Probabilistic Graphical Models 10-708
Homework 1: Due January 29, 2014 at 4 pm

Directions. This homework assignment covers the material presented in Lectures 1-3. You must complete all four problems to obtain full credit. To submit your assignment, please upload a pdf file containing your writeup and a zip file containing your code to Canvas by 4 pm on Wednesday, January 29th. We highly encourage that you type your homework using the LaTeXtemplate provided on the course website, but you may also write it by hand and then scan it.

1 Fundamentals [25 points]

This question will refer to the graphical models shown in Figures 1 and 2, which encode a set of independencies among the following variables: Season (S), Flu (F), Dehydration (D), Chills (C), Headache (H), Nausea (N), Dizziness (Z). Note that the two models have the same skeleton, but Figure 1 depicts a directed model (Bayesian network) whereas Figure 2 depicts an undirected model (Markov network).

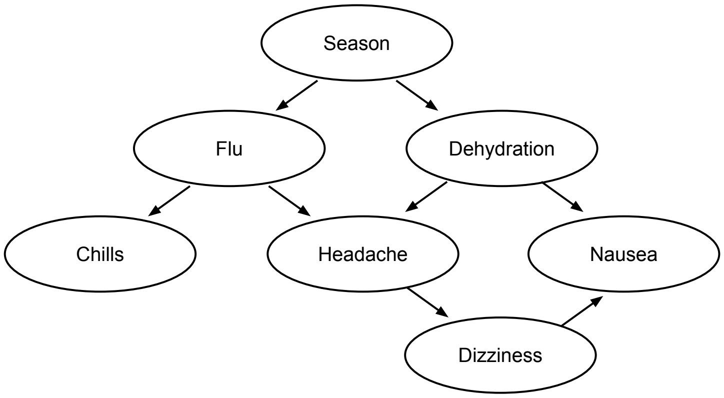

Figure 1: A Bayesian network that represents a joint distribution over the variables Season, Flu, Dehydration, Chills, Headache, Nausea, and Dizziness.

Part 1: Independencies in Bayesian Networks [12 points]

Consider the model shown in Figure 1. Indicate whether the following independence statements are true or false according to this model. Provide a very brief justification of your answer (no more than 1 sentence).

1. Season $\perp$ Chills
2. Season $\perp$ Chills | Flu

3. Season ⊥ Headache | Flu
4. Season ⊥ Headache | Flu, Dehydration
5. Season ⊥ Nausea | Dehydration
6. Season ⊥ Nausea | Dehydration, Headache
7. Flu ⊥ Dehydration
8. Flu ⊥ Dehydration | Season, Headache
9. Flu ⊥ Dehydration | Season
10. Flu ⊥ Dehydration | Season, Nausea
11. Chills ⊥ Nausea
12. Chills ⊥ Nausea | Headache

Part 2: Factorized Joint Distributions [4 points]

1. Using the directed model shown in Figure 1, write down the factorized form of the joint distribution over all of the variables, $P(S, F, D, C, H, N, Z)$.
2. Using the undirected model shown in Figure 2, write down the factorized form of the joint distribution over all of the variables, assuming the model is parameterized by one factor over each node and one over each edge in the graph.

$P(S = \text{winter})$	$P(S = \text{summer})$
0.5	0.5

	$P(F = \text{true} \mid S)$	$P(F = \text{false} \mid S)$
$S = \text{winter}$	0.4	0.6
$S = \text{summer}$	0.1	0.9

	$P(D = \text{true} \mid S)$	$P(D = \text{false} \mid S)$
$S = \text{winter}$	0.1	0.9
$S = \text{summer}$	0.3	0.7

	$P(C = \text{true} \mid F)$	$P(C = \text{false} \mid F)$
$F = \text{true}$	0.8	0.2
$F = \text{false}$	0.1	0.9

	$P(H = \text{true} \mid F, D)$	$P(H = \text{false} \mid F, D)$
$F = \text{true}$, $D = \text{true}$	0.9	0.1
$F = \text{true}$, $D = \text{false}$	0.8	0.2
$F = \text{false}$, $D = \text{true}$	0.8	0.2
$F = \text{false}$, $D = \text{false}$	0.3	0.7

	$P(Z = \text{true} \mid H)$	$P(Z = \text{false} \mid H)$
$H = \text{true}$	0.8	0.2
$H = \text{false}$	0.2	0.8

	$P(N = \text{true} \mid D, Z)$	$P(N = \text{false} \mid D, Z)$
$D = \text{true}$, $Z = \text{true}$	0.9	0.1
$D = \text{true}$, $Z = \text{false}$	0.8	0.2
$D = \text{false}$, $Z = \text{true}$	0.6	0.4
$D = \text{false}$, $Z = \text{false}$	0.2	0.8

Table 1: Conditional probability tables for the Bayesian network shown in Figure 1.

