Probablistic Graphical Models, Fall 2007
Final Exam

Due 12/13/2007 by 2pm electronically to 10708-07-instr@cs.cmu.edu or paper version to Monica Hopes

Instructions

The final must be done individually. You are *not allowed* to discuss the exam with anyone except the instructors. The exam is open-book but *not* open-Google, i.e, you are allowed to use any of (*and only*) the material distributed in class. This includes the slides and the handouts given in the class[1]. You cannot use your late days for the final.

1 [8 pts]I-maps

In this question, you are going to dig into your long-forgotten memories about I-maps.

1.1

Prove that two network structures $\mathcal{G}_1$ and $\mathcal{G}_2$ over the same variables are I-equivalent **if** the following two conditions hold:

1. the two graphs have the same set of trails;
2. a trail is active in $\mathcal{G}_1$ iff it is active in $\mathcal{G}_2$.

Hint: Think d-separation.

1.2

Using your result from part 1.1, prove that **if** $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same skeleton(i.e. the graphs have the same set of vertices and edges if you neglect edge directions) and the same set of v-structures then they are I-equivalent.

2 [10 pts] Search in Structure Learning

Consider learning the structure of a Bayesian network for some given ordering,$\prec$, of the variables, $X_1, \ldots, X_n$. As described in class, this can be done efficiently. Now assume that we want to perform search over the space of orderings, i.e. we are searching for the network (with bounded in-degree k) that has the highest score. We do this by defining the score of an ordering as the score of the (bounded in-degree) network with the maximum score consistent with that ordering, and then searching for the ordering with the highest score. We bound the in-degree so that we have a smaller and smoother search space.

We will define our search operator, o, to be Swap X_i and X_{i+1} for some $i \in \{1, \ldots, n-1\}$. Starting from some given ordering, $\prec$, we evaluate the BIC-score of all successor orderings, $\prec'$, where a successor ordering

[1]Please contact Monica Hopes(meh@cs)if you need a copy of a handout.

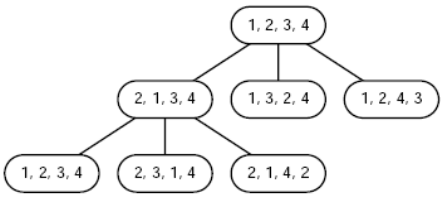

Figure 1: Partial search tree example for orderings over variables X_1, X_2, X_3, X_4. Successors to $\prec = (1,2,3,4)$ and $\prec' = (2,1,3,4)$ shown

is found by applying o to $\prec$(See figure 1). For example, if n, the number of variables in the BN is 4, then starting from $\prec = \{1, 2, 3, 4\}$, the possible orderings $\prec'$ that can be reached using o are $\{2, 1, 3, 4\}, \{1, 3, 2, 4\}$ and $\{1, 2, 4, 3\}$ We now choose a particular successor, $\prec'$ with the best BIC-score and repeat this process.

Provide an algorithm to efficiently compute the BIC-score for the successors of the new ordering, $\prec'$, given that we have already computed the scores for successors of $\prec$.

3 [10 pts] Factorial HMM

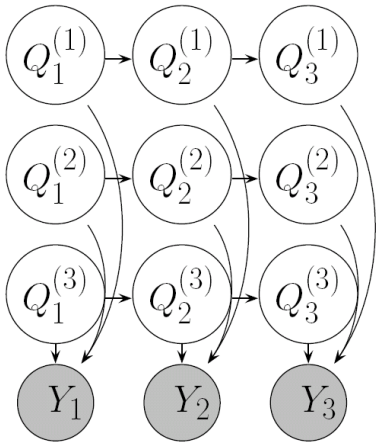

Figure 2: Factorial HMM for Question 3. The figure shows the model for n=3, T=3

A *factorial HMM* (Fig 2) is a DBN over $Q^{(1)}, Q^{(2)}, \ldots Q^{(n)}, Y$, such that the only parent of $Q_t^{(i)}$ is $Q_{t-1}^{(i)}$ and the parents of Y_t are $Q_t^{(1)}, Q_t^{(2)}, \ldots Q_t^{(n)}$. Consider the problem of using a structured variational approximation to perform inference over the unrolled network for a fixed number of T time slices.

