CS 475-675 Machine Learning: Midterm 2

Fall 2021

150 points.

Name (print): Dimitri Lezcano

JHED: dlezcan1

If you think a question is unclear or multiple answers are reasonable, please write a brief explanation of your answer, to be safe. Also, show your work if you want wrong answers to have a chance at some credit: it lets us see how much you understood.

This exam is open-book: permitted materials include textbooks, personal notes, lecture material, recitation material, past assignments, the course Piazza, and scholarly articles and papers. Other materials are otherwise not permitted. It is not permitted to discuss or share questions or solutions of this exam with any person, via any form of communication, other than the course instructors. It is not permitted to solicit or use any solutions to past exams for this course.

Declaration:

I have neither given nor received any unauthorized aid on this exam. In particular, I have not spoken to any other student about any part of this exam. The work contained herein is wholly my own. I understand that violation of these rules, including using an unauthorized aid, copying from another person, or discussing this exam with another person in any way, may result in my receiving a 0 on this exam.

Signature Date

Good luck!

True/False (50 points)

For each question, circle (or otherwise clearly indicate) either True or False. Regardless of which answer you chose, explain why.
2 points for correct True/False answer, -2 points for incorrect True/False answer, 3 points for a correct explanation, 0 points for an incorrect explanation.

1) Computing single variable marginals in a DAG model is always possible by treating it as a Markov random field model.
True False
Explanation:
False

Consider $A \to B \to C$. If we were to treat the model as an MRF, we would get that $A \perp\!\!\!\perp C$, however, this does not hold for the DAG model, therefore, the marginal would not be the same since the distributions are different.

2) A DAG model is always observationally equivalent to a Markov random field model.
True False
Explanation:
False

Consider $A \to B \leftarrow C$. Here, in the DAG, we have that $A \perp\!\!\!\perp C$ and $A \not\perp\!\!\!\perp C \mid B$. $A \perp\!\!\!\perp C$ means that A and C cannot be adjacent in the MRF. We also have that $A \not\perp\!\!\!\perp B$ and $B \not\perp\!\!\!\perp C$, therefore, we must have an edge connecting $A - B$ and $B - C$ in the MRF. However, given this model, we have that $A \perp\!\!\!\perp C \mid B$. Therefore, the distribution is not equivalent. The only other edge we could add to "fix" the MRF to match the DAG is an edge between $A - C$, however, this would violate $A \perp\!\!\!\perp C$ in the MRF, therefore this DAG cannot be modeled as an MRF exactly without loosening assumptions and therefore cannot be observationally equivalent to an MRF.

3) Consider the following predictor trained from a dataset $[D]$ with features $\vec{X}$ and outcomes Y. First, a k-means clustering algorithm is trained using k clusters, and treatment both $\vec{X}$ and Y as "features" when computing the centroid distance. Second, when a new data point $\vec{X}_{\text{new}}$ is to be classifier, the predictor finds the closest of the k centroids, and takes the majority vote among outcomes Y in all points in that centroid. This predictor is an example of a k-nearest neighbor algorithm.
True False
Explanation:
False

Consider the data ($X \in \mathbb{R}, Y$ binary): $\{(-1, 0), (-1/2, 0), (3/4, 1), (7/8, 1)(1, 1)\}$ with the clusters $(-3/4, 0), (7/8, 1)$. Then when looking at the new point $X_{new} = 0$, the closest three points would be $(-1/2, 0), (3/4, 1), (7/8, 1)$, however, the closest cluster is $(-3/4, 0)$, which would then perform voting on $(-1, 0)$ and $(-1/2, 0)$, not including a closer point $(3/4, 1)$.

4) The GES Algorithm will always run in polynomial time in the size of the graph k and the size of the data n.
True False
Explanation:
True

The worst case that the algorithm can run is $\binom{k}{2}$, addition or removal of edges when performing GES. This is also stated (using Selective GES) in the introduction of Chickering, M. (2020). Statistically Efficient Greedy Equivalence Search. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, in *Proceedings of Machine Learning Research* 124:241-249.Available from https://proceedings.mlr.press/v124/chickering20a.html.

