10708 Probabilistic Graphical Models: Final Exam

Due Dec 10th by Noon electronically to 10708-instr@cs.cmu.edu or paper version to Michelle Martin, or by fax to 412-268-3431,

Your final must be done individually. You may not discuss the questions with anyone other than Carlos or the TAs (you are free to ask us questions by e-mail or in person if you are having problems with a question). The exam is open book, but not open-Google, *i.e.*, you can use any materials we discussed in class or linked to from the class website. You are not allowed to look at other sources. However, you may use a calculator or Matlab to do numerical computations, if necessary. Each question has the name of one of the TAs beside it, to whom you should direct any inquiries regarding the question. Please submit your final in two parts, one for each TA. Also, please put the TA's name on top of the final.

If you hand in your assignment early, you can get bonus points.

HANDIN	BONUS
Monday, Dec 8, 2pm	3 pts
Tuesday, Dec 9, 2pm	2 pts

You may *NOT* use late days on the final.

1 Short answers [Amr] [6 pts]

1. For each of the following questions, answer *true* of *false* justifying your answer in (1-3 sentences).
   (a) i. G_1 is I-equivalent to G_2
   ii. G_1 is I-equivalent to G_3
   iii. If G_1 is a perfect map for $\mathcal{P}$, then G_3 is an I-map for $\mathcal{P}$
   (b) If G_1 is a perfect map for $\mathcal{P}$, given a infinite samples, $\mathcal{D}$, from $\mathcal{P}$:
   i. $\text{Score}_{ML}(G_1; \mathcal{D}) = \text{Score}_{ML}(G_2; \mathcal{D})$
   ii. $\text{Score}_{ML}(G_1; \mathcal{D}) < \text{Score}_{ML}(G_3; \mathcal{D})$
   iii. $\text{Score}_{BIC}(G_1; \mathcal{D}) = \text{Score}_{BIC}(G_2; \mathcal{D})$

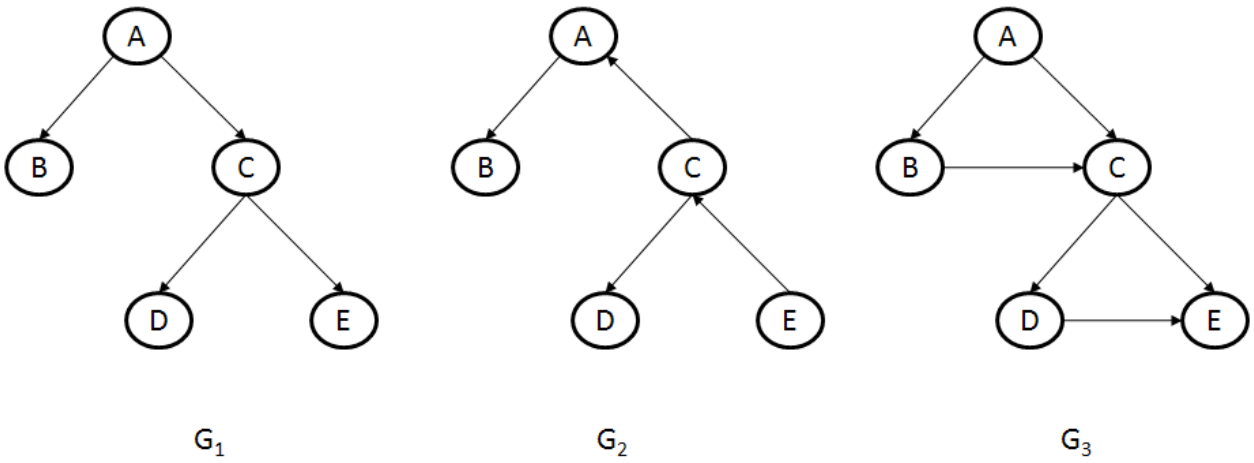


Figure 1: Graphs used in question 1

iv. $\text{Score}_{BIC}(G_1; \mathcal{D}) < \text{Score}_{BIC}(G_3; \mathcal{D})$

(c) Let $\mathcal{K}$ be a cluster graph for a distribution $\mathcal{P}$ (*i.e.* it satisfies the generalized running intersection and family preservation properties). If $\mathcal{K}$ happens to be a tree, then $\mathcal{K}$ is a clique tree as well for $\mathcal{P}$. (hint: either provide a counter example or show that the generalized RIP reduces to RIP in this case)

2 Structure Learning in Undirected Models [Dhruv] [12 pts]

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.