3.1

Consider a space of approximate distributions of Q composed of disjoint clusters $\{Q_1^{(i)} \cdots Q_T^{(i)}\}$ for i=1...n and $\{Y_t\}$ for t=1...T (i.e., each row of $Q^{(i)}$'s in Fig 2 forms a cluster and each individual Y_t forms a cluster). Show the variational update equations, describe the use of inference to compute the messages, and analyze the computational complexity of the algorithm.

3.2

Consider a space of approximate distributions of Q composed of disjoint clusters $\{Y_t, Q_t^{(1)} \dots Q_t^{(n)}\}$ for t=1...T (i.e., each cluster is a column in Fig 2). Again, show the variational update equations, describe the use of inference to compute the messages, and analyze the computational complexity of the algorithm.

3.3

Discuss the circumstances when you would use one approximation over the other.

4 [15 pts] Updating Clique Trees

Consider a simple chain structured graphical model $X_1 - X_2, \dots, X_n$ and its calibrated clique tree. The clique tree in this case would have cliques of size atmost 2. Now suppose, for some i, the factor $f(X_i, X_{i+1})$ in this model is modified, ie $f(X_i, X_{i+1})$ is replaced by $f'(X_i, X_{i+1})$. We wish to find the modified marginals of X_j for some $1 \leq j \leq n$, $j \neq i$.

4.1

In question 3 of Homework 2, we saw how to update a clique tree when a *new* factor is introduced. You showed that the junction tree can be re-calibrated by passing messages from the modified clique to all other cliques. Show (with simple pseudocode) how a similar approach can be used to solve this problem(Yes, this is easy). Your procedure should take as inputs $i, f'(X_i, X_{i+1})$ and j, and return the marginal for X_j. Your procedure should, in the worst case, take O(n) time to update all the marginals.

The aim of the rest of this exercise is to improve this worst-case estimate at the expense of some extra space overhead. Let us simplify the problem a bit to get some insight.

4.2

Suppose that both i and j are fixed. Write a procedure that takes $f'(X_i, X_{i+1})$ as a parameter and returns the marginal for X_j in O(1) time. *Hint: The only variables you care about are i and j.*

4.3

Question 4.2 shows that it is possible to speed up computation of marginals of a specific variable, if the effect of variables on the path are marginalized out. Now, consider the original problem that 4.1 solved in O(n) time. Show how to speed up this update process from O(n) to O(log(n)) time at the expense of a constant factor increase in memory requirement. Your procedure may take O(n) pre-processing time before the queries start. You will need to clearly describe your solution including pseudo code and a proof of correctness.

Hint: Construct a binary tree over the nodes of the graph. Store some extra information in the internal nodes of the tree as a pre-processing step. Send messages up the tree in case of a change in a factor and compute marginals by sending messages down the tree.

5 [10 pts] Gibbs Sampling

Consider the boltzman distribution on a lattice as shown in figure 3. The joint for this model is defined as

$$P(X) = \frac{1}{Z} \exp\left(\sum_{(i,j)} \theta_{ij} X_i X_j + \sum_i \theta_i X_i\right)$$

Each variable X_i in the graph has a discrete state space. The model has single and pairwise potentials characterized by θ_i and θ_{ij} respectively.

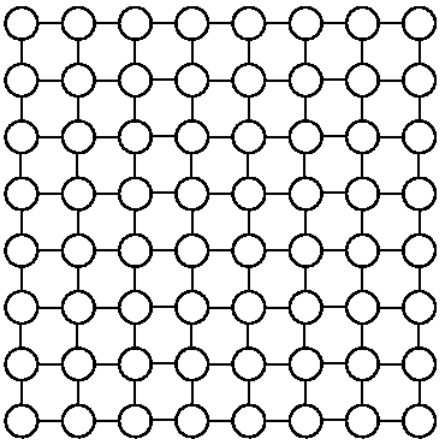

Figure 3: Graphical model for Question 5.1

5.1

Based on the lecture in class on Sampling, show how you would perform inference on this graphical model using a Gibbs Sampler.

5.2

Write the updates equations for the same model using naive mean field. Compare these equations with those of the Gibbs Sampler.

5.3

Now, consider any partition of the vertices into non-overlapping regions. Describe how you could change the solution to 5.1 to sample blocks of variables at a time where each block is a non-overlapping region of vertices. Compare these equations with the update equations of Generalized Mean Field with a cluster for every non-overlapping region.

