BSS2, Lecture #2

Answers to exercises

19 September 1999

Exercise 1 Let $W \subseteq U$, and let ϕ be a potential on U. The expression $\sum_{U \setminus W} \phi$ denotes the *(sum-)margin* of ϕ on W. Sometimes the domain of potentials are indicated explicitly: $\phi_U \equiv \phi$ and $\phi_W = \sum_{U\setminus W} \phi_U$. The marginal potential ϕ_W is defined by, for $x \in \mathcal{W} = \mathrm{Sp}(W)$,

$$\phi_W(x) = \left(\sum_{U \setminus W} \phi_U \right)(x) = \sum_{z \in \mathcal{U} \setminus \mathcal{W}} \phi_U(z.x),$$

where $z.x$ is the element in $\mathcal{U} = \mathrm{Sp}(U)$ with projections x to $\mathcal{W}$ and z to $\mathcal{U} \setminus W = \mathrm{Sp}(U \setminus W)$. Max-marginalization is defined similarly:

$$\phi_W^{\max}(x) = \left(\underset{U \setminus W}{\mathsf{M}} \phi_U \right)(x) = \max_{z \in \mathcal{U} \setminus \mathcal{W}} \phi_U(z.x).$$

Let $U = \{A, B, C, D\}$, $W = \{A, C\}$, and let ϕ_U be given by the table below.

		c_1 d_1	c_1 d_2	c_2 d_1	c_2 d_2
a_1	b_1	0.0957	0.0672	0.0341	0.0513
a_1	b_2	0.1021	0.0162	0.0634	0.1287
a_2	b_1	0.0174	0.1297	0.0040	0.1089
a_2	b_2	0.0624	0.0776	0.0307	0.0107

(a) Compute ϕ_W and $\phi_W^{\max}$.

(b) Compute $x_U^{\max} = \arg\max_{x \in \mathcal{U}} \phi_U$ and $(x_A^{\max}, x_B^{\max}, x_C^{\max}, x_D^{\max})$.

(c) Indicate the table for the evidence function $\mathcal{E}_D$ (defined on $\mathcal{D}$), representing the evidence "D is in state d_1".

(d) Compute $\phi_U * \mathcal{E}_D$.

(e) Compute the normalization constant, μ, in $P(U \mid e) = \mu * \phi_U * \mathcal{E}_D$.

Answer: Re (a):

ϕ_W :

	c_1	c_2
a_1	0.2812	0.2775
a_2	0.2871	0.1543

$\phi_W^{\max}$:

	c_1	c_2
a_1	0.1021	0.1287
a_2	0.1297	0.1089

Re (b): $\max_{x \in \mathcal{U}} \phi_U(x) = 0.1297$ for $x_U^{\max} = (a_2, b_1, c_1, d_2)$. Since, $\phi_A = (0.56, 0.44)$, $\phi_B = (0.51, 0.49)$, $\phi_C = (0.57, 0.43)$, and $\phi_D = (0.41, 0.59)$, we get $(x_A^{\max}, x_B^{\max}, x_C^{\max}, x_D^{\max}) = (a_1, b_1, c_1, d_2)$, so, in general, $x_U^{\max} \neq (x_{V_1}^{\max}, \ldots, x_{V_n}^{\max})$ for $U = \{V_1, \ldots, V_n\}$.

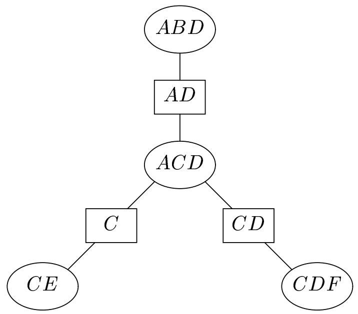
Re (c): The table is given by the vector $(1, 0)$.

Re (d): $\phi_U * \mathcal{E}_D$ is obtained by setting the contents of all entries of the table for ϕ_U for which $D \neq d_1$ to zero:

		c_1 d_1	c_1 d_2	c_2 d_1	c_2 d_2
a_1	b_1	0.0957	0	0.0341	0
a_1	b_2	0.1021	0	0.0634	0
a_2	b_1	0.0174	0	0.0040	0
a_2	b_2	0.0624	0	0.0307	0

Re (e): $\mu^{-1} = \sum_U \phi_U * \mathcal{E}_D = \phi_D(d_1) = 0.41.$ $\square$

Exercise 2 Calculate the number of arithmetic operations by a call to COLLECT EVIDENCE in clique ABD in the junction tree below, assuming sum-marginalization and binary variables.