Part 3: Evaluating Probability Queries [7 points]

Assume you are given the conditional probability tables listed in Table 1 for the model shown in Figure 1. Evaluate each of the probabilities queries listed below, and show your calculations.

1. What is the probability that you have the flu, when no prior information is known?
2. What is the probability that you have the flu, given that it is winter?
3. What is the probability that you have the flu, given that it is winter and that you have a headache?
4. What is the probability that you have the flu, given that it is winter, you have a headache, and you know that you are dehydrated?
5. Does knowing you are dehydrated increase or decrease your likelihood of having the flu? Intuitively, does this make sense?

Part 4: Bayesian Networks vs. Markov Networks [2 points]

Now consider the undirected model shown in Figure 2.

1. Are there any differences between the set of marginal independencies encoded by the directed and undirected versions of this model? If not, state the full set of marginal independencies encoded by both models. If so, give one example of a difference.
2. Are there any differences between the set of conditional independencies encoded by the directed and undirected versions of this model? If so, give one example of a difference.

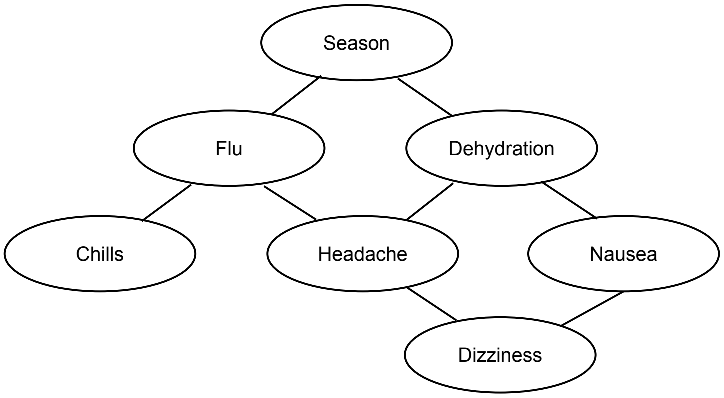

Figure 2: A Markov network that represents a joint distribution over the variables Season, Flu, Dehydration, Chills, Headache, Nausea, and Dizziness.

2 Bayesian Networks [25 points]

Part 1: Constructing Bayesian Networks [8 points]

In this problem you will construct your own Bayesian network (BN) for a few different modeling scenarios described as word problems. By standard convention, we will use shaded circles to represent observed quantities, clear circles to represent random variables, and uncircled symbols to represent distribution parameters.

In order to do this problem, you will first need to understand plate notation, which is a useful tool for drawing large BNs with many variables. Plates can be used to denote repeated sets of random variables. For example, suppose we have the following generative process:

- Draw $Y \sim \text{Normal}(\mu, \Sigma)$
- For $m = 1, \ldots, M$:

  Draw $X_m \sim \text{Normal}(Y, \Sigma)$

This BN contains $M + 1$ random variables, which includes M repeated variables $X_1, \ldots, X_M$ that all have Y as a parent. In the BN, we draw the repeated variables by placing a box around a single node, with an index in the box describing the number of copies; we've drawn this in Figure 3.

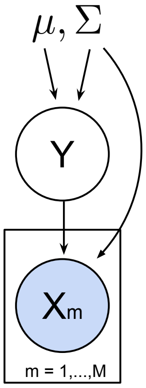

Figure 3: An example of a Bayesian network drawn with plate notation.

For each of the modeling scenarios described below, draw a corresponding BN. Make sure to label your nodes using the variable names given below, and use plate notation if necessary.

1. (Gaussian Mixture Model). Suppose you want to model a set of clusters within a population of N entities, $X_1, \ldots, X_N$. We assume there are K clusters $\theta_1, \ldots, \theta_K$, and that each cluster represents a vector and a matrix, $\theta_k = \{\mu_k, \Sigma_k\}$. We also assume that each entity X_n "belongs" to one cluster, and its membership is given by an assignment variable $Z_n \in \{1, \ldots, K\}$.