1. Show why knowing the Markov blanket of each node is sufficient for determining the graph structure.
2. For any node X and its Markov blanket $\text{MB}(X)$, we know that

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

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

3. For disjoint sets of variables $\mathbf{A}$ and $\mathbf{B}$, let conditional entropy be defined as,

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

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

4. For disjoint sets of variables $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, we have that

$$\mathrm{H}(\mathbf{A}|\mathbf{B}, \mathbf{C}) \leq \mathrm{H}(\mathbf{A}|\mathbf{B}).$$

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

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 your model, and c is the complexity of computing the conditional entropy $\mathrm{H}(X|\mathbf{Y})$, when $|\mathbf{Y}| \leq d$.

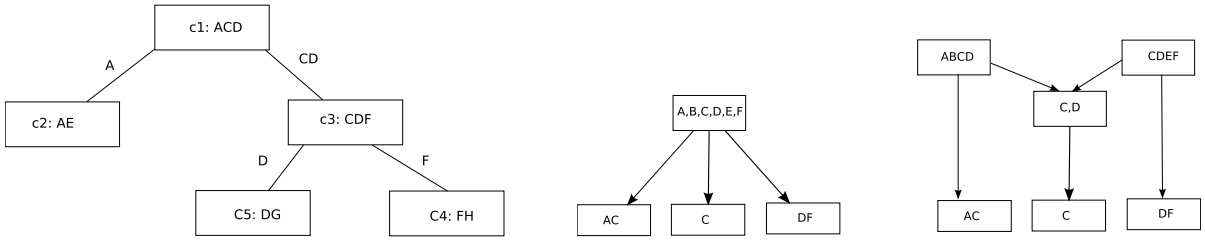
6. If we removed the constraint that each node have at most d neighbors and instead changed our optimization problem to include a penalty term, $\mathrm{MB}(X) = \mathrm{argmin}_{\mathbf{Y}}\{\mathrm{H}(X|\mathbf{Y}) + |Y|\}$, how would the time complexity of the algorithm change?

3 Trading Inference Accuracy vs. Efficiency [Amr 16 pts]

Note: Nearly half of this problem is marked as extra credit.

As we know from class, compact representation does not translate into efficient inference. For trees, the cost of inference is linear in the size of the factors. For loopy graphs, we can either convert it to a clique tree and then use belief propagation, or just run the belief propagation algorithm on the loopy graph(which is equivalent to first constructing a bipartite factor graph as we discussed in class, and then running BP on it.) In the former case, inference is exact but its cost is exponential in the largest clique size. In the later case inference is approximate, but its cost is linear in the size of the largest factor. The above two solutions comprises two extremes in the space of efficiency vs. exactness. In both cases inference is carried via BP over a given graph: in clique trees, the graph vertices correspond to maximal cliques, while in factor graphs, the graph vertices correspond to factors and singleton variables in the original network. Region graphs generalize the above two solutions by filling the gap between them, and allowing for trading efficiency vs. exactness. In this question we will explore this connection in more details.

Region Graphs (K&F 10.3): Recall that a region graph is a directed acyclic graph with vertices correspond to regions which are subsets of the variables. If an edge exists from region r to region r' then scope$[r'] \subset$ scope$[r]$. Each factor ϕ in the original network is assigned to a top-most region r such that Scope$[\phi] \subset$ scope$[r]$. The region graph is calibrated via a message-passing algorithm similar to BP. The exact message passing rules are given in 10.37 in (K&F). Upon calibration, the belief of each region is computed using 10.36 in (K&F).



(a) $\mathcal{T}_1$ (b) The split operator: $AB \perp EF|CD$

Figure 2

3.1 Clique trees and factor graphs as region graphs

1. Represent the clique tree $\mathcal{T}_1$ in Figure 2a as a region graph $\mathcal{R}_1$? (Hint: the region graph has two levels)
2. $\mathcal{T}_1$ can be calibrated using BP as follows. First pick a root, then pass messages upward from leaves to root using BP (i.e. sum-product), and finally pass messages downward from root to leaves using BP. Similarly $\mathcal{R}_1$ can be calibrated using message passing as in 10.37 in K&F. If we initialize all messages in $\mathcal{R}_1$ to 1, show that there exists a schedule of sending messages that 1) will converge in only one pass, and 2) will achieve the same final beliefs over all cliques and sepsets as in $\mathcal{T}_1$. (Hint: it is easier to find a schedule of sending messages over $\mathcal{R}_1$ that parallel the messages sent during calibrating the clique tree)
3. Given a clique tree $\mathcal{T}$ with cliques $\mathbf{C}$ and sepsets $\mathbf{S}$, sketch a scheme of representing it as an equivalent region graph $\mathcal{R}$. (hint: generalize your solution to part 1). If $\mathcal{T}$ has N cliques, how many regions and edges does $\mathcal{R}$ contain?
4. Given a factor graph, show how to represent it as a region graph. (hint: this is easy) We can also show that loopy belief propagation is equivalent to running the region graph's message-passing algorithm over the above representation (You do NOT have to show that).