5) Two different undirected graphs (on the same set of vertices) correspond to different Markov random field models.
True False
Explanation:
True

Suppose that $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ are different undirected graphs. Then (WLOG) $\exists$ at least a single pair $v_1, v_2 \in V$ s.t. $(v_1, v_2) \in E_1 \cap E_2'$. Therefore, since the graphs are MRFs, we can write the distributions of each of these graphs as

$$p_{\mathcal{G}_i}(\vec{v}) = \frac{1}{Z} \prod_{v \in V} \phi(v) \prod_{(v_j, v_k) \in E_i} \phi(v_j, v_k).$$

Note that $p_{\mathcal{G}_1} \neq p_{\mathcal{G}_2} \iff \exists p_{\mathcal{G}_1}(\vec{v}) / p_{\mathcal{G}_1}(\vec{v}) \neq 1$. So consider $p_{\mathcal{G}_1}(\vec{v}) / p_{\mathcal{G}_2}(\vec{v})$

$$\begin{aligned}
\frac{p_{\mathcal{G}_1}(\vec{v})}{p_{\mathcal{G}_2}(\vec{v})} &= \frac{\frac{1}{Z} \prod_{v \in V} \phi(v) \prod_{(v_j, v_k) \in E_1} \phi(v_j, v_k)}{\frac{1}{Z} \prod_{v \in V} \phi(v) \prod_{(v_j, v_k) \in E_2} \phi(v_j, v_k)} \\
&= \frac{\prod_{(v_j, v_k) \in E_1} \phi(v_j, v_k)}{\prod_{(v_j, v_k) \in E_2} \phi(v_j, v_k)} \\
&= \frac{\prod_{(v_j, v_k) \in E_1 \cap E_2'} \phi(v_j, v_k)}{\prod_{(v_j, v_k) \in E_2 \cap E_1'} \phi(v_j, v_k)} \text{ which is at least} \\
&= \phi(v_1, v_2) \frac{\prod_{(v_j, v_k) \in E_1 \cap E_2' \setminus (v_1, v_2)} \phi(v_j, v_k)}{\prod_{(v_j, v_k) \in E_2 \cap E_1' \setminus (v_1, v_2)} \phi(v_j, v_k)} \neq 1 \because \phi(v_1, v_2) \neq 1 \text{ generally}
\end{aligned}$$

Therefore, two different MRFs results in distributions that are not equivalent. Furthermore, we have that in $\mathcal{G}_1$, ($v_1 \perp\!\!\!\perp$ non-neighbors $\mid v_2$ and other neighbors), but in $\mathcal{G}_2$, ($v_1 \perp\!\!\!\perp$ non-neighbors $\mid$ neighbors that don't include v_2). This results in a different independence which yields a different distribution.

6) $\mathbb{E}[Y^{(1)} \mid \vec{X}]$ is a function of the observed data distribution $p(Y, X, R_Y)$ if Y is MAR given $\vec{X}$.
True False
Explanation:
True

$$\begin{aligned}
p(Y^{(1)} \mid \vec{X}) &= p(Y^{(1)} \mid \vec{X}, R_Y = 1) \because Y^{(1)} \perp\!\!\!\perp R_Y \mid \vec{X} \\
&= p(Y \mid \vec{X}, R_Y = 1) \because R_Y = 1 \implies Y = Y^{(1)}
\end{aligned}$$

Therefore, $p(Y^{(1)} \mid \vec{X})$ is a function of $p(Y \mid \vec{X}, R_Y) \implies \mathbb{E}\left[Y^{(1)} \mid \vec{X}\right]$ is a function of $p(Y \mid \vec{X}, R_Y)$.

7) LDA can be used as a clustering algorithm.
True False
Explanation:
True

LDA defines a method for grouping data points into groups that best define the output. Therefore, it can be considered as a clustering algorithm, using the features and the labels of the dataset, where the clusters are determined to be discriminated by the label of the dataset.