   Here's how the variables in the model relate. Each entity X_n is drawn from a so-called "mixture distribution," which in this case is a Gaussian distribution, based on its individual cluster assignment and the entire set of clusters, written $X_n \sim \text{Normal}(\mu_{Z_n}, \Sigma_{Z_n})$. Each cluster assignment Z_n has a prior, given by $Z_n \sim \text{Categorical}(\beta)$. Finally, each cluster θ_k also has a prior, given by $\theta_k \sim \text{Normal-invWishart}(\mu_0, \lambda, \Phi, \nu) = \text{Normal}(\mu_0, \frac{1}{\lambda}\Sigma) \cdot \text{invWishart}(\Phi, \nu)$.

2. (Bayesian Logistic Regression). Suppose you want to model the underlying relationship between a set of N input vectors $X_1, \ldots, X_N$ and a corresponding set of N binary outcomes $Y_1, \ldots, Y_N$. We assume there is a single vector β which dictates the relationship between each input vector and its associated output variable.

   In this model, each output is drawn with $Y_n \sim \text{Bernoulli}(\text{invLogit}(X_n\beta))$. Additionally, the vector β has a prior, given by $\beta \sim \text{Normal}(\mu, \Sigma)$.

Part 2: Inference in Bayesian Networks [12 points]

In this problem you will derive formulas for inference tasks in Bayesian networks. Consider the Bayesian network given in Figure 4.

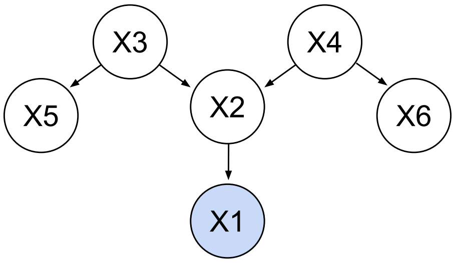

Figure 4: A Bayesian network over the variables $X_1, ..., X_6$. Note that X_1 is observed (which is denoted by the fact that it's shaded in) and the remaining variables are unobserved.

For each of the following questions, write down an expression involving the variables $X_1, \dots, X_6$ that could be computed by directly plugging in their local conditional probability distributions.

First, give expressions for the following three posterior distributions over a particular variable given the observed evidence $X1 = x1$.

1. $P(X_2 = x_2|X_1 = x_1)$
2. $P(X_3 = x_3|X_1 = x_1)$
3. $P(X_5 = x_5|X_1 = x_1)$

Second, give expressions for the following three conditional probability queries. Note that these types of expressions are useful for the inference algorithms that we'll learn later in the class.

4. $P(X_2 = x_2|X_1 = x_1, X_3 = x_3, X_4 = x_4, X_5 = x_5, X_6 = x_6)$
5. $P(X_3 = x_3|X_1 = x_1, X_2 = x_2, X_4 = x_4, X_5 = x_5, X_6 = x_6)$
6. $P(X_5 = x_5|X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = x_4, X_6 = x_6)$

Part 3: On Markov Blankets [5 points]

In this problem you will prove a key property of Markov blankets in Bayesian networks. Recall that the Markov blanket of a node in a BN consists of the node's children, parents, and coparents (i.e. the children's other parents). Also recall that there are four basic types of two-edge trails in a BN, which are illustrated in Figure 5: the causal trail (head-to-tail), evidential trail (tail-to-head), common cause (tail-to-tail), and common effect (head-to-head).

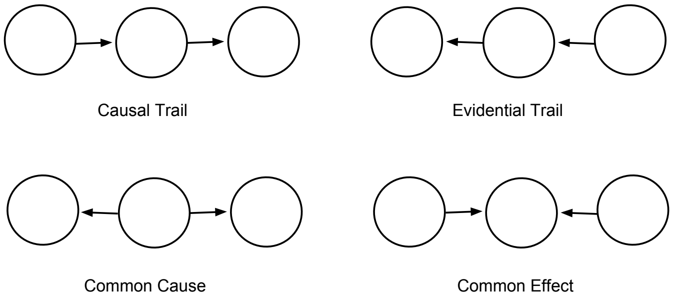

Figure 5: Illustration of the four basic types of two-edge trails in a BN.

