CS 6780: Advanced Machine Learning
Homework 4

Due date: 12pm. **Dec 3, 2010** (Printed submission.)

Note:
1. Write your name and net-id on the first page.
2. For derivation questions, please show all the necessary steps.
3. For data analysis / programming parts, you need to submit: (a) the specific values and plots requested, (b) explanation in the text if needed (don't give us the code printouts only!), and (c) the code in the appendix.

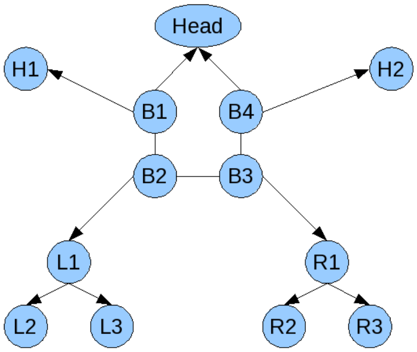
1 Basics of Graphical Models

(a) [3 points] Provide an example of a distribution $P(X_1, X_2, X_3)$ where for each $i \neq j$, we have that $(X_i \perp X_j)$ is true, but we also have that $(X_1, X_2 \perp X_3)$ is not true.
(b) [2 points] Give an example of a joint distribution over (X_1, X_2) (belonging to exponential family) for which the $\mathrm{Cov}(X_1, X_2) = 0$, but $X_1 \perp X_2$ is not true.
(c) [10 points] Let G_1 and G_2 be two graphs over X. If G_1 and G_2 have the same skeleton and the same set of v-structures then prove that they represent same independence assumptions.

2 Junction Tree

(a) [10 points] Eve is now looking for WallE using her cameras, but could not find WallE. Help her out by helping in building a WallE classifier!

Eve has small circuits for performing sum-product, etc. We know that even any graph can be converted into a "tree" — junction tree. So, your goal



is to design a **junction-tree** of the graph Eve has in her mind for WallE.[1] (10 points)

(b) [10 points]

In the juction-tree, start from the head node as 'root' node, and write the sum-product messages coming in from all the leaves to the root, and then from the root to all the leaves. (In order to save time in writing, you can assume the hands/legs of the WallE are symmetric, so you can just write it for one side, i.e., H1, all Bs, Head and all Ls.)

(c) [10 points]

Do part (b) for max-sum messages for leaf-to-root direction.

Read Bishop ch. 8, and explain concisely in a line what would you do for root-to-leaf messages.

3 Kalman Filter

(a) [5 points] One of the most computation intensive part in Kalman filter is inversion of matrices. For a state of size 10 and measurement of size 5, what is maximum size of the matrix one has to invert.

(b) [10 points] Consider the version of the Kalman filter described in the class (where everything is Gaussian and there is no control input). Now, lets change the prior distribution $p(x_0)$, i.e., initial state from Gaussian to a mixture of K Gaussians. Does the inference in Kalman filter remains

[1] Loopy methods are not being used here, because if they don't converge Eve might not be able to find WallE.

tractable[2] ? Explain concisely.

(c) [10 points] Now, we model the state update error ϵ, $p(\epsilon)$ using a mixture of L Gaussians. Does it remain tractable? Explain concisely.

4 Markov Random Field

(a) [12 points] In some applications such as stereo vision and image denoising, we deal with continuous variables. Gaussian densities are first ones that one could try.

Given a grid-MRF (e.g., each node is connected to the top, left, right and bottom node), each node $y_i \in \Re$ represents what we want to predict, and $x_i \in \Re^K$ are the input features for that pixel. For n nodes, we will have $i = 1, ..., n$, and use y to represent the set $y = \{y_1, ..., y_n\}$.

Our model is (similar to Saxena, Chung, Ng, NIPS 2005):

$$p(y|x;\theta) \propto \prod_{i=1}^{n} \exp(-(y_i - x_i^T\theta)^2/2\sigma_1^2) \prod_{j \in N(i)} \exp(-(y_i - y_j)^2/2\sigma_2^2)$$

where $N(i)$ are the neighbors of i.

During inference (testing) phase, we are given $\theta, \sigma_1^2, \sigma_2^2$ and x_i's, and we have to calculate $y^* = \arg\max_y p(y|x;\theta,\sigma_1,\sigma_2)$. Write the above maximization as a convex program (specifically a quadratic program).

(b) [18 points] In some other applications, such as image segmentation (where each pixel can belong to one of a few classes), we deal for discrete variables.

Given a grid-MRF (e.g., each node is connected to the top, left, right and bottom node), each node $y_i \in \{-1, 1\}$ represents what we want to predict, and $x_i \in \Re^K$ are the input features for that pixel. For n nodes, we will have $i = 1, ..., n$, and use y to represent the set $y = \{y_1, ..., y_n\}$.

Our model is (similar to Kumar and Hebert, 2004):

$$p(y|x;\theta) \propto \prod_{i=1}^{n} p_s(y_i|x_i;\theta) \prod_{j \in N(i)} \exp(\beta y_i y_j)$$

[2] Here, tractable means that the exact inference does not grows (exponentially) with time.

where $N(i)$ are the neighbors of i, and

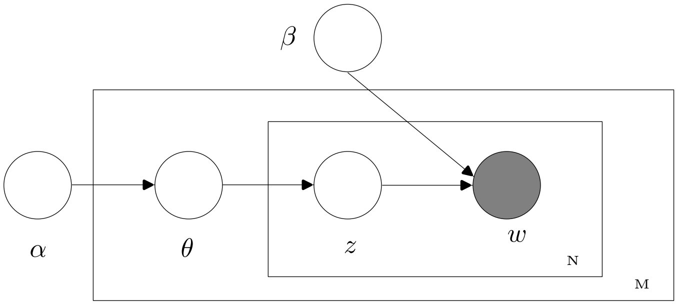
$$p_s(y_i|x_i;\theta) \propto \frac{1}{1+\exp(-\theta^T x_i y_i)}$$

.

Write code that can compute $y^* = \arg\max_y p(y|x;\theta)$, for $\beta = 0.9$, and for values of θ and x_i given in file `discretemrf-cs6780.txt`. Here we use tables to store the probabilities.
Bonus: Write concisely what would you do when you're given a grid of size 30x30 instead of 3x3.

5 Bonus/Optional: Latent Dirichlet allocation

Latent Dirichlet Allocation, or LDA, (Blei et. al. 2003) is a model for discovering topics in a collection of documents. The graphical model we will work with is as follows:



Here's how the data are generated according to the model:

- For each document, m = 1,...,M
  - Draw topic probabilities $\theta_m \sim p(\theta|\alpha)$
  - For each of the N words:
    - Draw a topic $z_{mn} \sim p(z|\theta_m)$
    - Draw a word $w_{mn} \sim p(w|z_{mn}, \beta)$

where $p(\theta|\alpha)$ is a Dirichlet distribution, and where $p(z|\theta_m)$ and $p(w|z_{mn}, \beta)$ are multinomial distributions. Treat α and β as fixed hyperparameters. Note that β is a matrix, with one column per topic, and the multinomial variable z_{mn} selects one of the columns of β to yield multinomial probabilities for w_{mn}. (See the paper "Latent Dirichlet Allocation" by Blei et. al. on the course website for more details if needed). Write down a Gibbs sampler for the LDA model. That is, write down the conditional probabilities of z and θ given their Markov blankets so that we can sample from these distributions. (20 points)