8) Value iteration will always converge to the true *value function* in a finite number of steps.
True False
Explanation:
False

Value iteration will always converge to the true value function in an infinite number of steps. We are not guaranteed convergence of a finite number of steps.

9) The Newton-Raphson algorithm is a special case of gradient descent.
True False
Explanation:
True

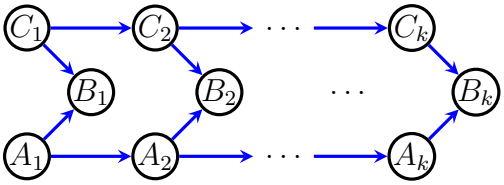
The NR algorithm uses the gradient to determine the optimal update direction and the Hessian to determine the size of the update size.

10) Super learner will do better than any single predictor in its library, as sample size goes to infinity.
True False
Explanation:
True

As per slide 8 of the Ensemble Methods lecture, it is stated that "[one] can show Super Learner does *as well as the best possibly weighted combination in* [the hypothesis class] as $N \to \infty$. Should we contain the full distribution, obtained by an infinite sample size, take any single learner to be the best performing learner, a Super Learner could learn the weighting that that only weights the best performing learner with $\alpha = 1$ and the rest 0. Therefore, any other improvements to learning by varying α parameters are lower-bounded by the performance of the best performing single-learner.

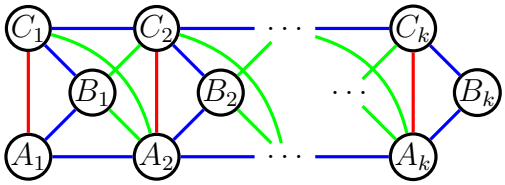
Multiple Part Questions (100 points).

11) **Message Passing (25 points)** Consider the DAG below.



(i) What is the moralized and triangulated undirected graph corresponding to this DAG? Since the DAG has repeating structure, feel free to only draw the first few "slices" of the graph.

Solution:

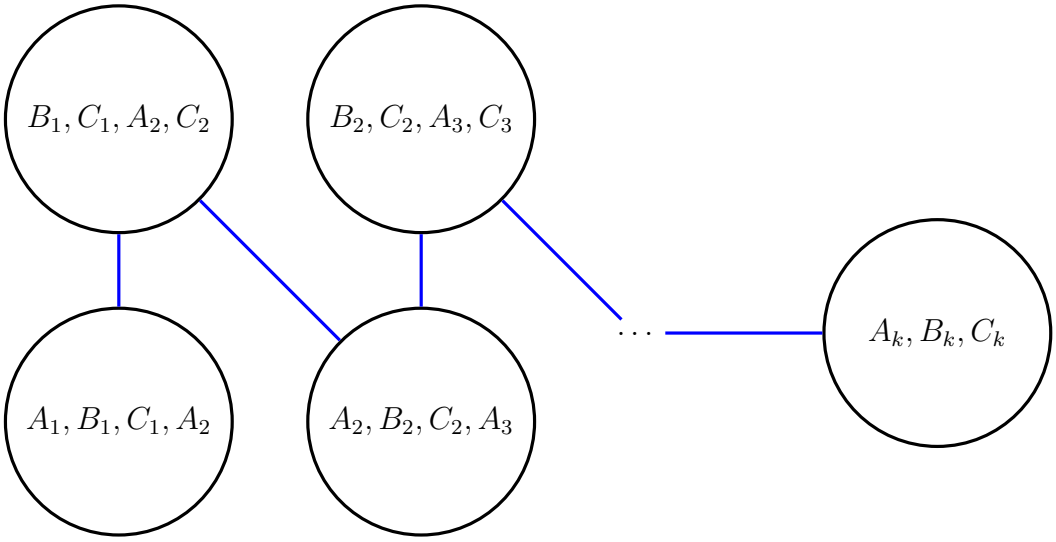


Cliques:

- $(A_i, B_i, C_i, A_{i+1}), \forall i = 1, \ldots, k-1$
- $(B_i, C_i, A_{i+1}, C_{i+1}), \forall i = 1, \ldots, k-1$
- (A_k, B_k, C_k)

(ii) Construct a clique tree from the graph in (i). Again, since the clique tree has repeating structure, feel free to only draw enough of it so it's clear what the repeated structure is.