Using the four trail types, prove the following property of BNs: *given its Markov blanket, a node in a Bayesian network is conditionally independent of every other set of nodes.*

3 Restricted Boltzmann Machines [25 points]

Restricted Boltzmann Machines (RBMs) are a class of Markov networks that have been used in several applications, including image feature extraction, collaborative filtering, and recently in deep belief networks. An RBM is a bipartite Markov network consisting of a visible (observed) layer and a hidden layer, where each node is a binary random variable. One way to look at an RBM is that it models latent factors that can be learned from input features. For example, suppose we have samples of binary user ratings (like vs. dislike) on 5 movies: Finding Nemo (V_1), Avatar (V_2), Star Trek (V_3), Aladdin (V_4), and Frozen (V_5). We can construct the following RBM:

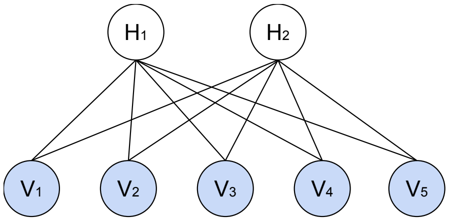

Figure 6: An example RBM with 5 visible units and 2 hidden units.

Here, the bottom layer consists of visible nodes $V_1, ..., V_5$ that are random variables representing the binary ratings for the 5 movies, and H_1, H_2 are two hidden units that represent latent factors to be learned during training (e.g., H_1 might be associated with Disney movies, and H_2 could represent the adventure genre). If we are using an RBM for image feature extraction, the visible layer could instead denote binary values associated with each pixel, and the hidden layer would represent the latent features. However, for this problem we will stick with the movie example. In the following questions, let $V = (V_1, ..., V_5)$ be a vector of ratings (e.g. the observation $v = (1, 0, 0, 0, 1)$ implies that a user likes only Finding Nemo and Aladdin). Similarly, let $H = (H_1, H_2)$ be a vector of latent factors. Note that all the random variables are binary and take on states in $\{0, 1\}$. The joint distribution of a configuration is given by

$$p(V = v, H = h) = \frac{1}{Z} e^{-E(v,h)} \tag{1}$$

where

$$E(v, h) = -\sum_{ij} w_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j$$

is the energy function, $\{w_{ij}\}, \{a_i\}, \{b_j\}$ are model parameters, and

$$Z = Z(\{w_{ij}\}, \{a_i\}, \{b_i\}) = \sum_{v,h} e^{-E(v,h)}$$

is the partition function, where the summation runs over all joint assignments to V and H.

1. [7 pts] Using Equation (1), show that $p(H|V)$, the distribution of the hidden units conditioned on all of the visible units, can be factorized as

$$p(H|V) = \prod_j p(H_j|V) \tag{2}$$

where

$$p(H_j = 1|V = v) = \sigma\left(b_j + \sum_i w_{ij} v_i\right)$$

and $\sigma(s) = \frac{e^s}{1+e^s}$ is the sigmoid function. Note that $p(H_j = 0|V = v) = 1 - p(H_j = 1|V = v)$.

2. [3 pts] Give the factorized form of $p(V|H)$, the distribution of the visible units conditioned on all of the hidden units. This should be similar to what's given in part 1, and so you may omit the derivation.

3. [2 pts] Can the marginal distribution over hidden units $p(H)$ be factorized? If yes, give the factorization. If not, give the form of $p(H)$ and briefly justify.

4. [4 pts] Based on your answers so far, does the distribution in Equation (1) respect the conditional independencies of Figure (6)? Explain why or why not. Are there any independencies in Figure 6 that are not captured in Equation (1)?

5. [7 pts] We can use the log-likelihood of the visible units, $\log p(V = v)$, as the criterion to learn the model parameters $\{w_{ij}\}, \{a_i\}, \{b_j\}$. However, this maximization problem has no closed form solution. One popular technique for training this model is called "contrastive divergence" and uses an approximate gradient descent method. Compute the gradient of the log-likelihood objective with respect to w_{ij} by showing the following:

$$\begin{aligned}\frac{\partial \log p(V = v)}{\partial w_{ij}} &= \sum_h p(H = h|V = v) v_i h_j - \sum_{v,h} p(V = v, H = h) v_i h_j \\ &= \mathbb{E}\left[V_i H_j | V = v\right] - \mathbb{E}\left[V_i H_j\right]\end{aligned}$$

   where $\mathbb{E}\left[V_i H_j | V = v\right]$ can be readily evaluated using Equation (2), but $\mathbb{E}\left[V_i H_j\right]$ is tricky as the expectation is taken over not just H_j but also V_i.

   Hint 1: To save some writing, do not expand $E(v, h)$ until you have $\frac{\partial E(v,h)}{\partial w_{ij}}$.

