10-708 PGM (Spring 2019): Homework 1 v1.1

Andrew ID: Moroccan Student
Name: [Belcaid Anass]
Collaborators: [Working alone]

1 Bayesian Networks [20 points] (Xun)

State True or False, and briefly justify your answer in a few sentences. You can cite theorems from Koller and Friedman (2009). Throughout the section, P is a distribution and $\mathcal{G}$ is a BN structure.

1. [**2 points**] If $A \perp B \mid C$ and $A \perp C \mid B$, then $A \perp B$ and $A \perp C$. (Suppose the joint distribution of A, B, C is positive.)

Solution

True, as this a special case for the **contraction** theorem on Koller and Friedman (2009).

$$(X \perp Y|Z, W) \;\&\; (X \perp Z|Y, W) \Longrightarrow (X \perp Y, Z|W)$$

We simply take $X = A$, $B = Y$, $C = Z$ and W is a sure event. We then get

$$(A \perp B, C)$$

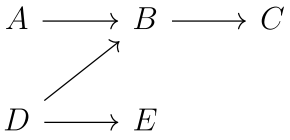


Figure 1: A Bayesian network.

2. [**2 points**] In Figure 1, $E \perp C \mid B$.

Solution

True, As any path to C is blocked by B the unique parent to C.

3. [**2 points**] In Figure 1, $A \perp E \mid C$.

Solution

False, There is an active trail using the V-structure $A \rightarrow B \leftarrow D$. This structure is activated a child of B, which is C, is activated. Hence the trail $A - B - D - E$ is active.

$$\begin{tikzcd} P \text{ factorizes over } \mathcal{G} \arrow[r, Rightarrow, "(1)"] & \mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(P) \arrow[r, Rightarrow, "(2)"] & \mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(P) \arrow[ll, Rightarrow, bend left, "(3)"] \end{tikzcd}$$

Figure 2: Some relations in Bayesian networks.

4. [**2 points**] In Figure 2, relation (1) is true.

Solution

True, direct result of the theorem (3.2) on Koller and Friedman (2009).

5. [**2 points**] In Figure 2, relation (2) is true.

Solution

True, If the set of independences by the d-separation are verified in a distribution P, then P must also verify the local independences in the Bayesian graph.

6. [**2 points**] In Figure 2, relation (3) is true.

Solution

True. Direct result of theorem (3.1) on Koller and Friedman (2009)

7. [**2 points**] If $\mathcal{G}$ is an I-map for P, then P may have extra conditional independencies than $\mathcal{G}$.

Solution

True, As the definition of I-map only forces P to verify all the of independences in $\mathcal{G}$. P could have additional hidden independences that are not reflected in the graph.

8. [**2 points**] Two BN structures $\mathcal{G}_1$ and $\mathcal{G}_2$ are I-equivalent iff they have the same skeleton and the same set of v-structures.

Solution

False, the v-structure is not sufficient here, as in one graph the parents could be connected and not in the other one. The correct theorem (3.8) Koller and Friedman (2009) states that two graphs are I-equivalent if they have the same skeleton and the same **immoralities**.

9. [**2 points**] The minimal I-map of a distribution is the I-map with fewest edges.

Solution

False, a minimal map simply means that it can accept an edge removal to be an I-map. Since this map is not unique, we could find several I-maps with different number of edges.

10. [**2 points**] The P-map of a distribution, if exists, is unique.

Solution

False. From (Koller and Friedman, 2009, page 84), the P-map is unique up to I-equivalence between networks.

2 Undirected Graphical Models [25 points] (Paul)

2.1 Local, Pairwise and Global Markov Properties [18 points]

1. Prove the following properties:

   - [**2 points**] If $A \perp (B, D) \mid C$ then $A \perp B \mid C$.

     Solution

     $$\begin{aligned} P(A,B|C) &= \sum_d P(A,B|C,D)P(D) \\ &= \sum_d P(A|B,C,D)P(B|C,D)P(D) \\ &= \sum_d P(A|C)P(B|C,D)P(D) \\ &= P(A|C)\sum_d P(B|C,D)P(D) \\ &= P(A|C)P(B|C) \end{aligned}$$

   - [**2 points**] If $A \perp (B, D) \mid C$ then $A \perp B \mid (C, D)$ and $A \perp D \mid (B, C)$.

     Solution

     We have

     $$\begin{aligned} P(A,B,D|C) &= P(A,B|D,C)P(D) && (1) \\ &= P(A|C,D)P(B|C,D)P(D) && (2) \end{aligned}$$

     From the equatlity between line (1) and (2), we could conclude that

     $$P(A,B|C,D) = P(A|C,D)P(B|C,D)$$

     the same proof, is applied for D.