Answer: The number of arithmetic operations by a call to COLLECT EVIDENCE in clique ABD is

Operation	Additions	Divisions	Multiplications
$\phi^*_{\{C,D\}} = \sum_F \phi_{\{C,D,F\}}$	4		
$\lambda_{\{C,D\}} = \dfrac{\phi^*_{\{C,D\}}}{\phi_{\{C,D\}}}$		4	
$\phi^*_{\{C\}} = \sum_E \phi_{\{C,E\}}$	2		
$\lambda_{\{C\}} = \dfrac{\phi^*_{\{C\}}}{\phi_{\{C\}}}$		2	
$\phi^*_{\{A,C,D\}} = \phi_{\{A,C,D\}}\lambda_{\{C,D\}}\lambda_{\{C\}}$			16
$\phi^*_{\{A,D\}} = \sum_C \phi^*_{\{A,C,D\}}$	4		
$\lambda_{\{A,D\}} = \dfrac{\phi^*_{\{A,D\}}}{\phi_{\{A,D\}}}$		4	
$\phi^*_{\{A,B,D\}} = \phi_{\{A,B,D\}}\lambda_{\{A,D\}}$			8
Total:	10	10	24

Notice that the number of multiplications in computing $\phi^*_{\{A,C,D\}}$ can be reduced to 12 by first multiplying the $\lambda_{\{C\}}$-table onto the $\lambda_{\{C,D\}}$-table. This trick can, however, only be used when calling COLLECT EVIDENCE from clique ABD. (The trick is not used in Hugin.) □

Exercise 3 Let $\mathcal{T}$ be a consistent junction tree with cliques $C_1 = \{A, B\}$ and $C_2 = \{B, C\}$, and clique tables given as follows.

	b_1	b_2
a_1	0.07	0.37
a_2	0.28	0.28

	b_1	b_2
c_1	0.13	0.24
c_2	0.22	0.41

(a) Compute $t^{\max}_{C_1}$ and $t^{\max}_{C_2}$ (i.e., the max-margins obtained through max-propagation).

(b) From max-equilibrium, perform sum-propagation. Compare the resulting tables with the ones above.

Answer: Re (a): Let's perform a max-propagation with C_1 as root (S will denote the separator between C_1 and C_2):

(1) Compute message $t^*_S \leftarrow \mathop{\mathsf{M}}_{A} t_{C_1} = (0.28, 0.37)$ from C_1 to C_2.

(2) Compute ratio $t^*_S/t_S = (0.800000, 0.569231)$.

(3) Absorb message: $t_{C_2} \leftarrow t_{C_2} * \dfrac{t^*_S}{t_S}$

$t^{\max}_{C_2} = t_{C_2}$:

	b_1	b_2
c_1	0.104000	0.136615
c_2	0.176000	0.233385

(4) Let $t_S \leftarrow t^*_S$.

(5) Compute message $t^*_S \leftarrow \mathop{\mathsf{M}}_{C} t^{\max}_{C_2} = (0.176000, 0.233385)$ from C_2 to C_1.

(6) Compute ratio $t^*_S/t_S = (0.628571, 0.630770)$.

(7) Absorb message: $t_{C_1} \leftarrow t_{C_1} * \dfrac{t^*_S}{t_S}$

$t^{\max}_{C_1} = t_{C_1}$:

	b_1	b_2
a_1	0.044000	0.233385
a_2	0.176000	0.176616

(8) Let $t_S \leftarrow t^*_S$.

Re (b): Next we perform sum-propagation with C_1 as root:

(1) Compute message $t^*_S \leftarrow \sum_A t_{C_1} = (0.22, 0.41)$ from C_1 to C_2.

(2) Compute ratio $t^*_S/t_S = (1.250000, 1.756754)$.

(3) Absorb message: $t_{C_2} \leftarrow t_{C_2} * \dfrac{t^*_S}{t_S}$

t_{C_2} :

	b_1	b_2
c_1	0.13	0.24
c_2	0.22	0.41

(4) Let $t_S \leftarrow t^*_S$.

(5) Compute message $t^*_S \leftarrow \sum_C t_{C_2} = (0.35, 0.65)$ from C_2 to C_1.

(6) Compute ratio $t^*_S/t_S = (1.590909, 1.585366)$.

(7) Absorb message: $t_{C_1} \leftarrow t_{C_1} * \dfrac{t^*_S}{t_S}$

t_{C_1} :

	b_1	b_2
a_1	0.07	0.37
a_2	0.28	0.28

(8) Let $t_S \leftarrow t_S^*$.

And so we observe that the original tables have been reestablished. □

Exercise 4 Let g be a probability function over V, let $(x_1, \dots, x_n)$ be an arbitrary but fixed ordering of $\mathrm{Sp}(V)$, and let L be a uniformly distributed random variable on the interval $]0; 1]$. On the basis of a simulated value l for L (generated by a (pseudo) random function), we can then determine a simulated value $x \in \mathrm{Sp}(V)$ for V (i.e., x is drawn with probability $g(x)$) as follows.

(1) Let $i = 1$ and sum $= 0$.

(2) While sum $< l$ do

  (a) sum $:=$ sum $+ g(x_i)$

  (b) $i := i + 1$.

(3) Let $x = x_{i-1}$.

