CS 475 Machine Learning: Homework 4 Analytical (70 points)

Assigned: Monday, Nov. 1st, 2021

Due: Monday, Nov. 15th, 2021, 11:59 pm US/Eastern

Partner 1: NAME (JHED), Partner 2: NAME (JHED)

Instructions

We have provided this LaTeX document for turning in this homework. We give you one or more boxes to answer each question. The question to answer for each box will be noted in the title of the box. You can change the size of the box if you need more space.

Other than your name, do not type anything outside the boxes. Leave the rest of the document unchanged.

Do not add text outside of the answer boxes. You are allowed to make boxes larger if needed.

We strongly recommend you review your answers in the generated PDF to ensure they appear correct. We will grade what appears in the answer boxes in the submitted PDF, NOT the original latex file.

MRFs

Question 1.

Consider the graphical model shown in Figure 1. In this model, $\mathbf{x}$ is a sequence of observations for which we want to output a prediction $\mathbf{y}$, which itself is a sequence, where the size of $\mathbf{y}$ is the same as $\mathbf{x}$. Assume that the potential functions have a log-linear form: $\psi(Z) = \exp\{\sum_i \theta_i f_i(Z)\}$, where Z is the set of nodes that are arguments to the potential function (i.e. some combination of nodes in $\mathbf{x}$ and $\mathbf{y}$,) θ are the parameters of the potential functions and f_i is a feature function.

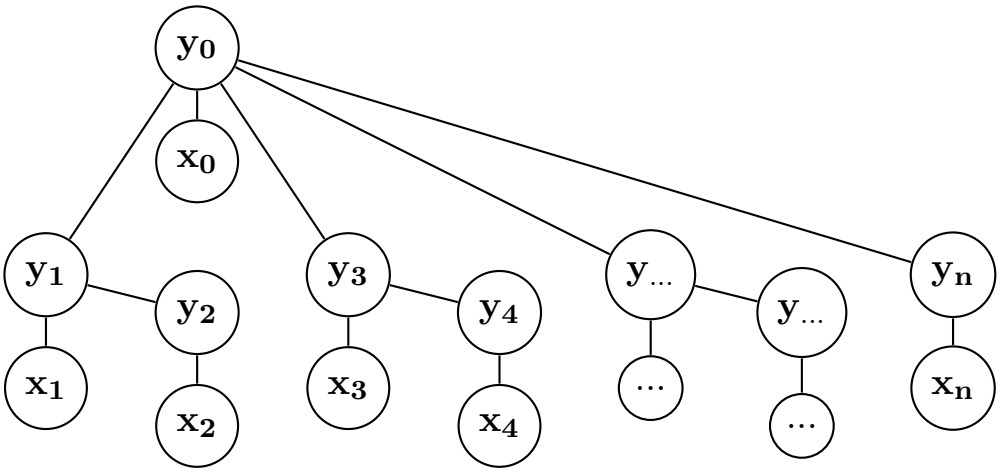


Figure 1: Tree structure model

(a) Write the log likelihood for this model of a single instance $\mathbf{x}$: $\log p(\mathbf{y}, \mathbf{x})$.

(b) Write the conditional log likelihood for this model of a single instance $\mathbf{x}$: $\log p(\mathbf{y}|\mathbf{x})$.

(c) Assume that each variable y_i can take one of k possible states, and variable x_i can take one of k' possible states, where k' is very large. Describe the computational challenges of modeling $\log p(\mathbf{y}, \mathbf{x})$ vs $\log p(\mathbf{y}|\mathbf{x})$.

Question 2.

(a) Suppose you wanted to compute $S = \sum_{x_1=1}^{100} \cdots \sum_{x_8=1}^{100} h(x)$ where

$$h(x) = \exp(x_1x_2 + x_4x_5 + x_7x_8) \prod_{i=2,5,7} (x_i + x_3 + x_6)^i.$$

It looks like the sum has $100^8 = 10^{16}$ terms, so it seems we must evaluate h 10^{16} times. Explain (precisely) how you can compute S with at most 10^7 evaluations of h or something simpler than h.