   - [**2 points**] For strictly positive distributions, if $A \perp B \mid (C, D)$ and $A \perp C \mid (B, D)$ then $A \perp (B, C) \mid D$.

     Solution

     using the chain rule we can write:

     $$\begin{aligned} P(A,B,C|D) &= P(A|B,C,D)P(B,C|D) \\ &= P(A|D)P(B,C|D,A) \end{aligned}$$

In the secon equation since $A \perp B|(C, D)$ and $A \perp C|(B, D)$, we could conclude that $P(B, C|D, A) = P(B, C|D)$. Now we combine both equations we found that

$$P(A|B, C, D)P(B, C|D) = P(A|D)P(B, C|D)$$

Hence, we conclude that $A \perp (B, C)|D$

2. [**6 points**] Show that for any undirected graph G and distribution P, if P factorizes according to G, then P will also satisfy the global Markov properties of G.

Solution

Let (X, Y, Z) three disjoint sets in $\mathcal{X}$ such as Z separates X and H in the markov model $\mathcal{H}$. We shoud proove that $P \models (X \perp Y|Z)$

(a) As a first case, we consider $X \cup Y \cup Z = \mathcal{X}$, As Z separates X and Y, any clique is either in $X \cup Z$ or $Y \cup Z$. We denote $\mathcal{I}_X$ the set of indices in $X \cup Z$ and do the same $\mathcal{I}_Y$ for Y. Then we could write the ditribution P such as:

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_{i \in \mathcal{I}_X} \phi(D_i) \prod_{i \in \mathcal{I}_Y} \phi(D_i) = \frac{1}{Z} f(X, Z) f(Y, Z)$$

Which proves that X and Y are independant given Z.

(b) If $X \cup Y \cup Z \subset \mathcal{X}$, then we denote the set $U = X - (X \cup Y \cup Z)$. We could divice the $U = U_1 \cup U_2$ such as U_1 is connected to X and U_2 is connected to Y. By doing so, we present the same argument where U_1 will be added to $\mathcal{I}_X$ and U_2 to $\mathcal{I}_Y$.

3. [**6 points**] Show that for any undirected graph G and distribution P, if P satisfies the local Markov property with respect to G, then P will also satisfy the pairwise Markov property of G.

Solution

Let X and Y to be two nodes from the Markov network $\mathcal{H}$.

- If X is connected to Y then by the local markov property we have

$$\big(X \perp Y|MB_{\mathcal{H}}(X)) \implies (X \perp Y|\mathcal{X} - (X, Y)\big)$$

- If X and Y are not connected. They are necesseraly separated by the rest of the nodes in $\mathcal{X}$.

$$\big(X \perp Y|\mathcal{X} - (X, Y)\big)$$

2.2 Gaussian Graphical Models [7 points]

Now we consider a specific instance of undirected graphical models. Let $X = \{X_1, ..., X_d\}$ be a set of random variables and follow a joint Gaussian distribution $X \sim \mathcal{N}(\mu, \Lambda^{-1})$ where $\Lambda \in \mathbb{S}^{++}$ is the precision matrix. Let X_j, X_k be two nodes in X, and $Z = \{X_i \mid i \notin \{j, k\}\}$ denote the remaining nodes. Show that $X_j \perp X_k \mid Z$ if and only if $\Lambda_{jk} = 0$.

Solution

- $(X_j \perp X_k \mid Z) \Longrightarrow \Lambda_{jk} = 0$ Conditioning on Z will reduce the joint probability on (X_j, X_k) to a Gaussian with the same precision Λ which is reduced to the set (j, k) and coefficient are scaled by a factor to ensure normalization.

  $$\Lambda_{ij} = \text{Cst}\begin{pmatrix} \lambda_{ii} & \lambda_{ij} \\ \lambda_{ij} & \lambda_{jj} \end{pmatrix}$$

  The probability $(X_j, X_k|Z)$ is given by

  $$P(X_j, X_k|Z) = \text{Cst}\exp\left(\Lambda_{jj}(x_j - \mu_j)^2 + \Lambda_{kk}(x_k - \mu_k)^2 + 2\Lambda_{jk}(x_k - \mu_k)^T(x_j - \mu_j)\right)$$

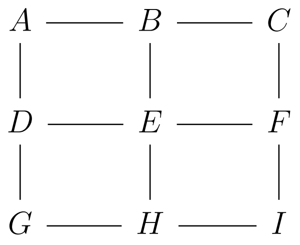
  But we know that $(X_j \perp X_k|Z)$, hence the third term must be null. Which implies that $\Lambda_{jk} = 0$.
- The inverse $\Lambda_{jk} = 0 \Longrightarrow (X_j \perp X_k|Z)$ follows directly from this property as

$$\begin{aligned} P(X_j, X_k|Z) &= \text{Cst}\exp(\Lambda_{jj}(x_j - \mu_j)^2 + \Lambda(x_k - \mu_k)^2) && (3) \\ &= \text{Cst}\exp(\Lambda_{jj}(x_j - \mu_j)^2)\exp(\Lambda(x_k - \mu_k)^2) && (4) \\ &= P(X_j|Z)P(X_k|Z) && (5) \end{aligned}$$

3 Exact Inference [40 points] (Xun)

3.1 Variable elimination on a grid [10 points]

Consider the following Markov network:



We are going to see how *tree-width*, a property of the graph, is related to the intrinsic complexity of variable elimination of a distribution.