Simulation in a junction tree, $\mathcal{T}$, is performed through outward propagation from a root clique, C^*. Let C be a clique in the tree with neighbour C' such that C' is located on the path from C to C^* and let $S = C \cap C'$. When configuration x_S (i.e., values) for variables S has been selected a configuration for $C \setminus S$ is found using the above algorithm with the modification that L must be uniformly distributed on $]0; \sigma]$, where $\sigma = \sum_{C \setminus S} f_C(x_S)$ and f is a probability function factorizing on $\mathcal{T}$.

Now, let $\mathcal{T}$ consist of two cliques $C_1 = \{X, Y, Z\}$ and $C_2 = \{Y, Z, W\}$ interconnected by separator S. Let f_{C_1} and f_{C_2} be given as follows.

f_{C_1}	y_1		y_2	
	z_1	z_2	z_1	z_2
x_1	0.048218	0.128582	0.063861	0.099339
x_2	0.140174	0.021026	0.037200	0.111600
x_3	0.117000	0.065000	0.048000	0.120000

f_{C_2}	z_1		z_2	
	w_1	w_2	w_1	w_2
y_1	0.133609	0.171783	0.067065	0.147543
y_2	0.087683	0.061378	0.141831	0.189108

Let $((x_1, y_1, z_1), (x_1, y_1, z_2), (x_1, y_2, z_1), (x_1, y_2, z_2), (x_2, y_1, z_1), (x_2, y_1, z_2), (x_2, y_2, z_1), (x_2, y_2, z_2), (x_3, y_1, z_1), (x_3, y_1, z_2), (x_3, y_2, z_1), (x_3, y_2, z_2))$ and $((y_1, z_1, w_1), (y_1, z_1, w_2), (y_1, z_2, w_1), (y_1, z_2, w_2), (y_2, z_1, w_1), (y_2, z_1, w_2), (y_2, z_2, w_1), (y_2, z_2, w_2))$ be orderings of, respectively, $\mathrm{Sp}(C_1)$ and $\mathrm{Sp}(C_2)$.

(a) Find the simulated values for X, Y, Z and W through "random propagation" from C_1 to C_2, using these orderings and the "random" numbers 0.67 and 0.22 (in that order) in the above algorithm.

(b) Hugin requires the junction tree to be in sum-equilibrium before random propagation is initiated. What do you think might be the reason for that?

Answer: Re (a): When ran on f_{C_1} with $l = 0.67$ the algorithm stops when $i = 11$, since the variable 'sum' has value 0.65 for $i = 10$ and 0.767 for $i = 11$. That is, $X = x_3$, $Y = y_1$, and $Z = z_1$. Now, since $f_{C_2}(y_1, z_1, W) = (0.133609, 0.171783)$ and $l = 0.22$, we get $W = w_2$.

Re (b): Whenever a junction tree is in sum-equilibrium we get $\sigma = f_S(x_S)$, where $x \in \mathrm{Sp}(C')$ is the configuration for C' that has been "randomly" chosen and where S is the separator between C' and a neighbour C in which a configuration for $C \setminus S$ is to be chosen next. That is, we do not need to compute $\sum_{C \setminus S} f_C(x_S)$ to determine σ, making it possible to implement "random propagation" more efficiently. □

Exercise 5 Prim's algorithm for constructing a maximal spanning tree (e.g. a junction tree over a junction graph) is given as follows.

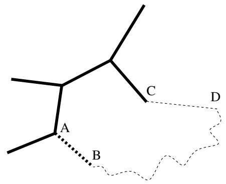
(1) Start with a tree, $\mathcal{T}$, containing a single node.

(2) Successively, add a node, V, to $\mathcal{T}$ whose intersection with a node of $\mathcal{T}$ is maximal.

(3) Link V to that node.

Show that Prim's algorithm always yields a maximum spanning tree.

Answer: Let $\mathcal{T}_s \subseteq \mathcal{T}$ be an arbitrary subtree constructed by Prim's algorithm, where $\mathcal{T}$ is a maximal spanning tree. Now, assume that the algorithm chooses an edge $\{A, B\}$ for which A belongs to $\mathcal{T}_s$ such that there does not exist a maximal spanning tree with $\mathcal{T}_s'$ as a subtree ($\mathcal{T}_s'$ is $\mathcal{T}_s$ extended with the edge $\{A, B\}$). Since $\mathcal{T}$ is a tree, there must exist a unique path in $\mathcal{T}$ from B to a node C in $\mathcal{T}_s$ such that $\{C, D\}$ is an edge in $\mathcal{T}$, but not in $\mathcal{T}_s$ (see figure below).



Since the algorithm chose the edge $\{A, B\}$ rather than $\{C, D\}$, it must be true that $W(A, B) \geq W(C, D)$ (as $\mathcal{T}$ is maximal, equality must hold), where $W(X, Y)$ is the weight of edge $\{X, Y\}$. That is, we may replace the edge $\{C, D\}$ in $\mathcal{T}$ by the edge $\{A, B\}$, such that the resulting tree is maximal and such that $\mathcal{T}_s'$ is a subtree of this tree. In conclusion, an arbitrary subtree generated by Prim's algorithm will always be a subtree of a maximal spanning tree. □