(b) Draw the MRF associated with this distribution.

(a) The summation is:

$$\begin{aligned} S &= \sum_{x_1=1}^{100} \cdots \sum_{x_8=1}^{100} \exp(x_1x_2 + x_4x_5 + x_7x_8) \prod_{i=2,5,7} (x_i + x_3 + x_6)^i \\ &= \sum_{x_1=1}^{100} \cdots \sum_{x_8=1}^{100} e^{x_1x_2} e^{x_4x_5} e^{x_7x_8} \prod_{i=2,5,7} (x_i + x_3 + x_6)^i \\ &= \sum_{x_2=1}^{100} \cdots \sum_{x_8=1}^{100} \left(\sum_{x_1=1}^{100} e^{x_1x_2} \right) e^{x_4x_5} e^{x_7x_8} \prod_{i=2,5,7} (x_i + x_3 + x_6)^i \end{aligned}$$

Now we can compute the summation $\left(\sum_{x_1=1}^{100} e^{x_1x_2}\right)$ easily as:

$$\phi^*_{x_2}(x_2) = \sum_{x_1=1}^{100} e^{x_1x_2}$$

This is an object that is only a function of x_2. Thus, we can pre-evaluate it and store it as a mapping function (distribution) of x_2. Only need 100 spaces to store this mapping function, and as only two variables are involved, the computation of this mapping function would need to evaluate $e^{x_1x_2}$ for $100^2 = 10^4$ times, so the total number of evaluations now is 10^4
Then we can apply the same thing again to sum x_4:

$$S = \sum_{x_2=1}^{100} \sum_{x_3=1}^{100} \sum_{x_5=1}^{100} \cdots \sum_{x_8=1}^{100} \phi^*_{x_2}(x_2) \left(\sum_{x_4=1}^{100} e^{x_4x_5} \right) e^{x_7x_8} \prod_{i=2,5,7} (x_i + x_3 + x_6)^i$$

Again we compute the summation $\left(\sum_{x_4=1}^{100} e^{x_4x_5}\right)$ easily as a mapping function (distribution) of x_5:

$$\phi^*_{x_5}(x_5) = \sum_{x_4=1}^{100} e^{x_4x_5}$$

We pre-evaluate this function and store it as a mapping function. Again, it requires 100 spaces and we need to evaluate $e^{x_4x_5}$ for $100^2 = 10^4$ times, so the total number of evaluations now is $10^4 + 10^4 = 2 \cdot 10^4$
We continue to apply this trick to sum x_8:

$$S = \sum_{x_2=1}^{100} \sum_{x_3=1}^{100} \sum_{x_5=1}^{100} \cdots \sum_{x_7=1}^{100} \phi^*_{x_2}(x_2)\phi^*_{x_5}(x_5) \left(\sum_{x_8=1}^{100} e^{x_7x_8} \right) \prod_{i=2,5,7} (x_i + x_3 + x_6)^i$$

Compute summation $\left(\sum_{x_8=1}^{100} e^{x_7x_8}\right)$ easily as mapping function (distribution) of x_7:

$$\phi^*_{x_7}(x_7) = \sum_{x_8=1}^{100} e^{x_7x_8}$$

Again we pre-evaluate this function and store it as a mapping function, which requires 100 spaces and we need to evaluate $e^{x_7x_8}$ for $100^2 = 10^4$ times, so the total number of evaluations now is $2 \cdot 10^4 + 10^4 = 3 \cdot 10^4$