3.2 Building region graphs top-down

The cost of sending messages over the region graph is exponential in the largest region size [1]. Therefore, by constructing a region graph with *bounded* region size, one can trade efficiency vs. exactness. We will define *the width of a region graph* to be the size of its largest region.

[1] Assuming a naive implementation of region graphs. In practice, for each region, one should only store its factors and then perform variable elimination when sending each message, therefore the effective width is the largest clique size that results during this inference step. As we know from class, we can always estimate this size once by inspecting the factors used in the elimination process and the elimination order

Definition 1: Region graph invariance: There are operations that when applied on a region graph $\mathcal{R}$, result in an equivalent region graph $\mathcal{R}'$ such that after calibration, both $\mathcal{R}$ and $\mathcal{R}'$ agrees on the same marginals. One of these operations is *split* illustrated in Figure 2b and defined as follows. Consider an outer region (that is a region in the top-most level) C_r. If we can find a partitioning of the variables in scope$[C_r]$ as (C_{r1}, C_{r2}, C_{r3}) such that $C_{r1} \perp C_{r3}|C_{r2}$, then we can replace C_r with three regions: C_{r12}, C_{r23} and C_{r2} whose scope is given by $C_{r1} \cup C_{r2}, C_{r2} \cup C_{r3}$ and C_{r2} respectively. Moreover, C_{r2} will become a child of both C_{r12}, C_{r23}, and all edges $C_r \rightarrow C_j$ will emanate from one of the new regions with the smallest scope that contains scope$[C_j]$ (see Figure 2b for an example).

1. Given a clique tree $\mathcal{T}$ with cliques $\mathbf{C}$ and sepsets $\mathbf{S}$, let R_T be its equivalent region graph as you derived in 3.1.3. Let R be a region graph that consists of a single region r such that scope$[r]= \mathcal{X}$ (i.e. all the variables). Show that using a sequence of split operations, R can be morphed into the region graph $\mathcal{R}_T$.
2. Briefly conclude that inference using $\mathcal{R}_T$ is thus exact.
3. **[Extra credit 4pts]** Given $\mathcal{R}_T$,one can not reduce its width further using the split operator. However, one can define a *soft_split* operator that behaves exactly like *split*, but only requires that C_{r1} is almost independent from $C_{r3}|C_{r2}$. We define almost independent such that $I(C_{r1}; C_{r3}|C_{r2}) <$ threshold. Clearly the resulting region graph after the soft_split operation will result in approximate inference. Using this new operator, show how to build a region graph with width W. (Hint: there is a subtle point here that you should think about. What happened if based on the soft_split criteria a region $r = ABCD$ that has two children $r_1 = BCD$ and $r_3 = ABC$ is splitted into AB, B and BCD?)
4. **[Extra credit 2pts]** Why is it hard to evaluate the operator *soft_split* in practice? (hint: this is an instance of a chicken and egg problem)

3.3 Building region graphs bottom-up [Extra credit]

One interpretation of the operation of calibrating a region graph is that it is equivalent to optimizing a free energy functional called Kikuchi free energy. In this view, the fixed-point regions' beliefs are maxima of this functional. Lets assume that for a calibrated region graph $\mathcal{R}$ that $F(\mathcal{R})$ is the value of the free energy achieved. We will use $F(\mathcal{R})$ as a proxy for the approximation quality when $\mathcal{R}$ is used for inference. More specifically, $\mathcal{R}_1$ gives a better approximation than $\mathcal{R}_2$ if $F(\mathcal{R}_1) > F(\mathcal{R}_2)$. Starting from a base region graph, we would like to find the best region graph with bounded width W by greedily adding a bigger outer region at each iteration[2]. We will use a factor graph as the base region graph[3].

[2]An outer region is a region with no parent.