Solution:



(iii) Do you expect message passing applied to the graph in (i) to run in polynomial time in k? Explain.

Solution:

Given that the max size of the clique is 4, which is constant over the graph. Therefore, we have that the message passing should run in polynomial time in k.

12) **Reinforcement Learning (25 points)**

Consider a reinforcement learning problem with states s_0, s_1, and two actions a_0, a_1. We know the following:

$$p(s^{(t+1)} = s_1 \mid s^{(t)} = s_0, a_0) = 0.1; \;\; p(s^{(t+1)} = s_1 \mid s^{(t)} = s_0, a_1) = 0.7$$
$$p(s^{(t+1)} = s_0 \mid s^{(t)} = s_1, a_1) = 0.8; \;\; p(s^{(t+1)} = s_0 \mid s^{(t)} = s_1, a_0) = 0.2$$
$$R_{a_0}(s_0, s_1) = 1; \;\; R_{a_1}(s_0, s_1) = -1, R = 0 \text{ for all other state transitions and actions.}$$

and let the discount factor be $\gamma = 0.5$.

(i) What is $V^{(1)}(s_0)$ and $V^{(1)}(s_1)$ (value functions for states s_0, s_1 after a single loop of the value iteration algorithm)? Show your work!

Solution:

$V^{(0)}(s_0) = V^{(0)}(s_1) = 0$. (Multiplications are probability of transition $\times$ reward of transition)

$$Q^{(1)}(s_0, a_0) = 0.1(1) + 0.9(0) = 0.1$$
$$Q^{(1)}(s_0, a_1) = 0.7(-1) + 0.3(0) = -0.7$$
$$V^{(1)}(s_0) = \max(Q^{(1)}(s_0, a_0), Q^{(1)}(s_0, a_1)) = 0.1$$

$$Q^{(1)}(s_1, a_0) = 0.2(0) + 0.8(0) = 0$$
$$Q^{(1)}(s_1, a_1) = 0.8(0) + 0.2(0) = 0$$
$$V^{(1)}(s_1) = \max(Q^{(1)}(s_1, a_0), Q^{(1)}(s_1, a_1)) = 0$$

(ii) Without performing an explicit calculation, what do you think is the optimal policy for this problem? Explain.

Solution:

The optimal policy will be $\pi^* = a_0$, since that policy is the only policy that has positive rewards for both states. Here, we can predict $V^{a_1}(s) < 0 < V^{a_0}(s), \forall s = s_0, s_1$

(iii) Assume that we did not know the transition probabilities in the above Markov chain. Suggest a modification to value iteration that would be able to handle this case, while still converging to the value function at every state.

Solution:

We are able to use Q learning in order to modify value iteration. First, we initialize $\hat{Q}(s,a) = 0, \forall s = s_0, s_1, \forall a = a_0, a_i$. Then, starting at some current state, we iteratively pick some action and update the current state and calculate a reward for the transition that occurs. In the background, we also hold a state counter that calculates the number of times that the state-action pair was visited as we continue to iterate until some convergence. We then update $\hat{Q}$ by

$$\hat{Q}(s,a) \leftarrow (1-\alpha)\hat{Q}(s,a) + \alpha\left(R_a(s,s') + \gamma \max_{\tilde{a}} \hat{Q}(s', \tilde{A})\right)$$

where $\alpha = 1/(1 +$ number of times (s,a) was visited), and s' is the new/updated state. Then, after determining the new $\hat{Q}$, use $\hat{Q}(s,a)$ to update the value function by taking $\hat{V}(s) \leftarrow \max_a \hat{Q}(s,a)$. Since we know by Q learning, $\hat{Q}$ converges to Q which implies that $\hat{V}$ converges to V because of value iteration convergence.

