CS228 Practice Final, Winter 2011-2012

If you thoroughly understand the material of CS228, these questions should not take very much of your time. We expect the longest answers to be on the order of a few sentences..

1. [**5 points**] **Bayes nets versus Markov nets**

   After completing CS228, you decide to take your newfound expertise on the road and consult for various companies in Silicon Valley. As companies hire you to solve their problems, you find that you frequently have to decide between using a Bayes net and a Markov net.

   (a) [**2 points**] For each of the following scenarios, say whether you'd suggest using a BN or MN, and give one sentence explaining why.

   i. Your task is diagnosis of the flu given a set of measurements/symptoms for each patient, not all of which are always known to the doctor. You will now be given a dataset from which to learn, but certain symptoms are medically known to be related to the flu and to each other, and for each one (for example, "fever"), a medical consultant for the company can tell you things like, "The flu causes a fever of at least 102 degrees in 30% of patients." Similarly, the measurements may be related to each other in specific ways that are known to the medical expert, who can tell you things like, "Coughing causes throat irritation in 50% of patients."
   Answer: BN, since all connections are causal and therefore naturally directed, and furthermore you are being given explicit CPDs (or factored CPDs) by the expert, so all parameters can be easily specified.

   ii. A company wants to classify webpages into a known set of classes. Some webpages are already labeled for you, and they want you to use the link relationships between webpages to model similarity in some way (for example, a professor's webpage may point to the webpages of her students and the courses she teaches, which may indicate that these webpages have similar content or topics).
   Answer: MN. While the links are directed, there are cycles in the directed connections between webpages, so an undirected approach, using properties of the links to build the factors, is more appropriate.

   (b) [**2 points**] The companies you deal with want to know how BNs and MNs compare in terms of inference. For each of the following classes of methods, say whether it can be applied to both BNs and MNs, and if so, if there is a difference in efficiency: message-passing, forward sampling, Gibbs sampling, MAP methods.

   Answer: Forward sampling is the set of methods that can only be applied to BNs. Among the others, the efficiency is the same except for MAP with MNs that have special potential forms, in which case the problem can be solved exactly and efficiently.

   (c) [**1 points**] The companies also want to know how BNs and MNs compare in terms of parameter estimation. Is there a reason to prefer one over the other in this regard?

   Answer: BNs can be learned much more easily (in closed form and without inference) in the case of fully observed data. Also, less inference needs to be done for BNs in the case of missing data.

2. **[2 points] Factorization of probability distributions**

   Consider a directed graph $\mathcal{G}$. We construct a new graph $\mathcal{G}'$ by removing one edge from $\mathcal{G}$. Briefly justify your answers to the questions below:

   (a) **True or False:** Any probability distribution P that factorizes over $\mathcal{G}$ also factorizes over $\mathcal{G}'$.

   (b) **True or False:** No probability distribution P that factorizes over $\mathcal{G}$ also factorizes over $\mathcal{G}'$.

   (c) **True or False:** Any probability distribution P that factorizes over $\mathcal{G}'$ also factorizes over $\mathcal{G}$.

   (d) How would your answers change if $\mathcal{G}$ and $\mathcal{G}'$ were undirected graphs?

   Answer: Removing an edge can only add independencies (in both directed and undirected case) since it can't create new active paths. So I($\mathcal{G}$) is a subset of I($\mathcal{G}'$). Any P that factorizes over $\mathcal{G}'$ would satisfy I($\mathcal{G}'$) and therefore I($\mathcal{G}$) and therefore also factorizes over $\mathcal{G}$ too. The opposite isn't always true. Thus (a) is False, and (c) is True.

   (b) is False: e.g., a probability distribution where all variables are independent factorizes over any graph, including both $\mathcal{G}$ and $\mathcal{G}'$.

3. **[2 points] Context-specific independencies**

   In this exercise, we consider the context-specific independencies that arise in ICI models (i.e., models which satisfy a property called *causal independence*, or *independence of causal influence*).

   (a) What context-specific independencies arise if $P(Y|X_1 \dots X_k)$ is a noisy-and (analogous to figure 5.9 in the book on page 176, but where the aggregate function is a determininstic AND)?

   Answer: X_i is independent of X_j given $Y = 1$ since then $X'_i = X'_j = 1$ which blocks the path between X_i and X_j.

   (b) What context-specific independencies arise if $P(Y|X_1 \dots X_k)$ is a noisy-max (where the aggregate function is a deterministic max), where we assume that Y and the X_i's take ordinal values $v_1, \dots v_n$, where $v_1 < v_2 < \dots < v_n$?

   Answer: X_i is independent of X_j given $Y = v_1$ since then $X'_i = X'_j = v_1$ which blocks the path between X_i and X_j.

4. **[2 points] Plate Models**

   Consider Figure 6.7c in the book. We now want to augment the model with the following. Each class and student is associated with a university, which has a reputation. Class difficulties depend on the reputation (perhaps some professors feel that they have a standard to maintain, while others think they can slack off now that the university is famous). Furthermore, the students can make the class honor roll, which depends only on their grades.