[3]in this case, the Kikuchi free energy reduces to the Bethe free energy discussed in class

1. **[5 points]** Assume there exists an oracle named Candidates($\mathcal{R}$) that given a region graph, it returns candidate regions that are NOT strictly contained in any current region in $\mathcal{R}$(but definitely overlap with some regions in $\mathcal{R}$,why?). Starting from a base factor graph, give an algorithm that iteratively builds the best (in the sense defined above) region graph of bounded width w.
2. **[2 points]** How would you implement the Candidates($\mathcal{R}$) oracle? (your answer should be brief and *efficient*, there is no need for a theoretical justification [this is an open problem], you can just give the intuition behind your reasoning)

4 KL Projection in Assumed Density Filtering [Dhruv] [12 pts]

In class, we discussed the Boyen-Koller algorithm, an instance of assumed density filtering, where the belief state is represented by a clique tree. At each time step, the belief state becomes more complex, and we project it into a simpler clique tree by doing a simple marginalization. This approach may seem like a hack, but, in this question, you will show that this marginalization is a well-defined, KL-minimizing projection.
Consider the clique tree T_1 (corresponding to the complex belief state in Boyen-Koller):

$$ABC - BCD - CDE$$

Let the cliques be calibrated, so that we have all the clique marginal probabilities.
Consider also the clique tree T_2 (corresponding to the simpler belief state in BK):

$$AB - BC - CD - DE$$

A KL projection of a distribution P over a set of distributions S is given by,

$$P_S = \arg\min_{Q \in S} KL(P||Q)$$

Let P denote the distribution given by the calibrated clique tree T_1; and let S denote the set of distributions represented by clique tree T_2 (i.e., for which T_2 is an I-map). Show that the KL Projection of P over S is given by setting the clique probabilities in T_2 to be the marginals of the corresponding clique probabilities in T_1. That is, $P_{T_2}(AB) = \sum_C P_{T_1}(ABC)$, and so on.

5 Gaussian Graphical Models [Dhruv] [13 pts]

5.1 Warming up

Lets first get acquainted with the canonical parameterization of Gaussian distributions. (Notation: Throughout this question, we will refer to Gaussians in standard form as $N(\cdot;\mu,\Sigma)$ and Gaussians in canonical form as $N_c(\cdot;\eta,\Lambda)$) You are given the joint distribution over X and Y in standard form:

$$P(X,Y) = N\big(X,Y;\mu = \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.75 \end{bmatrix}\big)$$

1. Write down $P(X,Y)$ in canonical form.
2. Write down $P(Y)$ in canonical form.
3. Using your answers from parts (1) and (2), write down $P(X|Y)$ in canonical form.

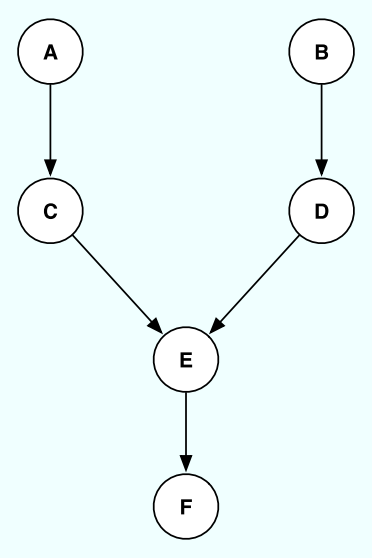
   (Hint 1: To multiply two Gaussians in canonical form, you simply add the parameters, and to divide two Gaussians in canonical form, you subtract the parameters, filling in with zeros as necessary. Refer to the Kalman filter slides for more details.)

   (Hint 2: Your answer to part (c) in canonical form will be represented as a multivariate distribution over X and Y, not a univariate distribution over X as would be the case in standard form.)

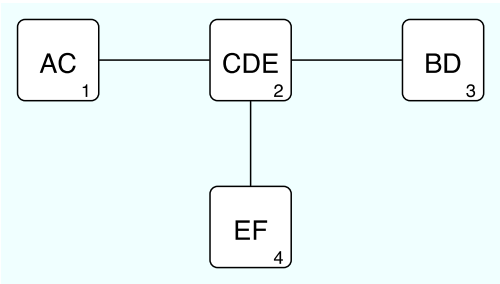
5.2 Message passing in Gaussian Graphical Models

In Figure 3a we give you a Gaussian graphical model with the following conditional probability distributions (all given in canonical form):

$$\begin{aligned}
P(A) &= N_c\big(A;\eta=9,\Lambda=1\big) \\
P(B) &= N_c\big(B;\eta=1,\Lambda=0.6\big) \\
P(C|A) &= N_c\big(C,A;\eta=\begin{bmatrix} 0 \\ 0 \end{bmatrix},\Lambda=\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\big) \\
P(D|B) &= N_c\big(D,B;\eta=\begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix},\Lambda=\begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix}\big) \\
P(E|C,D) &= N_c\big(E,C,D;\eta=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\Lambda=\begin{bmatrix} 1 & -9 & 0.5 \\ -9 & 81 & -4.5 \\ 0.5 & -4.5 & 0.25 \end{bmatrix}\big) \\
P(F|E) &= N_c\big(F,E;\eta=\begin{bmatrix} 2 \\ 2 \end{bmatrix},\Lambda=\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\big)
\end{aligned}$$



