Tutorial 9 - MAP and MPE inference

COMP9418 – Advanced Topics in Statistical Machine Learning

Lecturer: Gustavo Batista

Lecture: MAP and MPE inference

Topic: Questions from lecture topics

Last revision: Sunday 15th November, 2020 at 05:49

Question 1

Given the following Bayesian network

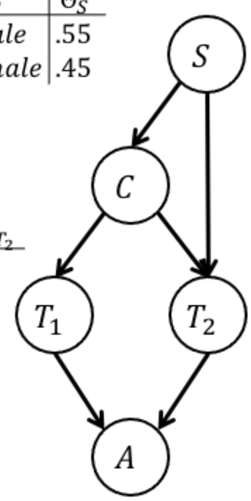
S	C	$\Theta_{S\|C}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
$male$	.55
$female$	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



a. Compute the MPE probability and corresponding MPE instantiation given evidence $A = no$.
b. What is the most likely outcome of the two tests T_1 and T_2 for a female on whom the tests came differently?

Answer

a. Let us use the algorithm MPE VE to solve this query. This algorithm is in slide 12 from lecture 12.

   We start deciding on an elimination order. We use the min-degree heuristic and select variables in the following order: A, S, C, T_1, T_2. Next, we set the evidence $A = no$ by eliminating the rows in CPT $P(A|T_1, T_2)$ that do not match the evidence.

T_1	T_2	A	$P(A = no \mid T_1, T_2)$
ve	ve	no	0
ve	$\bar{ve}$	no	1
$\bar{ve}$	ve	no	1
$\bar{ve}$	$\bar{ve}$	no	0

As A is not present in any other factor, we can eliminate it directly from $P(A = no|T_1, T_2)$

T_1	T_2	$\tau_1(T_1, T_2)$
ve	ve	0
ve	$\bar{ve}$	1
$\bar{ve}$	ve	1
$\bar{ve}$	$\bar{ve}$	0

Here, we are not storing in the extended factor the evidence $A = no$. This procedure conforms with the definition of MPE instantiation query:

$$MPE(\mathbf{e}) = argmax_{\mathbf{q}} P(\mathbf{q},\mathbf{e})$$

However, as you will see in the practical part of the tutorial, our implementation of MPE query includes the evidence in the MPE instantiation. This decision was made to make the implementation simple.

The next variable to be eliminated is S. This variable is present in three factors: $P(S), P(C|S)$, and $P(T_2|S, C)$. Let us start joining $P(S)$ and $P(C|S)$ resulting in $\sigma_1(S, C)$:

S	C	$\sigma_1(S, C)$
$male$	yes	.0275
$male$	no	.5225
$female$	yes	.0045
$female$	no	.4455

Now we join $\sigma_1(S, C)$ and $P(T_2|S, C)$ to get $\sigma_2(S, C, T_2)$:

S	C	T_2	$\sigma_2(S, C, T_2)$
$male$	yes	ve	.022
$male$	yes	$\bar{ve}$	.0055
$male$	no	ve	.1045
$male$	no	$\bar{ve}$	.418
$female$	yes	ve	.004275
$female$	yes	$\bar{ve}$	.000225
$female$	no	ve	.022275
$female$	no	$\bar{ve}$	.423225

We are in position to eliminate S by maximization of $\sigma_2(S, C, T_2)$ resulting in $\tau_2(C, T_2)$

C	T_2	$\tau_2(C, T_2)$	$\tau_2[C, T_2]$
yes	ve	.022	$(S = male)$
yes	$\bar{v}e$	.0055	$(S = male)$
no	ve	.1045	$(S = male)$
no	$\bar{v}e$	.423225	$(S = female)$

It is time to eliminate C. Let us join the two factors that involve C: $P(T_1|C)$ and $\tau_2(C, T_2)$.

C	T_1	T_2	$\sigma_3(C, T_1, T_2)$	$\sigma_3[C, T_1, T_2]$
yes	ve	ve	.0176	$(S = male)$
yes	ve	$\bar{v}e$	.0044	$(S = male)$
yes	$\bar{v}e$	ve	.0044	$(S = male)$
yes	$\bar{v}e$	$\bar{v}e$	.0011	$(S = male)$
no	ve	ve	.0209	$(S = male)$
no	ve	$\bar{v}e$	.084645	$(S = female)$
no	$\bar{v}e$	ve	.0836	$(S = male)$
no	$\bar{v}e$	$\bar{v}e$	.33858	$(S = female)$

And now we maximize out C.

T_1	T_2	$\tau_3(T_1, T_2)$	$\tau_3[T_1, T_2]$
ve	ve	.0209	$(S = male, C = no)$
ve	$\bar{v}e$	.084645	$(S = female, C = no)$
$\bar{v}e$	ve	.0836	$(S = male, C = no)$
$\bar{v}e$	$\bar{v}e$	.33858	$(S = female, C = no)$

We move on and eliminate T_1 by first joining $\tau_3(T_1, T_2)$ and $\tau_1(T_1, T_2)$.