(Continued)
Now further expand our equation we have:

$$S = \sum_{x_2=1}^{100}\sum_{x_3=1}^{100}\sum_{x_5=1}^{100}\cdots\sum_{x_7=1}^{100}\phi^*_{x_2}(x_2)\phi^*_{x_5}(x_5)\phi^*_{x_7}(x_7)(x_2+x_3+x_6)^2(x_5+x_3+x_6)^5(x_7+x_3+x_6)^7$$

This time we can sum x_2 first:

$$S = \sum_{x_3=1}^{100}\sum_{x_5=1}^{100}\cdots\sum_{x_7=1}^{100}\left(\sum_{x_2=1}^{100}\phi^*_{x_2}(x_2)(x_2+x_3+x_6)^2\right)\phi^*_{x_5}(x_5)\phi^*_{x_7}(x_7)\prod_{i=5,7}(x_i+x_3+x_6)^i$$

We can compute the summation $\left(\sum_{x_2=1}^{100}\phi^*_{x_2}(x_2)(x_2+x_3+x_6)^2\right)$ easily as mapping function (distribution) of x_3 and x_6:

$$\phi^{x_2\rightarrow}_{x_3,x_6}(x_3,x_6) = \sum_{x_2=1}^{100}\phi^*_{x_2}(x_2)(x_2+x_3+x_6)^2$$

This is an object that is only a function of x_3 and x_6. Thus, we can pre-evaluate it and store it as a mapping function (distribution) of x_3 and x_6. This time we need 10^4 spaces to store this mapping function, and as only three variables are involved, the computation of this mapping function would need to evaluate $\phi^*_{x_2}(x_2)(x_2+x_3+x_6)^2$ for $100^3 = 10^6$ times, so the total number of evaluations now is $3\cdot 10^4 + 10^6$
We apply the same trick to sum x_5:

$$S = \sum_{x_3=1}^{100}\sum_{x_6=1}^{100}\sum_{x_7=1}^{100}\phi^{x_2\rightarrow}_{x_3,x_6}(x_3,x_6)\left(\sum_{x_5=1}^{100}\phi^*_{x_5}(x_5)(x_5+x_3+x_6)^5\right)\phi^*_{x_7}(x_7)(x_7+x_3+x_6)^7$$

Similarly we compute the summation $\left(\sum_{x_5=1}^{100}\phi^*_{x_5}(x_5)(x_5+x_3+x_6)^5\right)$ as a mapping function (distribution) of x_3 and x_6:

$$\phi^{x_5\rightarrow}_{x_3,x_6}(x_3,x_6) = \sum_{x_5=1}^{100}\phi^*_{x_5}(x_5)(x_5+x_3+x_6)^5$$

Again we pre-evaluate this function and store it as a mapping function, which requires 10^4 spaces and we need to evaluate $\phi^*_{x_5}(x_5)(x_5+x_3+x_6)^5$ for $100^3 = 10^6$ times, so the total number of evaluations now is $3\cdot 10^4 + 10^6 + 10^6 = 2\cdot 10^6 + 3\cdot 10^4$
Next we sum x_7

$$S = \sum_{x_3=1}^{100}\sum_{x_6=1}^{100}\phi^{x_2\rightarrow}_{x_3,x_6}(x_3,x_6)\phi^{x_5\rightarrow}_{x_3,x_6}(x_3,x_6)\left(\sum_{x_7=1}^{100}\phi^*_{x_7}(x_7)(x_7+x_3+x_6)^7\right)$$

We compute summation $\left(\sum_{x_7=1}^{100}\phi^*_{x_7}(x_7)(x_7+x_3+x_6)^7\right)$ easily as mapping function (distribution) of x_3 and x_6:

$$\phi^{x_7\rightarrow}_{x_3,x_6}(x_3,x_6) = \sum_{x_7=1}^{100}\phi^*_{x_7}(x_7)(x_7+x_3+x_6)^7$$

(Continued)