(a) Gaussian Graphical Model



(b) Junction Tree

Figure 3

In this problem, you will use the Shafer-Shenoy message passing scheme for Gaussians to perform exact inference on this model. The Shafer-Shenoy algorithm for Gaussian graphical models is analogous to the algorithm presented in class for discrete models. However, you will need to take into account how to multiply potentials together and how to marginalize out variables in the canonical Gaussian setting as discussed in 5.1. In your answers to the questions below, your Gaussian distribution should be written in *canonical* form.

1. Give an elimination ordering for the Bayesian network in Figure 3a that would result in the junction tree in Figure 3b.
2. Using the Family Preserving Property, assign the given CPDs to appropriate cliques in the junction tree, then compute the initial clique potentials $\Pi_1^{(0)}$, $\Pi_2^{(0)}$, $\Pi_3^{(0)}$, and $\Pi_4^{(0)}$.
3. Compute $P(C, D, E)$ using Shafer-Shenoy message passing. Write down the three messages that were needed to compute this probability. Both your final answer and your messages should be represented as Gaussian distributions in canonical form.
4. Given the messages computed for part 3, what additional message would you need if you wanted to compute $P(A|C)$? Write down this message. (Note: you do not need to compute $P(A|C)$.)
5. Given that a minimal junction tree for a Bayesian network with n nodes can have at most n cliques, what is the time complexity of Shafer-Shenoy on a Gaussian graphical model with n nodes and induced tree width w? Briefly justify your answer in one or two sentences. (Hint: Time complexity of matrix inversion for a $k \times k$ matrix is $O(k^3)$.) What is the running time of a discrete model over the same BN structure, where each

variable takes on at most c values?

6. **[Extra Credit] [3 pts]** If we wanted to compute $P(C, D, E)$, an alternative to message passing would have been to multiply together all the CPDs, form a single matrix for the distribution $P(A, B, C, D, E, F)$ and directly marginalize out all the other variables. What is the time complexity of this operation? Briefly justify your answer in one or two sentences.

6 Variational Free Energy [Amr] [18 pts]

Once upon a time there were three bears, a mother bear, a father bear, and a baby bear. The mother was a frequentist; the father a pragmatic Bayesian. However, the baby bear thought that his parents were terribly silly – his mother prone to knitting sweaters that were far too snug, that overfit; his father far too stern in his insistence that he choose a single sweater to wear each day. The baby bear preferred to think of himself as wearing a distribution over sweaters, making him the only proper Bayesian bear in the whole forest.

Knowing the bears' fondness for parameter estimation, a little blonde girl comes along to steal their work. Finding no one home, she looks at the desks of each bear and finds,

Definition 1 (Maximum Likelihood) *If y are the observed variables and θ the model parameters then the maximum likelihood criterion is*

$$\hat{\theta}_{ML} = \operatorname*{argmax}_{\theta} \log p(y|\theta)$$

Definition 2 (Maximum a Posteriori) *If y are the observed variables and θ the model parameters then the maximum a posteriori criterion is*

$$\hat{\theta}_{MAP} = \operatorname*{argmax}_{\theta} \log p(\theta|y)$$

Definition 3 (Fully Bayesian) *If y are the observed variables and θ the model parameters then the fully Bayesian criterion demands the full posterior*

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)}$$

6.1 EM-ML

It is well known to any frequentist bear that, denoting the latent variables z and introducing any distribution over latent variables, $q(z)$,

$$\begin{aligned}\log p(y|\theta) &\geq \sum_z q(z) \log \frac{p(y, z|\theta)}{q(z)} \\ &= E_{q(z)}[\log p(y, z|\theta)] + H[q(z)] \equiv F(q, \theta).\end{aligned}$$

The EM algorithm consists of maximizing a lower bound on $p(y|\theta)$ by iterating the following steps over time t:

$$\textbf{E-step:} \quad q^{(t+1)} = \operatorname*{argmax}_q F(q, \theta^{(t)})$$
$$\textbf{M-step:} \quad \theta^{(t+1)} = \operatorname*{argmax}_\theta F(q^{(t+1)}, \theta)$$