   (a) Draw a plate model to represent this scenario.

   Answer: There is a university plate around everything, which has reputation in it, which points to Difficulty. The intersection of the Student plate with the Class plate also gets the Honor Roll variable, which has Grade as a parent.

(b) We now want to introduce a university-wide honor roll in addition to the class honor roll. Can we use a plate model to represent this scenario? If so, draw the plate model; if not, briefly justify your answer.

Answer: You can not, since this would require a directed edge from the student's Grade variable to the honor roll variable which is outside of the student plate.

5. **[3 points] Sampling**

Consider the following scenarios:

(a) You set a robot to execute an open-loop plan, i.e., you give it a fixed sequence of steps to follow (without receiving any feedback from the environment). Since the environment is stochastic, your plan might fail at any of the steps (for example, the robot might run into a chair that is unexpectedly blocking the hallway). You want to estimate the probability that your plan succeeded. Would you use forward sampling, likelihood weighting, or Gibbs sampling? Briefly explain why.

Answer: Forward sampling, since this scenario describes a Bayesian network without evidence.

(b) You have a densely-connected Markov network and would like to estimate the probability over a subset of variables in the network. Would you use forward sampling, likelihood weighting, or Gibbs sampling? Briefly explain why.

Answer: Gibbs sampling.

(c) You are given a data association model where you'd like to associate a discrete-valued property of objects $\boldsymbol{X}$ to a set of observations $\boldsymbol{Y}$. You model this with a selector variable and multiplexor CPD for each Y_i so that $P(Y_i = x_j \mid \boldsymbol{X}, S_i = j) = 1$ and all other probabilities in the CPD are 0. We'd like to estimate the probability over a subset of $\boldsymbol{X}$ using Gibbs sampling, by sampling from the posterior $P(\boldsymbol{X}, \boldsymbol{S}|\boldsymbol{y})$. You can assume the domain for all $\boldsymbol{X}$ and $\boldsymbol{Y}$ is the same. Will this work? Briefly explain why.

Answer: No, because the Gibbs chain is not regular. Since Y_i is deterministically determined by $\mathbf{X}$ and S_i, we can't change X_i if S_i and Y_i are kept fixed. Similarly, assuming all the properties $\boldsymbol{X}$ are distinct, we can't change S_i while keeping $\boldsymbol{X}$ and Y_i fixed.

6. **[3 points] Decomposable Utility Functions**

Recall that a utility function is a mapping from an outcome (assignment to variables) to a real number. Suppose we have M variables, $X_1 \dots X_M$, each of which has a domain of size $|Val(X_i)| = d$, and a utility function $U(X_1 \dots X_M)$ over these variables. Our goal in this problem is to find the "ideal" outcome (i.e., the one that gives us the highest utility). Concretely, we are searching for:

$$(x_1^* \dots x_M^*) = \arg\max_{x_1 \dots x_M} U(x_1 \dots x_M).$$

Assume that this outcome is unique (that is, there are no ties for first-place outcome).

Suppose that our U is decomposable as a sum of utilities for relatively small subsets of the variables:

$$U(X_1 \dots X_M) = \sum_{i=1}^{k} U(\boldsymbol{Y}_i),$$

where $\boldsymbol{Y}_i \subset \{X_1 \dots X_M\}$.

(a) [**2 points**] Describe an algorithm that exactly determines $(x_1^* \dots x_M^*)$ and $U(x_1^* \dots x_M^*)$ and that exploits the structure of the utility function to make it computationally more efficient. Hint: You may use as a subroutine any algorithm that we described in class; in particular, you should be able to use one of the inference algorithms we studied in a straight foward manner.

(b) [**1 points**] Consider the case of chain decomposability, where:

$$U(X_1 \dots X_M) = \sum_{i=1}^{M-1} U(X_i, X_{i+1}).$$

What is the computational complexity of the algorithm?

Answer: Let $V(X) = \exp(U(X))$. Perform max-product with the set of factors defined by V. Then $U^* = U(x^*) = \log V(x^*)$, where x^* is the maximizing assignment found by max-product.

For (b), the biggest clique you need is of size 2, so you have $O(d^2)$ operations per clique, giving a total run time of $O(Md^2)$.

7. [**3 points**] **Structure learning**

In class, we introduced the BIC score as being

$$\text{score}_{\text{BIC}}(\mathcal{G} : \mathcal{D}) = \ell(\langle \mathcal{G}, \hat{\theta}_{\mathcal{G}} \rangle : \mathcal{D}) - \frac{\dim(\mathcal{G})}{2} \ln M.$$

where $\dim(\mathcal{G})$ is the dimension of the model you're using, and M is the number of training instances in $\mathcal{D}$. In assessing the quality of a model, one can also use a related score called AIC (Akaike Information Criterion) that can be defined by:

$$\text{score}_{\text{AIC}}(\mathcal{G} : \mathcal{D}) = \ell(\langle \mathcal{G}, \hat{\theta}_{\mathcal{G}} \rangle : \mathcal{D}) - \dim(\mathcal{G})$$

(a) [**2 points**] As M tends to infinity, we know that the BIC score is consistent. Is this also the case for the AIC score? Briefly explain.