5.4

Consider the LDA model that we discussed in class and in HW4. Derive the Gibbs sampling equations for this model. You need not derive equations for the collapsed gibbs sampler mentioned in class.

6 [11 pts] Structure Learning in Undirected Graphical Models

For this problem, assume that you have i.i.d. data sampled from a distribution $P(\mathcal{X})$. P is represented by a Markov Random Field whose graph structure is unknown. However, you do know that each node has at most d neighbors.

6.1

Show why knowing the Markov blanket of each node is sufficient for determining the graph structure.

6.2

For any node X and its Markov blanket $MB(X)$, we know that

$$P \models (X \perp \mathcal{X} - X - MB(X) | MB(X))$$

Briefly, why might you need *a lot* of data to test for this conditional independence directly?

6.3

For disjoint sets of variables **A** and **B**, let conditional entropy be defined as,

$$H(A|B) = -\sum_{a,b} P(A = a, B = b) \log P(A = a|B = b)$$

Prove that for any node X, $H(X|MB(X)) = H(X|\mathcal{X} - X)$.

6.4

For disjoint sets of variables A, B and C, we have that

$$H(A|B, C) \leq H(A|B)$$

In other words, information never hurts. Prove that $MB(X) = \arg\min_{\mathbf{Y}} H(X|\mathbf{Y})$

6.5

Using the intuition developed in the previous parts, describe a structure learning algorithm for Markov Random Fields, assuming the constraint that each node has at most d neighbors. Your algorithm should run in $O(n\binom{n}{d}c)$ time, where n is the number of nodes in the model, and c is the complexity of computing the conditional entropy $H(X|Y)$, when $|Y| \leq d$.

6.6

If we removed the constraint that each node have at most d neighbors and instead changed our optimization problem to include a penalty term $MB(X) = argmin_{\mathbf{Y}} H(X|Y) + |Y|$, how would the time complexity of the algorithm change?

7 [11 pts]Free Energy and EM

Consider an undirected graphical model with discrete nodes. Let S represent the vector of values of all of the nodes. Define an *energy function* U to be a function from configurations S to the reals. For an arbitrary probability distribution $q(s)$, define the *variational free energy*:

$$F(q) = \sum_s q(s)U(s) + \sum_s q(s) \ln q(s) \tag{1}$$

7.1

Show that the Boltzmann distribution:

$$p(s) = \frac{e^{-U(s)}}{Z}$$

minimizes the variational free energy across all possible choices of $q(s)$.

7.2

Now define the *free energy*:

$$F = \sum_s p(s)U(s) + \sum_s p(s) \ln p(s) \tag{2}$$

where $p(s)$ is the Boltzmann distribution, and show that $F = -\ln Z$. This is the fundamental relationship between the free energy and the partition function.

7.3

Now partition S as $S = H \cup E$, where H are the hidden variables and E is the evidence, and consider conditional distribution $q(h|e)$. Recall the general form of the E step of the EM algorithm:

$$q^{(t+1)} = \arg\max_q \mathcal{L}(q, \theta^{(t)}).$$

where $\theta^{(t)}$ is the estimate of the paratemers θ at time step t. Show that $\mathcal{L}$ is a negative variational free energy under an appropriate definition for the energy function.

Using this energy function and the result you derived in 7.1, where the sum is taken only over the possible assignments h to the hidden variables H, prove that $q^{(t+1)}(h|e) = p(h|e)$ is the general form for an exact E step. Substitute this distribution into the free energy, evaluate the free energy, and, using the result from 7.2, specify what the partition function Z is in statistical terminology.

8 [15 pts]Sampling for a switching Kalman Filters

Consider a Markovian switching Kalman Filter (Z, X, O) where Z denotes the switching variable, X denotes the state variable and O denotes the observation variable. The switching variables here are binary and Markovian. Thus, Z_{i+1} has Z_i as its parent; X_{i+1} has X_i and Z_i as its parents; and O_{i+1} has X_{i+1} as its parent.
Let the parameters of the model be given by,

$$P(X_{i+1}|X_i, Z_i = j) \sim N(\beta_0^j + \beta^j X_i, \Sigma^j) \tag{3}$$
$$P(Z_{i+1}|Z_i = j) \sim Bernoulli(\alpha^j) \tag{4}$$
$$P(O_{i+1}|X_{i+1}) \sim N(\beta_0 + \beta X_{i+1}, \Sigma) \tag{5}$$

In this question, we look at the task of computing the expected value of a function $f(X, Z)$ under the posterior $P(X_t, Z_t|o_{1:t})$ via sampling, at each time step t.