Once more we pre-evaluate this function and store it as a mapping function, which requires 10^4 spaces and we need to evaluate $\phi^*_{x_7}(x_7)(x_7+x_3+x_6)^7$ for $100^3 = 10^6$ times, so the total number of evaluations now is $2\cdot 10^6 + 3\cdot 10^4 + 10^6 = 3\cdot 10^6 + 3\cdot 10^4$ The final expression we need to evaluate looks like:

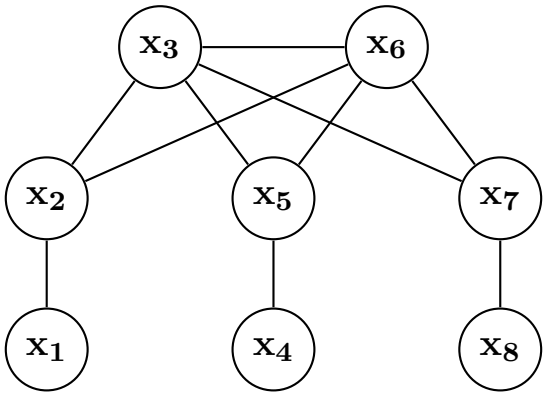
$$S = \sum_{x_3=1}^{100}\sum_{x_6=1}^{100} \phi^{x_2\rightarrow}_{x_3,x_6}(x_3,x_6)\phi^{x_5\rightarrow}_{x_3,x_6}(x_3,x_6)\phi^{x_7\rightarrow}_{x_3,x_6}(x_3,x_6)$$

All of $\phi^{x_2\rightarrow}_{x_3,x_6}(x_3,x_6)$, $\phi^{x_5\rightarrow}_{x_3,x_6}(x_3,x_6)$ and $\phi^{x_7\rightarrow}_{x_3,x_6}(x_3,x_6)$ are mapping functions that we already computed and stored, so the evaluation of S now is fairly easy. We only need to sum over two variables x_3 and x_6, so the number of evaluation is $100^2 = 10^4$. The total time of evaluations now is: $3\cdot 10^6 + 3\cdot 10^4 + 10^4 = 3\cdot 10^6 + 4\cdot 10^4$. We can see clearly that:

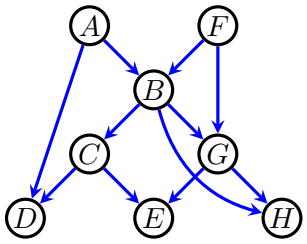
$$3\cdot 10^6 + 4\cdot 10^4 < 10^7$$

So our total number of evaluations is less than 10^7, and the method described above meets the requirement of this question.

(b) from part (a) we know from the equation and computation that we have 6 cliques in total: $(x_1,x_2),(x_4,x_5),(x_7,x_8),(x_3,x_6,x_2),(x_3,x_6,x_5),(x_3,x_6,x_7)$. We can draw a MRF that contains all of them:



DAGs, Clique Trees and Message Passing.



In a statistical DAG model for the graph shown, let $\mathbf{V} = \{A, B, C, D, E, F, G, H\}$.

(a) Answer (and explain your answer) the following d-separation queries:

$A \perp\!\!\!\perp F \mid D$
$A \perp\!\!\!\perp G \mid B, C$
$G \perp\!\!\!\perp A \mid B, H, D, E, F$
$F \perp\!\!\!\perp D \mid A, B$
$C \perp\!\!\!\perp H \mid B$

(b) Write down the local Markov property of this model.

(c) Consider a new graph where we reverse the direction of the edge $B \to G$ to point the other way: $B \leftarrow G$ (and leave the other edges the same). Does the new graph represent the same model as the old?

Hint: write down the local Markov property for the new graph, and see if all statements in it are implied by d-separation in the original graph. In general, if local Markov of $\mathcal{G}_1$ is implied by global Markov of $\mathcal{G}_2$, and local Markov of $\mathcal{G}_2$ is implied by global Markov of $\mathcal{G}_1$, then $\mathcal{G}_1$ and $\mathcal{G}_2$ represent the same model. Otherwise they do not.