T_1	T_2	$\sigma_4(T_1, T_2)$	$\sigma_4[T_1, T_2]$
ve	ve	0	$(S = male, C = no)$
ve	$\bar{v}e$	.084645	$(S = female, C = no)$
$\bar{v}e$	ve	.0836	$(S = male, C = no)$
$\bar{v}e$	$\bar{v}e$	0	$(S = female, C = no)$

And we maximize out T_1.

T_2	$\tau_4(T_2)$	$\tau_4[T_2]$
ve	.0836	$(S = male, C = no, T_1 = \bar{v}e)$
$\bar{v}e$	.084645	$(S = female, C = no, T_1 = ve)$

Finally, we maximize out T_2 from its only factor $\tau_4(T_2)$.

$\tau_5(\,)$	$\tau_5[\,]$
.084645	$(S = female, C = no, T1 = ve, T2 = \bar{v}e)$

b. This second query is a MAP with evidence $S = female, A = no$ and MAP variables T_1 and T_2. We will use the MAP VE algorithm. For reference, this algorithm is listed in slide 64 in lecture 12. We use the same elimination order A, S, C, T_1, T_2 since the MAP variables T_1 and T_2 are the last ones in the order.

We start setting the evidence by eliminating rows. Similarly to (a), we set $A = no$.

T_1	T_2	A	$P(A = no \vert T_1, T_2)$
ve	ve	no	0
ve	$\bar{ve}$	no	1
$\bar{ve}$	ve	no	1
$\bar{ve}$	$\bar{ve}$	no	0

But we also need to set $A = female$. This will change three factors: $P(S)$, $P(C|S)$ and $P(T_2|S, C)$.

S	$P(A = female)$
$female$	.45

S	C	$P(C \vert S = female)$
$female$	yes	.01
$female$	no	.99

S	C	T_2	$P(T_2 \vert S = female, C)$
$female$	yes	ve	.95
$female$	yes	$\bar{ve}$	.05
$female$	no	ve	.05
$female$	no	$\bar{ve}$	.95

As A is not present in any other factor, we can eliminate it directly from $P(A = no|T_1, T_2)$

T_1	T_2	$\tau_1(T_1, T_2)$
ve	ve	0
ve	$\bar{ve}$	1
$\bar{ve}$	ve	1
$\bar{ve}$	$\bar{ve}$	0

We eliminate variable C by first joining $P(S = female)$ and $P(C|S = female)$.

S	C	$\sigma_1(S, C)$
$female$	yes	.0045
$female$	no	.4455

And now we join $\sigma_1(S, C)$ and $P(T_2, S = female, C)$

S	C	T_2	$\sigma_2(S, C, T_2)$
$female$	yes	ve	.004275
$female$	yes	$\bar{ve}$	.000225
$female$	no	ve	.022275
$female$	no	$\bar{ve}$	.423225

We can now eliminate S by summation.

C	T_2	$\tau_2(C, T_2)$
yes	ve	.004275
yes	$\bar{ve}$	.000225
no	ve	.022275
no	$\bar{ve}$	.423225

We move on and eliminate C. We join the recently created factor $\tau_2(C, T_2)$ with $P(T_1|C)$.

C	T_1	T_2	$\sigma_3(C, T_1, T_2)$
yes	ve	ve	.00342
yes	ve	$\bar{ve}$	.00018
yes	$\bar{ve}$	ve	.000855
yes	$\bar{ve}$	$\bar{ve}$	.000045
no	ve	ve	.004455
no	ve	$\bar{ve}$	.084645
no	$\bar{ve}$	ve	.01782
no	$\bar{ve}$	$\bar{ve}$	.33858

And we eliminate C by summation.

T_1	T_2	$\tau_3(T_1, T_2)$
ve	ve	.007875
ve	$\bar{ve}$	.084825
$\bar{ve}$	ve	.018675
$\bar{ve}$	$\bar{ve}$	.338625

Now, it is time to eliminate T_1, out first MAP variable. The elimination from now on will be by maximization. We first join $\tau_3(T_1, T_2)$ and $\tau_1(T_1, T_2)$.

T_1	T_2	$\sigma_4(T_1, T_2)$
ve	ve	0
ve	$\bar{ve}$	.084825
$\bar{ve}$	ve	.018675
$\bar{ve}$	$\bar{ve}$	0

And eliminate T_1 by maximization, taking note of the instantiation in the extended field.

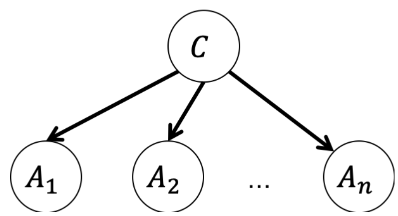
T_2	$\tau_4(T_2)$	$\tau_4[T_2]$
ve	.018675	$(T1 = \bar{ve})$
$\bar{ve}$	.084825	$(T1 = ve)$

Finally, we eliminate T_2 by maximizing $\tau_4(T_2)$ that is the only factor that refers to T_2,

$\tau_5(\,)$	$\tau_5[\,]$
.084825	$(T1 = ve, T2 = \bar{ve})$

Question 2

Consider the naive Bayes stucture



What is the complexity of computing the MPE probability for this network using algorithm MPE VE? What is the complexity of computing the MAP probability for MAP variables $A_1, \ldots, A_n$ using algorithm MAP VE? How does the complexity of these computations change when we have evidence on variable C?