8.1

One approach is to use the weighted resampling technique of Particle Filtering. In particle filtering, the distribution $P(X_t, Z_t|o_{1:t})$ is represented by a weighted set of samples (particles) of X and Z.

8.1.1

Give the time and measurement updates for particle filtering for the above model. Recall that the time update involves getting samples from $P(X_{t+1}, Z_{t+1}|o_{1:t})$ given samples from $P(X_t, Z_t|o_{1:t})$; and the measurement update involves getting samples from $P(X_{t+1}, Z_{t+1}|o_{1:t+1})$ given the above samples from $P(X_{t+1}, Z_{t+1}|o_{1:t})$. You are given two sampling subroutines, one that samples from $Bernoulli(\alpha)$, for any α, and one that samples from $N(\mu, \Sigma)$, for any μ and Σ.

8.1.2

Express the estimate of $E_{p(X_t,Z_t|o_{1:t})}[f(X,Z)]$ in terms of the weighted samples from the distribution $P(X_t, Z_t|o_{1:t})$.

8.2

Particle Filtering is a high variance method particularly for high dimensional state variables. In class, we saw how to reduce the variance of sampling by Rao-Blackwellization. Here, the distribution $P(X_t, Z_t|o_{1:t})$ is represented explicitly by a mixture of Gaussians. Specifically, you will have weighted samples of Z, and each of these samples will be associated with a Gaussian bump over X.

8.2.1

Give the time and measurement updates for the Rao-Blackwellized particle filter. Recall that you will using sampling for the transition of the discrete variable, but exact Gaussian filtering for the continuous part. In addition to the sampling subroutines, you are given subroutines for multiplying, conditioning, and marginalizing conditional linear Gaussians (please, be explicit about what distributions you are giving as the input to these subroutines.)

8.2.2

Using the samples of Z and an expectation over the conditional Gaussian distribution over X, compute the Rao-Blackwellized sampling estimate for $E_{P(X_t,Z_t|o_{1:t})}[f(X,Z)]$. Here, you are given a subroutine that computes $E_{N(\mu,\Sigma)}[f(X,Z)]$.

9 [10 pts]MLE in Undirected Graphical Models

Consider an undirected graphical model $\mathcal{G}$ with the joint defined as

$$p(x_\nu) = 1/Z \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

where x_ν is the set of all random variables in the graphical model, $\mathcal{C}$ is the set of cliques in the graph and ψ_C, the clique potential for clique C. Suppose we have N observed datum generated (i.i.d) by this model and that wish to find maximum likelihood estimates of ψ_C given this data.

9.1

Write the formula for the log-likelihood l of the data under this model.

9.2

Show that

$$\frac{\partial l}{\partial \psi_C(x_C)} = N\left(\frac{\tilde{p}(x_C)}{\psi_C(x_C)} - \frac{p(x_C)}{\psi_C(x_C)}\right) \tag{6}$$

where $\tilde{p}(x_C)$ is the empirical marginal distribution of x_C. Therefore $\hat{p_{ML}}(x_C) = \tilde{p}(x_C)$ where is the empirical marginal distribution of x_C.

9.3

A common strategy to solve systems of implicit equations like Eqn.6 is to iterate until a fixed-point is reached. Let us apply the same approach to construct an algorithm for MLE estimates in such models.

The update equations at any of the step t of the algorithm, can be written as

$$\psi_C^{t+1}(x_C) = \psi_C^t(x_C)\frac{\tilde{p}(x_C)}{p^t(x_C)}$$

where $\tilde{p}(x_C)$ is the empirical marginal, $\psi_C^t(x_C)$ the estimates of the parameters at iteration t and $p^t(x_C)$ the probability distribution at time t based on these estimates.

Show that $p^{t+1}(x_C)$ is equal to the empirical marginal $\tilde{p}(x_C)$ and that the normalization factor remains constant across our updates, ie. $Z^{t+1} = Z^t$

9.4

Show that this implies that the update can be alternatively formulated in terms of joint probabilities as

$$p^{t+1}(x_\nu) = p^t(x_\nu)\frac{\tilde{p}(x_C)}{p^t(x_C)}$$