(iv) Assume rewards were not fixed, but drawn from distributions: $R_{a_0}(s_0, s_1) \sim f_1(a_0, s_0, s_1)$ and $R_{a_1}(s_0, s_1) \sim f_2(a_1, s_0, s_1)$. Suggest a modification to value iteration that would be able to handle this case, while still converging to the value function $V^*(s)$ at every state s, where the value function is now defined as

$$V^*(s) = \mathbb{E}[R_{a=\pi^*(s)}(s, s^{(1)})] + \gamma \mathbb{E}[V^*(s^{(1)})],$$

where the expectation is taken both with respect to the state transition probabilities, and f_1 and f_2.

Solution:

Keep a table of rewards for $R(s_0, s_1, a)$ (2 values to update here). As you iterate through value iteration. Then, after action, a, is executed in value iteration, sample the reward and append the sampled reward if a takes you from an initial state $s_0 \rightarrow s_1$ to the action a part of the table. Then, when evaluating the Q function of value iteration, use the mean of the list of rewards $R(s_0, s_1, a)$ when as the Q function. Here, we have in the limit as $t \rightarrow \infty$, mean$\{R_t(s_0, s_1, a)\} \rightarrow \mathbb{E}\left[R_\pi(s)(s, s^{(1)})\right]$ (the actual expected reward.

13) **Prediction With Missing Features (25 points)**

Assume we want to learn parameters β of a regression problem $\mathbb{E}[Y \mid \vec{X}^{(1)}]$ we wish to solve, such that $\vec{X}^{(1)} = \{X_1^{(1)}, X_2^{(1)}\}$ is a pair of real-valued features, where either feature could be missing in our data. Recall that in missing data problems, $X_i = X_i^{(1)}$ if $R_i = 1$, and $X_i =?$ otherwise.

Assume that

$$R_1 \perp\!\!\!\perp R_2, X_1^{(1)} \mid Y, X_2^{(1)}; \;\; R_2 \perp\!\!\!\perp R_1, X_2^{(1)} \mid Y, X_1^{(1)}.$$

(i) Is the model MCAR, MAR or MNAR? Explain.

Solution:

This model is not MCAR, since R_i and $X_i^{(1)}$ both depend on Y for $i = 1, 2$. Considering MAR, due to symmetry of the independence assumptions, showing that X_1 is MAR, this would imply X_2 is MAR, as well as if X_1 is not MAR. So consider $p(R_1 \mid X_1^{(1)}, X_2^{(1)}, Y)$.

$$\begin{aligned} p(R_1 \mid X_1^{(1)}, X_2^{(1)}, Y) &= p(R_1 \mid X_2^{(1)}, Y) \;\; \because R_1 \perp\!\!\!\perp X_1^{(1)} \mid Y, X_2^{(1)} \\ &= p(R_1 \mid X_2^{(1)}, Y, R_2 = 1) \;\; \because R_1 \perp\!\!\!\perp R_2 \mid X_2^{(1)}, Y \\ &= p(R_1 \mid X_2, Y, R_2 = 1) \end{aligned}$$

which is not a MAR distribution, therefore X_1 (and X_2) is not MAR. **Thus, the model is MNAR.**

(ii) Use these assumptions, and the graphoid axioms to show that $p(r_1, r_2 \mid x_1^{(1)}, x_2^{(1)}, y)$ is a function of the observed data distribution $p(y, x_1, x_2, r_1, r_2)$.

Solution:

Weak union over assumption #1 implies

$$R_1 \perp\!\!\!\perp R_2 \mid X_1^{(1)}, X_2^{(1)}, Y$$

So looking at the presented distribution:

$$\begin{aligned} p(r_1, r_2 \mid x_1^{(1)}, x_2^{(1)}, y) &= p(r_1 \mid x_1^{(1)}, x_2^{(1)}, y) p(r_2 \mid x_1^{(1)}, x_2^{(1)}, y) \;\; \because R_1 \perp\!\!\!\perp R_2 \mid X_1^{(1)}, X_2^{(1)}, Y \\ &= p(r_1 \mid x_1, x_2^{(1)}, y) p(r_2 \mid x_1^{(1)}, x_2, y) \;\; \because R_i \perp\!\!\!\perp X_i^{(1)} \mid Y, X_j^{(i)} \end{aligned}$$