Answer

The best elimination order for the MPE query and naive Bayes structure is $A_1, \ldots, A_n, C$ that results in an elimination width $w = 1$. Therefore, the MPE algorithm has time and space complexities of $O(n)$.

For the MAP algorithm, we have to change the elimination order, moving the MAP variables to the end of the ordering. The elimination order $C, A_1, \ldots, A_n$ has width $w = n$. Therefore, the time and space complexities are $O(n \exp(w))$.

If we set C as evidence, then the MPE and MAP queries become the same. Both will answer the most likely instantiation for the variables $A_1, \ldots, A_n$ given evidence on C. The time and space complexities are $O(n)$.

Question 3

Can we compute a MAP instantiation $MAP(\mathbf{M}, e)$ by computing the MAP instantiations $\mathbf{m}_x = MAP(\mathbf{M}, \mathbf{e}x)$ for every value of x of some variable $X \notin \mathbf{M}$ and then returning $argmax_{\mathbf{m}_x} P(\mathbf{m}_x, \mathbf{e})$? What if $X \in \mathbf{M}$? Justify your response.

In this question, $\mathbf{M}$ is the set of query variables, $\mathbf{e}$ is the evidence instantiation. $\mathbf{e}x$ is notation for adding the instantiation x of the variable X to the evidence. $\mathbf{m}_x$ is the MAP instantiation of the query variables $\mathbf{M}$, when x has been added to the evidence. $P(\mathbf{m}_x, \mathbf{e})$ is the probability of all the variables being set to the values in $\mathbf{m}_x$ and $\mathbf{e}$, also called the MAP probability (Notation was $MAP_P(\mathbf{M}, \mathbf{e})$ in lectures).

For example, if X has 3 possible outcomes (1,2 and 3), then we would compute 3 MAP instantiations ($\mathbf{m}_1$,$\mathbf{m}_2$ and $\mathbf{m}_3$). Then we would choose the instantiation that has the highest MAP probability, and return that instantiation.

Answer

For the first case, the answer is no. If we adopt this procedure, we are effectively pushing a non-MAP variable X to be eliminated last using a maximization.

In the second case, the answer is yes. Using the specified procedure, we are splitting the problem according to the different values of the MAP variable X. Later on, we join these parts together in a single factor with $|X|$ entries. Each entry has the answer for MAP query with evidence set to $\mathbf{e}$ and $X = x_1$. Maximizing this last factor has the same effect as leaving X to be the last eliminated variable.

Question 4

Consider the classical jointree algorithm as defined by the following equations for message M_{ij} and factor f_i

$$M_{ij} = \sum_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \Phi_i \prod_{k \neq j} M_{ki}$$

$$f_i(\mathbf{C}_i) = \Phi_i \prod_{k} M_{ki}$$

Suppose we replace all summations by maximizations when computing messages,

$$M_{ij} = \max_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \Phi_i \prod_{k \neq j} M_{ki}$$

What are the semantics of the factor $f_i(\mathbf{C}_i)$ in this case? In particular, can we use it to recover answers to MPE queries?

Answer

Each factor f_i stores a max-marginal over variables in cluster $\mathbf{C}_i$. In other words, f_i stores the marginals for variables in $\mathbf{C}_i$ after we maximize out all other varibles in the network not present in $\mathbf{C}_i$.

Any factor f_i can be used to answer an MPE query. It is necessary to maximize out the variables in $\mathbf{C}_i$. If we want to answer a query with evidence and all evidence variables are in the same cluster $\mathbf{C}_i$, then we can select the rows of $\mathbf{C}_i$ that match the evidence before maximize out the variables in the cluster.

Question 5

Consider the classical jointree algorithm as defined by the following equations for message M_{ij} and factor f_i

$$M_{ij} = \sum_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \Phi_i \prod_{k \neq j} M_{ki}$$

$$f_i(\mathbf{C}_i) = \Phi_i \prod_{k} M_{ki}$$

Suppose we replace all summations over MAP variables $\mathbf{M}$ by maximizations when computing messages, as follows:

$$M_{ij} = \max_{(\mathbf{C}_i \setminus \mathbf{S}_{ij}) \cap \mathbf{M}} \sum_{(\mathbf{C}_i \setminus \mathbf{S}_{ij}) \setminus \mathbf{M}} \Phi_i \prod_{k \neq j} M_{ki}$$

What are the semantics of the factor $f_i(\mathbf{C}_i)$ in this case? In particular, can we use it to recover answers to MAP queries?

Answer

Notice that when we expand the product for factor f_i, including the incoming messages M_{ki} each message will be composed by a sequence of maximization and summation operations. To provide a correct response to MAP queries, we need to guarantee that the summation operations will precede the maximizations. There exist jointrees that satisfy the constrained ordering, and these structures can be used to solve MAP problems exactly using the modified equations. However, for general jointrees the constrained order is not respected. However, we still can marginalize each cluster to get upper bounds for its associated MAP variables.