= \sum_{S} P(S)P(A|S)(\sum_{G} P(G|S)) = \sum_{S} P(S)P(A|S)
\end{aligned}
$$

VE for marginal inference

Step by step:

1. sum over missing variables (marginalisation) for the full distribution.
2. factorise the full distribution.
3. rearrange the sum operator to reduce the computation.
4. eliminate the variables.

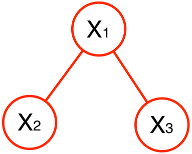
Variable elimination — BayesNets



Marginal inference $P(A)$?

$$
\begin{aligned}
P(A) &= \sum_{B,C} P(A,B,C) \\
&= \sum_{B,C} P(B)P(C)P(A|B,C) \\
&= \sum_{B} P(B) \sum_{C} P(C)P(A|B,C) \\
&= \sum_{B} P(B)m_1(A,B) \quad (C \text{ eliminated}) \\
&= m_2(A) \quad (B \text{ eliminated})
\end{aligned}
$$

Variable elimination — MRFs

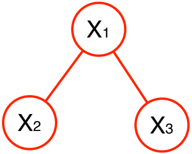


$$P(x_1, x_2, x_3) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3)$$

ψ are given. Show example using the document camera.

$$\begin{aligned}
P(x_1) &= \sum_{x_2, x_3} \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3) \\
&= \frac{1}{Z}\sum_{x_2, x_3} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3) \\
&= \frac{1}{Z}\psi(x_1)\sum_{x_2}\Big(\psi(x_1, x_2)\psi(x_2)\Big)\sum_{x_3}\Big(\psi(x_1, x_3)\psi(x_3)\Big) \\
&= \frac{1}{Z}\psi(x_1)m_{2\to 1}(x_1)m_{3\to 1}(x_1)
\end{aligned}$$

Variable elimination — MRFs



$$
\begin{aligned}
P(x_2) &= \sum_{x_1, x_3} \frac{1}{Z} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3) \\
&= \frac{1}{Z}\psi(x_2) \sum_{x_1} \left(\psi(x_1, x_2)\psi(x_1) \sum_{x_3} \left[\psi(x_1, x_3)\psi(x_3) \right] \right) \\
&= \frac{1}{Z}\psi(x_2) \sum_{x_1} \psi(x_1, x_2)\psi(x_1) m_{3\to 1}(x_1) \\
&= \frac{1}{Z}\psi(x_2) m_{1\to 2}(x_2)
\end{aligned}
$$

Variable elimination — factor graphical models

Works too.

Replace the ψ by factors $f_1, f_2, \ldots$

VE for MAP inference

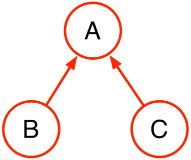
MAP inference:

$$(x_1^*, x_2^*, x_3^*, \dots, x_n^*) = \operatorname*{argmax}_{x_1, x_2, x_3, \dots, x_n} P(x_1, x_2, x_3, \dots, x_n)$$

Step by step:

1. max over the full distribution.
2. factorise the full distribution.
3. rearrange the max operator to reduce the computation.
4. eliminate the variables.

Variable elimination — BayesNets

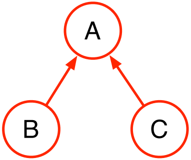


MAP inference $\operatorname{argmax}_{A,B,C} P(A,B,C)$?

$$
\begin{aligned}
\max_{A,B,C} P(A,B,C) &= \max_{A,B,C} P(B)P(C)P(A|B,C) \\
&= \max_A \left\{ \max_B \left[P(B) \max_C \left(P(C)P(A|B,C) \right) \right] \right\} \\
&= \max_A \left\{ \max_B \left[P(B) m_1(A,B) \right] \right\} \ \ (C \text{ eliminated, record its best assignment}) \\
&= \max_A m_2(A) \ \ (B \text{ eliminated, record its best assignment, and A's best assignment})
\end{aligned}
$$

MAP solution?