1. [**2 points**] Write down largest clique(s) for the elimination order $E, D, H, F, B, A, G, I, C$.

Solution

The larget clique for the this elimination will be (A, B, C, G, I) after eliminating the node **H**.

2. [**2 points**] Write down largest clique(s) for the elimination order $A, G, I, C, D, H, F, B, E$.

Solution

This ordering is amazing keeping the maximum clique at most order **3**. From the first elimination A we get the a maximal clique BDE. After each node will remove a simple node and keep cliques at most of lenght 3.

3. [**2 points**] Which of the above ordering is preferable? Explain briefly.

Solution

The **second** oredering is clearly better as the complexity of the elimination algorithm is determined by n_{max}^{K} where n is the cardinality of the random variables and K_{max} is the size of the maximal clique.

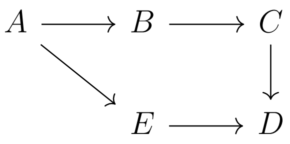
4. [**4 points**] Using this intuition, give a reasonable ($\ll n^2$) upper bound on the tree-width of the $n \times n$ grid.

Solution

Since we process the cliques from the extremities, we assure that the nodes from the left, if we divide the square by half, will never be connected to the right part. Hence the maximal tree width will be at most $\mathbf{\frac{n}{2}}$

3.2 Junction tree in action: part 1 [10 points]

Consider the following Bayesian network $\mathcal{G}$:

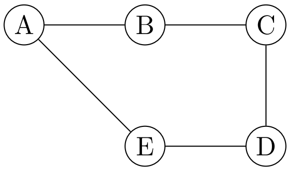


We are going to construct a junction tree $\mathcal{T}$ from $\mathcal{G}$. Please sketch the generated objects in each step.

1. [**1 pts**] Moralize $\mathcal{G}$ to construct an undirected graph $\mathcal{H}$.

Solution

Since each node contains at most one parent, the **Moralized graph:** is simply the undirected graph

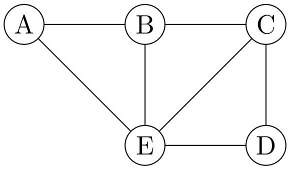


2. [**3 pts**] Triangulate $\mathcal{H}$ to construct a chordal graph $\mathcal{H}^*$.

(Although there are many ways to triangulate a graph, for the ease of grading, please use the triangulation that corresponds to the elimination order A, B, C, D, E.)

Solution

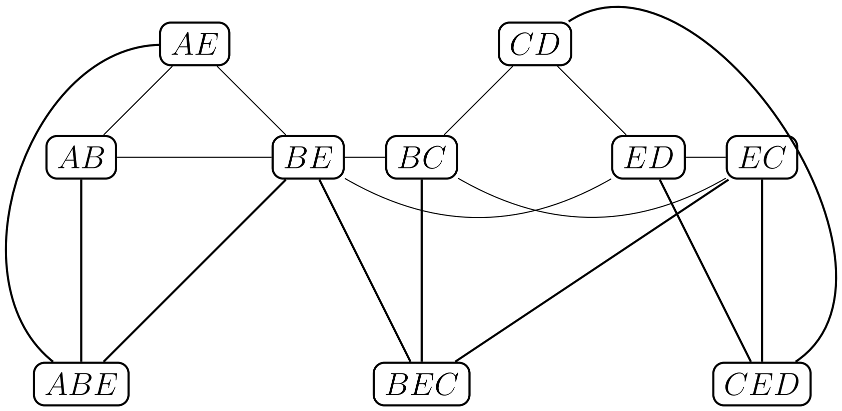
Here is the **triangulated graph** $\mathcal{H}^*$:



3. [**3 pts**] Construct a cluster graph $\mathcal{U}$ where each node is a maximal clique $\boldsymbol{C}_i$ from $\mathcal{H}^*$ and each edge is the sepset $\boldsymbol{S}_{i,j} = \boldsymbol{C}_i \cap \boldsymbol{C}_j$ between adjacent cliques $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$.

Solution

here is the **cliqure tree** obtained from the *chordal* graph $\mathcal{H}^*$.