(d) A moralized graph $\mathcal{G}^a$ is obtained from a DAG $\mathcal{G}$ by connecting all non-adjacent variables V_i and V_j such that $V_i \to V_k \leftarrow V_j$ is in the graph (for some V_k), and replacing all directed edges by undirected edges. What is the moralized graph for the DAG in this problem?

(e) Write down the MRF factorization of the moralized graph $\mathcal{G}^a$.

(f) Is this graph chordal? If not, add a set of edges to make it chordal. If you added edges, write the factorization of the new graph.

(g) Create a clique tree from the triangulated graph (either $\mathcal{G}^a$ or the graph obtained from $\mathcal{G}^a$ by adding new edge(s)).

(h) Pick a root $\mathbf{R}$ of the clique tree, and calculate both incoming messages $\phi^{\mathbf{S}_i \to \mathbf{S}_j}$ from each $\mathbf{S}_i$ towards its neighbor $\mathbf{S}_j$ closer to the root, and outgoing messages $\phi^{\mathbf{S}_k \leftarrow \mathbf{S}_i}$ from $\mathbf{S}_i$ to each neighbor $\mathbf{S}_k$ further than $\mathbf{S}_i$ from the root, in terms of clique potentials and other messages.

(i) By substituting in the clique factors in each message, show that in this example, for each leaf node $\mathbf{S}_i$ with a neighbor node $\mathbf{S}_j$,

$$p(\mathbf{S}_i) = \frac{\phi^{\mathbf{S}_i \leftarrow \mathbf{S}_j}_{\mathbf{S}_j \setminus \mathbf{S}_i} \phi_{\mathbf{S}_i}}{\sum_{\mathbf{S}_i} \phi^{\mathbf{S}_i \leftarrow \mathbf{S}_j}_{\mathbf{S}_j \setminus \mathbf{S}_i} \phi_{\mathbf{S}_i}} = \frac{\sum_{\mathbf{V} \setminus \mathbf{S}_i} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}{\sum_{\mathbf{V}} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}$$

for each non-leaf note $\mathbf{S}_i$ with a neighbor $\mathbf{S}_j$ closer to the root, and neighbors $\mathbf{S}_1, \ldots, \mathbf{S}_m$ further from the root that

$$p(\mathbf{S}_i) = \frac{\phi_{\mathbf{S}_j \setminus \mathbf{S}_i}^{\mathbf{S}_i \leftarrow \mathbf{S}_j} \left(\prod_{k=1}^{m} \phi_{\mathbf{S}_k \setminus \mathbf{S}_i}^{\mathbf{S}_k \rightarrow \mathbf{S}_i} \right) \phi_{\mathbf{S}_i}}{\sum_{\mathbf{S}_i} \phi_{\mathbf{S}_j \setminus \mathbf{S}_i}^{\mathbf{S}_i \leftarrow \mathbf{S}_j} \left(\prod_{k=1}^{m} \phi_{\mathbf{S}_k \setminus \mathbf{S}_i}^{\mathbf{S}_k \rightarrow \mathbf{S}_i} \right) \phi_{\mathbf{S}_i}} = \frac{\sum_{\mathbf{V} \setminus \mathbf{S}_i} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}{\sum_{\mathbf{V}} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}$$

and finally for the root node $\mathbf{S}_i$ with neighbors $\mathbf{S}_1, \ldots, \mathbf{S}_m$ that

$$p(\mathbf{S}_i) = \frac{\left(\prod_{k=1}^{m} \phi_{\mathbf{S}_k \setminus \mathbf{S}_i}^{\mathbf{S}_k \rightarrow \mathbf{S}_i} \right) \phi_{\mathbf{S}_i}}{\sum_{\mathbf{S}_i} \left(\prod_{k=1}^{m} \phi_{\mathbf{S}_k \setminus \mathbf{S}_i}^{\mathbf{S}_k \rightarrow \mathbf{S}_i} \right) \phi_{\mathbf{S}_i}} = \frac{\sum_{\mathbf{V} \setminus \mathbf{S}_i} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}{\sum_{\mathbf{V}} \prod_{C \in \mathcal{C}(\mathcal{G})} \phi_C}$$

Here $\mathbf{V}$ is all variables in the graph, and $\mathcal{C}(\mathcal{G})$ is the set of maximal cliques in the graph.