Variable elimination — BayesNets



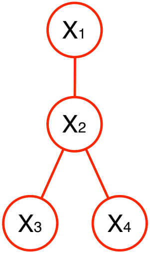
MAP inference $\mathrm{argmax}_{A,B,C}\, P(A,B,C)$?

$$\begin{aligned}
\max_{A,B,C} P(A,B,C) &= \max_{A,B,C} P(B)P(C)P(A|B,C)\\
&= \max_{A}\left\{\max_{B}\left[P(B)\max_{C}\Big(P(C)P(A|B,C)\Big)\right]\right\}\\
&= \max_{A}\left\{\max_{B}\left[P(B)m_1(A,B)\right]\right\}\ \ (C \text{ eliminated, record its best assignment})\\
&= \max_{A} m_2(A)\ \ (B \text{ eliminated, record its best assignment, and A's best assignment})
\end{aligned}$$

MAP solution? $\mathrm{argmax} = A, B, C$'s best assignments.

Variable elimination — MRFs

$$
\begin{aligned}
&\max_{x_1,x_2,x_3,x_4} P(x_1,x_2,x_3,x_4)\\
&= \max_{x_1,x_2,x_3,x_4} \psi(x_1,x_2)\psi(x_2,x_3)\psi(x_2,x_4)\psi(x_1)\psi(x_2)\psi(x_3)\psi(x_4)\\
&= \max_{x_1,x_2}\left[\ldots \max_{x_3}\Big(\psi(x_2,x_3)\psi(x_3)\Big)\max_{x_4}\Big(\psi(x_2,x_4)\psi(x_4)\Big)\right]\\
&= \max_{x_1}\left[\psi(x_1)\max_{x_2}\Big(\psi(x_2)\psi(x_1,x_2)m_{3\to 2}(x_2)m_{4\to 2}(x_2)\Big)\right]\\
&= \max_{x_1}\Big(\psi(x_1)m_{2\to 1}(x_1)\Big)
\end{aligned}
$$



argmax = recorded best assignments.
What if you didn't (or don't want to) record the assignments?
How to get them back?
Answer:
backtrack the best assignments (in the reversed the elimination order)

$$
\begin{aligned}
x_1^* &= \mathrm{argmax}_{x_1}\Big(\psi(x_1)m_{2\to 1}(x_1)\Big)\\
x_2^* &= \mathrm{argmax}_{x_2}\Big(\psi(x_2)\psi(x_1^*,x_2)m_{3\to 2}(x_2)m_{4\to 2}(x_2)\Big)\\
x_3^* &= \mathrm{argmax}_{x_3}\Big(\psi(x_2^*,x_3)\psi(x_3)\Big)\\
x_4^* &= \mathrm{argmax}_{x_4}\Big(\psi(x_2^*,x_4)\psi(x_4)\Big)
\end{aligned}
$$

Variable elimination — factor graphical models

Works too.

Replace the ψ by factors $f_1, f_2, \ldots$

Message Passing

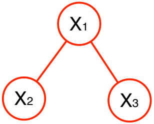
Reuse the intermediate results (called messages) of VE
$\Rightarrow$ Message Passing:

- VE for marginal inference $\Rightarrow$ sum-product message passing
- VE for MAP inference $\Rightarrow$ max-product message passing

Revisit VE for marginal

Assume $$P(x_1, x_2, x_3) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3)$$

$$\begin{aligned}
P(x_1) &= \frac{1}{Z}\psi(x_1)\sum_{x_2}\Big(\psi(x_1, x_2)\psi(x_2)\Big)\sum_{x_3}\Big(\psi(x_1, x_3)\psi(x_3)\Big) \\
&= \frac{1}{Z}\psi(x_1)m_{2\to 1}(x_1)m_{3\to 1}(x_1)
\end{aligned}$$



$$\begin{aligned}
P(x_2) &= \frac{1}{Z}\psi(x_2)\sum_{x_1}\Big(\psi(x_1, x_2)\psi(x_1)\sum_{x_3}\Big[\psi(x_1, x_3)\psi(x_3)\Big]\Big) \\
&= \frac{1}{Z}\psi(x_2)\sum_{x_1}\psi(x_1, x_2)\psi(x_1)m_{3\to 1}(x_1) \\
&= \frac{1}{Z}\psi(x_2)m_{1\to 2}(x_2)
\end{aligned}$$

$m_{3\to 1}(x_1)$ can be reused instead of computing twice.

Sum-product

Can we compute all messages first, and then use them to compute all marginal distributions?

Sum-product

Can we compute all messages first, and then use them to compute all marginal distributions?

Yes, it's called sum-product.

Sum-product

Can we compute all messages first, and then use them to compute all marginal distributions?

Yes, it's called sum-product.