1. In the E-step, prove that $q^{(t+1)} = p(z|y, \theta^{(t)})$.
2. In class it was explained that the M-step updated parameters using expected counts. Show that in the E-step expected counts are computed using

$$E_{q(z)}[\text{Count}(\mathbf{A}_O = \mathbf{a}_O, \mathbf{A}_H = \mathbf{a}_H)] = \sum_{j=1}^{R} \mathbf{1}(\mathbf{A}_O^{(j)} = \mathbf{a}_O) P(\mathbf{A}_H = \mathbf{a}_H | O^{(j)}, \theta^{(t)})$$

   where $O^{(j)}$ is the j^{th} record in the data set, $\mathbf{A}_O$ are the observed variables, $\mathbf{A}_H$ the unobserved (latent) variables, and $\theta^{(t)}$ the estimate of the Bayesian network parameters at step t of the EM algorithm. $\mathbf{1}(\mathbf{A}_O^{(j)} = \mathbf{a}_O)$ is an indicator function for whether variables $\mathbf{A}_O$ take on value $\mathbf{a}_O$ is record j.
3. You are given two inference routines, one for variable elimination and another for junction trees. Which routine is more appropriate for the E-step in part 2 ? Briefly explain you answer.

6.2 EM-MAP

The father bear has scrawled down the following equation:

$$\log p(\theta|y) \geq E_{q(z)}[\log p(y, z|\theta)] + H[q(z)] + \log p(\theta) \equiv F(q, \theta) \tag{6.1}$$

where $q(z)$ is some distribution over unobserved variables z, which is typically called the variational free distribution, and $p(\theta)$ is a parameter prior. The EM algorithm consists of maximizing a lower bound on $p(\theta|y)$.

$$\textbf{E-step:} \quad q^{(t+1)} = \operatorname*{argmax}_q F(q, \theta^{(t)})$$
$$\textbf{M-step:} \quad \theta^{(t+1)} = \operatorname*{argmax}_\theta F(q^{(t+1)}, \theta)$$

1. Prove equation 6.1.

6.3 Fully Bayesian

The bears come home to find the little blonde girl rummaging through their desks. Regular bears would simply maul her. However, since these are not regular bears they chain the girl up and give her choice: answer the following questions or be eaten alive.

1. Briefly explain why computing $p(y)$ exactly is difficult ?
2. Assuming some free distribution that factors over latent variables and parameters, $q(z, \theta) = q(z)q(\theta)$, prove that
$$\log p(y) \geq \int q(\theta) \ln \frac{p(\theta)}{q(\theta)}\, d\theta + \int q(\theta) \sum_z q(z) \ln \frac{p(y, z|\theta)}{q(z)}\, d\theta \equiv F(q(z), q(\theta))$$
3. Prove that maximizing $F(q(z), q(\theta))$ corresponds to minimizing the KL-divergence $D(q(z, \theta)||p(z, \theta|y))$.

We can estimate the marginal likelihood $p(y)$ by maximizing the lower bound $F(q(z), q(\theta))$ with the following EM-style algorithm (which you do not need to prove):

$$\textbf{E-step:} \quad q^{(t+1)}(z) \propto \exp \int q^{(t)}(\theta) \ln p(y, z|\theta)\, d\theta$$
$$\textbf{M-step:} \quad q^{(t+1)}(\theta) \propto p(\theta) \exp \int \ln p(y, z|\theta) q^{(t+1)}(z)\, dz$$

When asked to derive this, the little blonde girl decided that she'd rather be eaten alive. The end.

7 MPE inference in MRFs with Graph-Cuts [Dhruv] [23 pts]

Recall that an MPE (Most Probable Explanation) query attempts to find the most probable assignment to all the non-evidence variables. More precisely, if $\mathcal{X}$ is a set of random variables, E is the set of evidence variables, and $W = \mathcal{X} - E$ is the set of query variables:

$$\text{MPE}\,(W \mid e) = \operatorname*{argmax}_w P(W = w \mid E = e). \tag{7.1}$$

Also recall that an MRF can be parametrized with energy functions:

$$P(\mathcal{X}) = \frac{1}{\mathcal{Z}} \prod \Phi_C(X_C) \tag{7.2}$$
$$= \frac{1}{\mathcal{Z}} \prod \exp(-\mathcal{E}_C(X_C)), \tag{7.3}$$

where $\mathcal{E}_C(X_C) = -\log \Phi_C(X_C)$ is the energy of the clique C. With this parametrization, the MPE problem can be rewritten as:

$$\operatorname*{argmax}_{\mathcal{X}} P(\mathcal{X}) = \operatorname*{argmax}_{\mathcal{X}} \log P(\mathcal{X}) \tag{7.4}$$
$$= \operatorname*{argmin}_{\mathcal{X}} \left(\sum \mathcal{E}_{\mathcal{C}} - \log \mathcal{Z} \right) \tag{7.5}$$
$$= \operatorname*{argmin}_{\mathcal{X}} \sum \mathcal{E}_{\mathcal{C}}, \tag{7.6}$$

where we can ignore $\mathcal{Z}$ in the last step because the partition function does not depend on the assignment of the variables. Thus we have reparametrized the MPE problem into that of energy minimization.