Answer: As M tends to infinity, AIC becomes equivalent to the likelihood score, which we know is not consistent.

(b) [**1 points**] As M tends to infinity, will the models chosen by the AIC be too simple or too complex? Briefly explain.

Answer: As a consequence of (a), AIC will select models that are too complex when M tends to infinity.

8. [**3 points**] **Multi-class image labelling using a CRF**

Consider a multi-class image labeling task using pairwise CRF, where we want to label each pixel in the image into one of the following 21 classes: *building, grass, tree, cow, sheep, sky, airplane, water, face, car, bicycle, flower, sign, bird, book, chair, road, cat, dog, body, boat.* The energy function (base model) we use is in the form of:

$$\boldsymbol{E}(\boldsymbol{X} \mid \mathcal{I}) = \sum_{i=1}^{P} \epsilon_i(X_i) + \sum_{(i,j) \in \mathcal{E}} \epsilon_{i,j}^c(X_i, X_j)$$

where, X_i is a random variable taking one of the 21 classes for pixel i, P is the number of pixels in the image, $\epsilon_i(X_i)$ is a unary term encoding the cost of labeling pixel i into different class, $\epsilon^c_{ij}(X_i, X_j)$ is a pairwise contrast term for each pair of adjacent pixels to encourage label consistency between two adjacent pixels, specifically $\epsilon^c_{ij}(X_i, X_j) = e^{-\text{dist}(p_i,p_j)} \cdot \mathbf{1}\{X_i \neq X_j\}$ which means that if pixel i and j take the same value, there is no cost, otherwise the cost depends inversely on the distance of the appearance of these two pixels (measured by the function $\text{dist}(p_i, p_j)$, where p_i and p_j are the feature vectors in some pixel representation space, e.g., RGB space). We find the optimal labeling as $\boldsymbol{x}^* = \text{argmin}_{\boldsymbol{x}} \boldsymbol{E}(\boldsymbol{x} \mid \mathcal{I})$.

(a) **[1 points]** Now suppose we want to improve the base model by adding another pairwise term $\epsilon^l_{ij}(X_i, X_j)$ that encodes the full pairwise affinity between classes. Specifically, this factor can be thought of as a 21×21 matrix that encodes the potentials between the classes (as opposed to $\epsilon^c_{ij}(X_i, X_j)$ which just looks at whether pixels i and j belong to the same class or not). What does this allow to represent that we couldn't before in the base model? Give your answers as a couple of examples involving some of the classes. Hint: think about the spatial relations between classes in the image.

Answer: This allows us to represent that cows are usually on grass, cars are rarely in water, etc.

(b) **[1 points]** Suppose now you are given a set of training images for each of which pixels are labeled into different classes manually. However, due to the laziness of the labeler, lots of the boundaries between two object classes are not labeled; e.g., there are often areas as large as 5 pixels wide labeled “unassigned” between the car and the road, or the cow and the grass. Consider how we can model these “unassigned” labels in the data. If we add an “unassigned” class label, what are the issues in terms of the parameters we learned for $\epsilon^l_{i,j}(X_i, X_j)$? Hint: think about the motivation of introducing the new term in part(a).

Answer:

Since the “void” class separates two adjacent classes, we don't learn meaningful parameters that encode the affinity between the original two adjacent classes. In the learned model, everything will be likely to be next to “void” class which is uninformative.

(c) **[1 points]** What would be a better way of modeling these “unassigned” pixels?

Answer: You can treat those “void” pixels as hidden variables which we don't observe during the training.

We can use EM to train the model, where the E-step will complete the label in a soft way. Note that since we have hidden variables, the model may be unidentifiable.

Comments: in practice, we don't use this model at the level of pixels, since the case where pixels with the same label are next to each other is much more frequent than pixels with different labels (the number of boundaries pixels are much smaller than the number of pixels inside the objects). Therefore, we usually group pixels based on the similarity of their appearances into larger regions first, and use this model on top of the regions instead of pixels.

9. **[2 points] Value of Imperfect Information**

In a decision problem, we know how to calculate the value of perfect information of X at decision D. Now imagine that we cannot observe the exact value of X, but we can instead observe a noisy estimate of X.

For this problem, assume X is binary. Also assume the noisy observation has a false positive rate of p and a false negative rate of q. (That is when $X = 0$ we observe 1 with probability p, and when $X = 1$ we observe 0 with probability q.)

Give a simple method by which we can calculate the improvement in MEU from observing this imperfect information. Your answer should be just a couple lines long, but you should explain exactly how p and q are used. Hint: consider using VPI as a black box.

Answer: We introduce a new node, Y into our influence diagram, as a child of X, and CPD according to the given noise model - $P(Y|X) = 1 - p, q; p, 1 - q$. Then simply calculate the VPI for Y at decision D.