$$\begin{aligned} p(r_1 \mid x_1, x_2^{(1)}, y) &= p(r_1 \mid x_1, x_2^{(1)}, y, r_2 = 1) \;\; \because R_2 \perp\!\!\!\perp R_1 \mid X_2^{(1)}, Y \\ &= p(r_1 \mid x_1, x_2, r_2 = 1, y) \;\; \because \text{ def'n of } R_2 \,\&\, X_2^{(1)} \end{aligned}$$

$$\begin{aligned} \implies p(r_1, r_2 \mid x_1^{(1)}, x_2^{(1)}, y) &= p(r_1 \mid x_1, x_2^{(1)}, y) p(r_2 \mid x_1^{(1)}, x_2, y) \\ &= p(r_1 \mid x_1, x_2, r_2 = 1, y) p(r_2 \mid x_1, x_2, r_1 = 1, y) \\ &= f(p(y, x_1, x_2, r_1, r_2)) \quad \blacksquare \end{aligned}$$

(iii) Noting that $p(y, x_1^{(1)}, x_2^{(1)}) = \frac{p(y,x_1,x_2,r_1=1,r_2=1)}{p(r_1=1,r_2=1|x_1^{(1)},x_2^{(1)},y)}$, show that $p(y, x_1^{(1)}, x_2^{(1)})$ is a function of the observed data distribution $p(y, x_1, x_2, r_1, r_2)$.

Solution:

Using (ii),

$$\begin{aligned} p(y, x_1^{(1)}, x_2^{(1)}) &= \frac{p(y, x_1, x_2, r_1 = 1, r_2 = 1)}{p(r_1 = 1, r_2 = 1 \mid x_1^{(1)}, x_2^{(1)}, y)} \\ &= \frac{p(y, x_1, x_2, r_1 = 1, r_2 = 1)}{p(r_1 = 1 \mid x_1, x_2, r_2 = 1, y)p(r_2 = 1 \mid x_1, x_2, r_1 = 1, y)} \quad \because \text{(ii)} \\ &= g(p(y, x_1, x_2, r_1, r_2)) \quad \blacksquare \end{aligned}$$

14) **The Noisy-OR Classifier (25 points)**

(i) Given a set of binary features $\vec{X} = \{X_1, \ldots, X_k\}$ a *noisy-or* model for the outcome Y has the form

$$p(Y = 0 \mid x_1, \ldots, x_k) = p(Y = 0 \mid \tilde{x}_1, \ldots, \tilde{x}_k) \prod_{i=1}^{k} p(\tilde{x}_i \mid x_i),$$

where $p(Y = 0 \mid \tilde{x}_1 = 0, \ldots, \tilde{x}_k = 0) = 1$ ($Y = 0$ with probability 1 if all $\tilde{x}_i$ are zero), and $p(Y = 1 \mid \tilde{x}_1, \ldots, \tilde{x}_k) = 1$ otherwise. Here, every $\tilde{x}_i$ is a hidden variable, and the model is parameterized by probabilities $p(\tilde{x}_i = 0 \mid x_i = 1)$ and $p(\tilde{x}_i = 0 \mid x_i = 0)$, since:

$$p(Y = 0 \mid x_1, \ldots, x_k) = \prod_{i=1}^{k} p(\tilde{x}_i = 0 \mid x_i),$$

$$p(Y = 1 \mid x_1, \ldots, x_k) = 1 - \prod_{i=1}^{k} p(\tilde{x}_i = 0 \mid x_i),$$

Write down the conditional likelihood function for this model.

Solution:

$$\begin{aligned}
\mathcal{L} &= \prod_i p(Y_i \mid \mathbf{x}_i) \\
&= \prod_i p(Y_i = 0 \mid \mathbf{x}_i)^{(1-Y_i)} p(Y_i = 1 \mid \mathbf{x}_i)^{Y_i} \\
&= \prod_i \left(\prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right)^{(1-Y_i)} \left(1 - \prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right)^{Y_i}
\end{aligned}$$

(ii) Can this conditional likelihood be maximized in closed form? If so, explain how. If not, suggest an iterative procedure for maximizing it.