In this question, you will show that for a certain class of energy functions, MPE problem in binary-valued MRFs can be solved optimally using a very simple graph-cut algorithm. The most surprising part of this reduction is that the algorithm is guaranteed to return the optimal solution in polynomial time, regardless of the structural complexity of the underlying graph, which is in contrast to most inference methods we have seen so far, where polynomial-time solutions were obtainable only for graphs with low tree widths.

For simplicity, we will assume that the evidence variables have been instantiated and absorbed into the node potentials, and allow only pairwise cliques. Thus, if the MRF has graph structure $\mathcal{G} = (V, E)$, our energies are of the form:

$$\mathcal{E}(\mathcal{X}) = \sum_{i \in V} \mathcal{E}_i(X_i) + \sum_{(i,j) \in E} \mathcal{E}_{ij}(X_i, X_j) \tag{7.7}$$

In addition, we will assume that the energy function is *submodular*, which means:

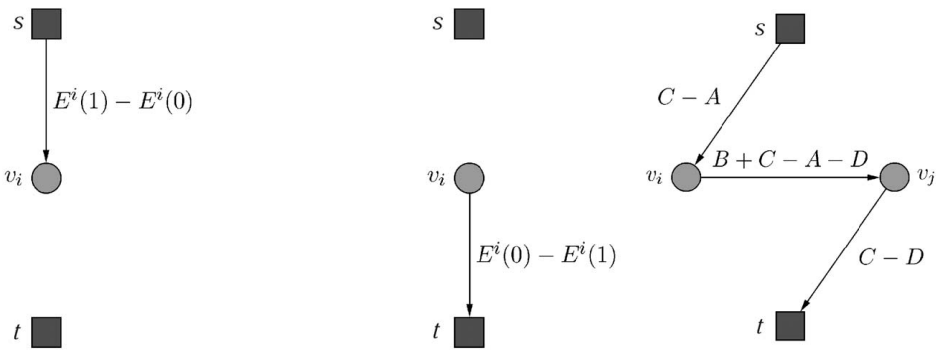
$$\mathcal{E}_{ij}(0,0) + \mathcal{E}_{ij}(1,1) \leq \mathcal{E}_{ij}(1,0) + \mathcal{E}_{ij}(0,1) \qquad \forall (i,j) \tag{7.8}$$

Review of min-cut. Before we proceed, here is a quick review of the min-cut problem. The s-t min-cut problem is defined for networks, which are directed graphs with a vertex set V, plus two special nodes called the source s, and the sink t. We have a set of directed edges E over $V \cup \{s, t\}$, where each edge is associated with a non-negative cost $cost(v_i, v_j)$. A *cut* is a disjoint partition of $V \cup \{s, t\}$ into $V_s \cup V_t$ such that $s \in V_s$, and $t \in V_t$. The cost of the cut is given by the sum of the cost on the edges originating in V_s and ending in V_t.

$$cost(V_s, V_t) = \sum_{v_i \in V_s, v_j \in V_t} cost(v_i, v_j) \tag{7.9}$$

The *min-cut* is the partition V_s, V_t that achieves the minimum cost. These edges (v_i, v_j) which originate in V_s and end in V_t are called cut-edges. Polynomial time algorithms for finding this min-cut exist which are based on max-flow computations.

So how do we reduce the MPE problem to one of computing min-cut on a graph? Intuitively, we need to design our graph so that a cut corresponds to an assignment of $\mathcal{X}$, and the cost of a cut corresponds to the energy of the corresponding assignment (plus perhaps a constant). We will construct a directed graph $\mathcal{G}' = (V', E')$, such every variable $X_i \in \mathcal{X}$ corresponds to a node v_i. In addition there are two special nodes, a source (s) and a sink (t). The source connects *to* all v_i (i.e., $(s, v_i) \in E', \forall i$), and the sink has an edge *from* all v_i (i.e., $(v_i, t) \in E', \forall i$).