In general,

$$P(x_i) = \frac{1}{Z}\Big(\psi(x_i) \prod_{j \in Ne(i)} m_{j \to i}(x_i)\Big)$$

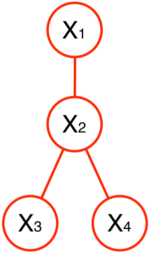
$$m_{j \to i}(x_i) = \sum_{x_j} \Big(\psi(x_j)\psi(x_i, x_j) \prod_{k \in Ne(j) \setminus \{i\}} m_{k \to j}(x_j)\Big)$$

$Ne(i)$: neighbouring nodes of i (*i.e.* nodes that connect with i).

Revisit VE for MAP

$$
\begin{aligned}
&\max_{x_1,x_2,x_3,x_4} P(x_1,x_2,x_3,x_4) \\
&= \max_{x_1,x_2,x_3,x_4} \psi(x_1,x_2)\psi(x_2,x_3)\psi(x_2,x_4)\psi(x_1)\psi(x_2)\psi(x_3)\psi(x_4) \\
&= \max_{x_1,x_2} \left[\ldots \max_{x_3} \Big(\psi(x_2,x_3)\psi(x_3)\Big) \max_{x_4} \Big(\psi(x_2,x_4)\psi(x_4)\Big) \right] \\
&= \max_{x_1} \left[\psi(x_1) \max_{x_2} \Big(\psi(x_2)\psi(x_1,x_2) m_{3\to 2}(x_2) m_{4\to 2}(x_2)\Big) \right] \\
&= \max_{x_1} \Big(\psi(x_1) m_{2\to 1}(x_1)\Big)
\end{aligned}
$$

$$
\begin{aligned}
x_1^* &= \operatorname{argmax}_{x_1} \Big(\psi(x_1) m_{2\to 1}(x_1)\Big) \\
x_2^* &= \operatorname{argmax}_{x_2} \Big(\psi(x_2)\psi(x_1^*,x_2) m_{3\to 2}(x_2) m_{4\to 2}(x_2)\Big) \\
x_3^* &= \operatorname{argmax}_{x_3} \Big(\psi(x_2^*,x_3)\psi(x_3)\Big) \\
x_4^* &= \operatorname{argmax}_{x_4} \Big(\psi(x_2^*,x_4)\psi(x_4)\Big)
\end{aligned}
$$



Max-product

Variable elimination for MAP $\Rightarrow$ Max-product:

$$x_i^* = \operatorname*{argmax}_{x_i} \left(\psi(x_i) \prod_{j \in Ne(i)} m_{j \to i}(x_i) \right)$$

$$m_{j \to i}(x_i) = \max_{x_j} \left(\psi(x_j) \psi(x_i, x_j) \prod_{k \in Ne(j) \setminus \{i\}} m_{k \to j}(x_j) \right)$$

$Ne(i)$: neighbouring nodes of i (*i.e.* nodes that connect with i).

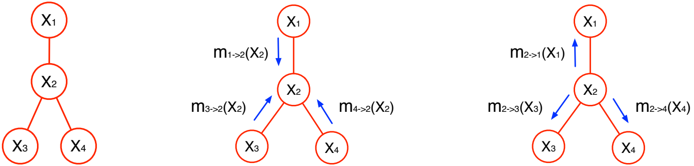
$Ne(j) \setminus \{i\} = \emptyset$ if j has only one edge connecting it. *e.g.* x_1, x_3, x_4. For such node j,

$$m_{j \to i}(x_i) = \max_{x_j} \left(\psi(x_j) \psi(x_i, x_j) \right)$$

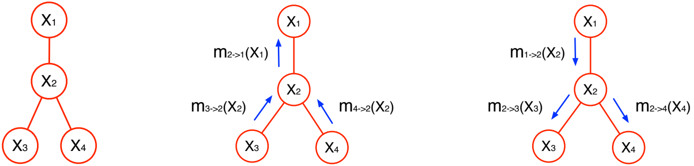
Easier computation!

Max-product

Order matters: message $m_{2\to3}(x_3)$ requires $m_{1\to2}(x_2)$ and $m_{4\to2}(x_2)$.



Alternatively, leaves to root, and root to leaves.



Extension

To avoid over/under flow, often operate in the log space.

Max/sum-product is also known as Message Passing and Belief Propagation (BP).

In graphs with loops, running BP for several iterations is known as Loopy BP (no longer exact: neither convergence nor optimal guarantee in general).

Extend to Junction Tree Algorithm (exact, but expensive) and Clusters-based BP.

That's all

Thanks!