Solution:

Start by taking the log of $\mathcal{L}$.

$$\begin{aligned}
\log \mathcal{L} &= \sum_i (1 - Y_i) \log \left(\prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right) + Y_i \log \left(1 - \prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right) \\
&= \sum_i (1 - Y_i) \sum_{j=1}^{k} \left(\log p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right) + Y_i \log \left(1 - \prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij}) \right)
\end{aligned}$$

Now consider taking a derivative $\frac{\partial \log \mathcal{L}}{\partial p(\tilde{x}_l = 0 \mid x_l)}$.

$$\frac{\partial \log \mathcal{L}}{\partial p(\tilde{x}_l = 0 \mid x_l)} = 0 \iff$$

$$0 = \sum_i \mathbb{I}(x_{il} = x_l) \left((1 - Y_i) \frac{1}{p(\tilde{x}_l = 0 \mid x_l)} - Y_i \frac{\prod_{j \neq l} p(\tilde{x}_{ij} = 0 \mid x_{ij})}{1 - \prod_{j=1}^{k} p(\tilde{x}_{ij} = 0 \mid x_{ij})} \right)$$

$$0 = \sum_i \mathbb{I}(x_{il} = x_l) \left((1 - Y_i) \frac{1}{p(\tilde{x}_l = 0 \mid x_l)} - Y_i \frac{p(Y = 0 \mid \vec{x}_i)}{p(\tilde{x}_l = 0 \mid x_l) p(Y_i = 1 \mid \vec{x}_i)} \right) \iff$$

$$0 = \frac{1}{p(\tilde{x}_l = 0 \mid x_l)} \sum_i \left(1 - Y_i - Y_i \frac{p(Y_i = 0 \mid \vec{x}_i)}{p(Y_i = 1 \mid \vec{x}_i)} \right) \mathbb{I}(x_{il} = x_l) \iff$$

$$1 - P(Y_i = 0 \mid \vec{x}_i) = P(Y_i = 1 \mid \vec{x}_i) = Y_i \text{ s.t. } x_{il} = x_l$$

Which does not have a closed form solution for a single parameter. However, EM could be used to update $p(\tilde{x}_l = 0 \mid x_l)$ iteratively or gradient ascent using the above gradient.

(iii) Assume any X_i may be missing completely at random, and Y is always observed. Is it appropriate to maximize the conditional log likelihood using only rows where all variables are observed? Explain.

Solution:

If X_i is MCAR and Y is always observed, then we have that

$$p(x_i^{(1)}) = p(x_i \mid r_i = 1)$$

Therefore, using this, we have that

$$p(\tilde{x}_i \mid x_i^{(1)}) = p(\tilde{x}_i \mid x_i, r_i = 1)$$

So maximizing the likelihood would be equivalent as maximizing likelihood over the fully-observed data rows.

(iv) Assume the noisy-or model is the true model of the conditional density for $p(Y \mid \vec{X})$ obtained from the observed data distribution for our dataset $[D]$. Will the classifier $\arg\max_y p(Y = y \mid X_1 = x_1, \ldots, X_k = x_k)$ minimize the Bayes risk? Explain.

Solution:

Yes, since we have that the Bayes Risk is given by

$$R_b(\hat{f}) = \mathbb{E}\left[1 - p(\hat{f}(\vec{X}) \mid \vec{X})\right]$$

therefore, using $f(\vec{x}) = \text{argmax}_y\, p(Y = y \mid \vec{X} = \vec{x})$,

$$R_b(f) = \mathbb{E}\left[1 - p(f(\vec{x}) \mid \vec{x})\right] = \mathbb{E}\left[1 - \max_{\hat{f}} p(\hat{f}(\vec{x} \mid \vec{x})\right] = \min_{\hat{f}} \mathbb{E}\left[1 - p(\hat{f}(\vec{x} \mid \vec{x})\right]$$