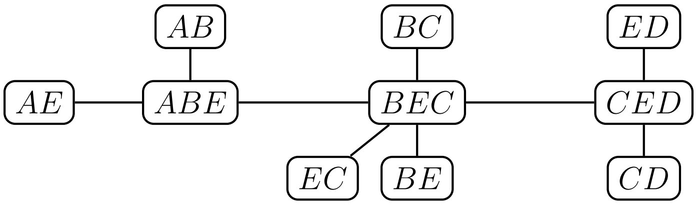


4. [**3 pts**] Run maximum spanning tree algorithm on $\mathcal{U}$ to construct a junction tree $\mathcal{T}$.

(The cluster graph is small enough to calculate maximum spanning tree in one's head.)

Solution

Here is the maximum junction tree:



3.3 Junction tree in action: part 2 [20 points]

Continuing from part 1, now assume all variables are binary and the CPDs are parameterized as follows:

We are going to implement belief propagation on $\mathcal{T}$. The provided template `junction_tree.py` contains the following tasks:

- `initial_clique_potentials()`: Compute initial clique potentials $\psi_i(\boldsymbol{C}_i)$ from factors ϕ_i.

A	$P(A)$
0	x_0

A	B	$P(B\|A)$
0	0	x_1
1	0	x_2

A	E	$P(E\|A)$
0	0	x_3
1	0	x_4

B	C	$P(C\|B)$
0	0	x_5
1	0	x_6

C	E	D	$P(D\|C,E)$
0	0	0	x_7
0	1	0	x_8
1	0	0	x_9
1	1	0	x_{10}

- `messages()`: Compute messages $\delta_{i\to j}$ from initial clique potentials $\psi_i(\boldsymbol{C}_i)$.
- `beliefs()`: Compute calibrated clique beliefs $\beta_i(\boldsymbol{C}_i)$ and sepset beliefs $\mu_{i,j}(\boldsymbol{S}_{i,j})$, using initial clique potentials $\psi_i(\boldsymbol{C}_i)$ and messages $\delta_{i\to j}$.
- Using the beliefs $\beta_i(\boldsymbol{C}_i), \mu_{i,j}(\boldsymbol{S}_{i,j})$, compute
  - `query1()`: $P(B)$
  - `query2()`: $P(A|C)$
  - `query3()`: $P(A, B, C, D, E)$

Please finish the unimplemented TODO blocks and submit completed `junction_tree.py` to Gradescope (`https://www.gradescope.com/courses/36025`).

In the implementation, please represent factors as `numpy.ndarray` and store different factors in a dictionary with its scope as the key. For example, as provided in the template, `phi['ab']` is a factor ϕ_{AB} represented as a 2×2 matrix, where `phi['ab'][0, 0]` $= \phi_{AB}(A = 0, B = 0) = P(B = 0|A = 0) = x_1$. For messages, one can use `delta['ab_cd']` to denote a message from AB to CD. Most functions can be written in 3 lines of code. You may find `np.einsum()` useful.

4 Parameter Learning [15 points] (Xun)

$$\begin{tikzcd}
Y_1 \arrow[r] \arrow[d] & Y_2 \arrow[r] \arrow[d] & \cdots \arrow[r] & Y_T \arrow[d] \\
X_1 & X_2 & & X_T
\end{tikzcd}$$

Consider an HMM with $Y_t \in [M]$, $X_t \in \mathbb{R}^K$ $(M, K \in \mathbb{N})$. Let $(\pi, A, \{\mu_i, \sigma_i^2\}_{i=1}^M)$ be its parameters, where $\pi \in \mathbb{R}^M$ is the initial state distribution, $A \in \mathbb{R}^{M \times M}$ is the transition matrix, $\mu_i \in \mathbb{R}^K$ and $\sigma_i^2 > 0$ are parameters of the emission distribution, which is defined to be an isotropic Gaussian. In other words,

$$P(Y_1 = i) = \pi_i \tag{6}$$

$$P(Y_{t+1} = j | Y_t = i) = A_{ij} \tag{7}$$

$$P(X_t | Y_t = i) = \mathcal{N}(X_t; \mu_i, \sigma_i^2 I). \tag{8}$$

We are going to implement the Baum-Welch (EM) algorithm that estimates parameters from data $\boldsymbol{X} \in \mathbb{R}^{N \times T \times K}$, which is a collection of N observed sequences of length T. Note that there are different forms of forward-backward algorithms, for instance the (α, γ)-recursion, which is slightly different from the (α, β)-recursion we saw in the class. For the ease of grading, however, please implement the (α, β) version, and remember to normalize the messages at each step for numerical stability.

Please complete the unimplemented TODO blocks in the template `baum_welch.py` and submit it to Gradescope (`https://www.gradescope.com/courses/36025`). The template has its own toy problem to verify the implementation. The test cases are ran on other randomly generated problem instances.

References

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.