(a) Construction 1 (b) Construction 2

Construction 1: Unary Energies Only. Assume for this part that there are no pairwise energies (i.e., $\mathcal{E}_{ij}(X_i, X_j) = 0,\ \forall(X_i, X_j)$). Consider unary energy $\mathcal{E}_i$ for a variable X_i. If $\mathcal{E}_i(0) < \mathcal{E}_i(1)$, then we put weight $\mathcal{E}_i(1) - \mathcal{E}_i(0)$ on edge (s, v_i), and weight 0 on edge (v_i, t). If $\mathcal{E}_i(0) > \mathcal{E}_i(1)$, then we put weight $\mathcal{E}_i(0) - \mathcal{E}_i(1)$ on edge (v_i, t), and weight 0 on edge (s, v_i). Otherwise, if $\mathcal{E}_i(0) = \mathcal{E}_i(1)$, we put weight 0 on both edges. Figure 4a shows this visually (zero weight edges have not been drawn).

1. Describe how a cut in this constructed graph would look like (structurally)? Specifically, can a node be connected to two cut-edges? Can a node be connected to zero cut-edges?

2. Using the intuition for what a cut looks like, describe how a cut in this graph ($\mathcal{G}'$) would correspond to an assignment of $\mathcal{X}$.

   (*Hint*: construct a mapping that would take a cut as input and produce an assignment of $\mathcal{X}$. Prove that this mapping is a bijection.)

3. Prove that the cost of a cut in the above constructed graph is equal to the energy of the corresponding assignment (plus possibly a constant that does not depend on the assignment of variables).

   (*Hint*: Use the fact that these energies can be shifted by a constant without affecting the MPE outcome. Thus you can assume that either $\mathcal{E}_i(0)$ or $\mathcal{E}_i(1)$ is 0, by subtracting the smaller of the two from both.)

Construction 2: Pairwise Energies. Consider a pairwise energy term $\mathcal{E}_{ij}$ for a pair of variables (X_i, X_j). We now show how the four parameters describing this pairwise energy (eqn 7.10) will be incorporated onto weights on edges in $\mathcal{G}'$. We notice that this pairwise energy can be broken into four terms as shown in equation 7.11. Note that the first term is a constant, the second term is a function of only X_i, the third term is a function of only X_j, and the fourth term is a function of both X_i and X_j.

$$\mathcal{E}_{ij} = \begin{array}{|c|c|}\hline \mathcal{E}_{ij}(0,0) & \mathcal{E}_{ij}(0,1) \\ \hline \mathcal{E}_{ij}(1,0) & \mathcal{E}_{ij}(1,1) \\ \hline \end{array} = \begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline \end{array} \tag{7.10}$$

$$\begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline \end{array} = A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & D-C \\ \hline 0 & D-C \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline \end{array} \tag{7.11}$$

Corresponding to each of the terms, we will add weights on edges in the graph.

- For the fourth term, we set the weight of edge (v_i, v_j) as $(B + C - A - D)$, and that of (v_j, v_i) to 0.
- For the second and third terms, we follow same rules as construction 1 by treating them as unary energies. If $C - A > 0$, we add a weight of $C - A$ on edge (s, v_i), otherwise we add a weight of $A - C$ on edge (v_i, t). If $D - C > 0$, we add a weight of $D - C$ on edge (s, v_j), otherwise we add a weight of $C - D$ on edge (v_j, t). For example, figure 4b shows the weights added on edges when $C > A$ and $C > D$.
- The first term (A) is ignored.

It should be noted that this contribution of pairwise energies would be added to any contribution from the unary energies. Thus in the current example, the final weight on (s, v_i) would be $\mathcal{E}_i(1) - \mathcal{E}_i(0) + C - A$.

Finally, here are your questions:

4. Extend your previous solution to include pairwise energies. Specifically, show that the cost of a cut on the graph constructed by the above construction is equal to the energy of the corresponding assignment (plus possibly a constant that does not depend on the assignment of variables).
5. In construction 2 why did we not add any edges for the first term in equation 7.11 (i.e., A)?
6. Using the intuition developed by the past couple of questions, provide a polynomial-time algorithm for performing MPE inference queries in binary MRFs with submodular energies. (You can use a min-cut algorithm as a black-box).
7. We know from class that the MPE problem in general graphs is NP-complete. Yet you were able to provide a polytime algorithm in the above part. Why?
8. **Extra Credit:** Can this method be extended to non-binary MRFs? If each variable X_i were a K-ary variable, can we still construct a graph where a cut corresponds to an assignment of $\mathcal{X}$? Provide either an algorithm for MPE inference in K-ary MRFs, or comment why finding such an algorithm is hard.

8 Feedback [0 pts]

The following are questions that we use to calibrate the exam in future years. Your answers are appreciated.

1. How many hours did it take to complete the exam ?
2. Which two questions did you find hardest ?
3. Which two questions did you find easiest ?