K-Means

(a) Is it possible to initialize the k-means algorithm in such a way that it fails to terminate successfully?

(b) Say our input to k-means is a set of $2k$ points with 2-coordinates arranged in line, e.g. with coordinates:

$$(0,0),(0,1),(0,2),\ldots,(0,k),(0,k+1),\ldots,(0,2k).$$

Say we initialize k-means with 2 clusters, with initial centroids given by $(0,k)$ and $(0,k+1)$. In many iterations will k-means terminate? What will be the final cluster assignments and centroids?

(a) No, it is not possible. By definition, k-means will always converge. In every iteration of k-means, the sum-of-distances to the center is reduced, and eventually it will stop changing, and the algorithm just terminate. Based on this definition, in each iteration the loss of k-mean will always decrease, or stay the same. The loss will never increase back, so the algorithm will always converge at some point of lower loss.

(b) It will terminate after just 2 iterations (the change only happens in the 1st iteration, and in the 2nd iteration nothing changes, so it terminates).
We can work out this step by step.
It is obvious that in the first step, (0,0),(0,1),...,(0,k) are closer to centroid (0, k), and (0,k+1),(0,k+2),...,(0,2k) are closer to centroid (0, k+1) Thus, as x value is always 0, the new y value of the first centroid calculated from (0,0),(0,1)...(0,k) will be:

$$y_1 = \frac{0+1+2+...+k}{k+1} = \frac{(0+k)*(k+1)/2}{k+1} = \frac{(0+k)}{2} = \frac{k}{2}$$

Simiarly, the new value of y value of the second centroid calculated from (0,k+1),(0,k+2)...(0,2k) will be:

$$y_2 = \frac{(k+1)+(k+2)+...+(2k)}{k} = \frac{(k+1+2k)*(k)/2}{k} = \frac{(3k+1)}{2} = \frac{3k}{2}+\frac{1}{2}$$

Now the new centroids are $(0, \frac{k}{2})$ and $(0, \frac{3k}{2}+\frac{1}{2})$.
Then in second iteration, it is obvious that most points will remain where the are, and we only need to check the edge cases (0,k) and (0,k+1). First we calculate the distance of (0,k) to both centroids:

$$d_1 = |k - \frac{k}{2}| = \frac{k}{2} \qquad d_2 = |\frac{3k}{2}+\frac{1}{2}-k| = \frac{k}{2}+\frac{1}{2}$$

We can see clearly that $d_1 < d_2$, so (0,k) is still closer to the first centroid and will remain in cluster 1.
Then we calculate the distance of (0,k+1) to both centroids:

$$d_1 = |k+1-\frac{k}{2}| = \frac{k}{2}+1 \qquad d_2 = |\frac{3k}{2}+\frac{1}{2}-k-1| = \frac{k}{2}-\frac{1}{2}$$

We can see clearly that $d_1 > d_2$, so (0,k+1) is still closer to the second centroid and will remain in cluster 2.

(continued)
Thus, we can see that in iteration 2 nothing changes, so the algorithm will terminate. The number of iteration is 2 and the final cluster assignments and centroids are:

$$\text{Cluster 1: centroid: } \frac{k}{2} \text{ Points: } (0,0),(0,1),...,(0,k)$$

$$\text{Cluster 2: centroid: } \frac{3k}{2}+\frac{1}{2} \text{ Points: } (0,k+1),(0,k+2),...,(0,2k)$$