   Hint 2: The partition function, Z, is a function of w_{ij}.

6. [2 pts] After training, suppose $H_1 = 1$ corresponds to Disney movies, and $H_2 = 1$ corresponds to the adventure genre. Which w_{ij} do you expect to be positive, where i indexes the visible nodes and j indexes the hidden nodes? List all of them.

4 Image Denoising [25 points]

This is a programming problem involving Markov networks (MNs) applied to the task of image denoising. Suppose we have an image consisting of a 2-dimensional array of pixels, where each pixel value Z_i is binary, i.e. $Z_i \in \{+1, -1\}$. Assume now that we make a noisy copy of the image, where each pixel in the image is flipped with 10% probability. A pixel in this noisy image is denoted by X_i. We show the original image and the image with 10% noise in Figure 7.

Given the observed array of noisy pixels, our goal is to recover the original array of pixels. To solve this problem, we model the original image and noisy image with the following MN. We have a latent variable Z_i for each noise-free pixel, and an observed variable X_i for each noisy pixel. Each variable Z_i has an edge leading to its immediate neighbors (to the Z_i associated with pixels above, below, to the left, and to the right, when they exist). Additionally, each variable Z_i has an edge leading to its associated observed pixel X_i. We illustrate this MN in Figure 8.

Denote the full array of latent (noise-free) pixels as $\mathbf{Z}$ and the full array of observed (noisy) pixels as $\mathbf{X}$. We define the energy function for this model as

$$E(\mathbf{Z} = \mathbf{z}, \mathbf{X} = \mathbf{x}) = h \sum_i z_i - \beta \sum_{\{i,j\}} z_i z_j - \nu \sum_i z_i x_i \tag{3}$$

where the first and third summations are over the entire array of pixels, the second summation is over all pairs of latent variables connected by an edge, and $h \in \mathbb{R}$, $\beta \in \mathbb{R}_+$, and $\nu \in \mathbb{R}_+$ denote constants that must be chosen.

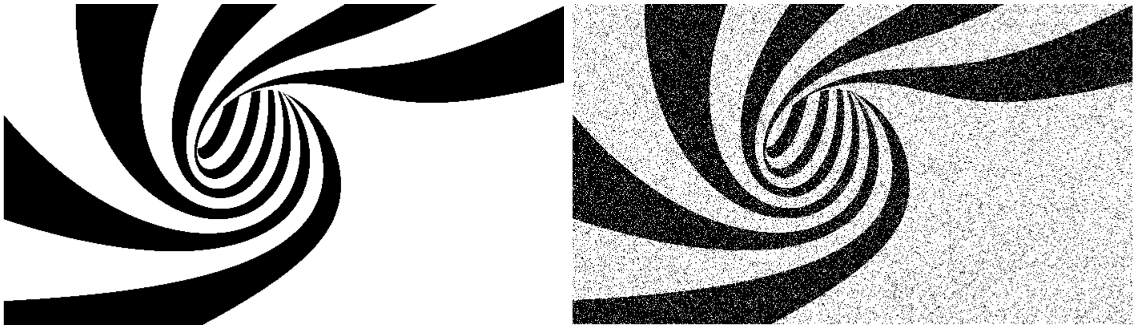

Figure 7: The original binary image is shown on the left, and a noisy version of the image in which a randomly selected 10% of the pixels have been flipped is shown on the right.

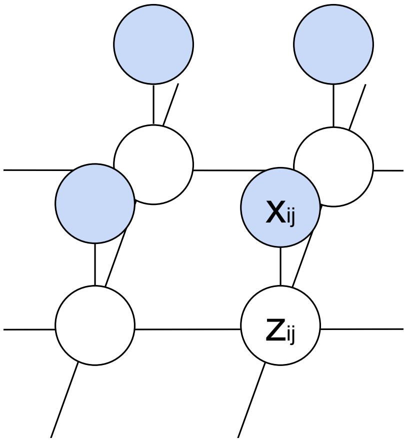

Figure 8: Illustration of the Markov network for image denoising.

Using the binary image data saved in `hw1_images.mat`, your task will be to infer the true value of each pixel (+1 or −1) by optimizing the above energy function. To do this, initialize the Z_i's to their noisy values, and then iterate through each Z_i and check whether setting it's value to +1 or −1 yields a lower energy (higher probability). Repeat this process, making passes through all of the pixels, until the total energy of the model has converged. You must specify values of the constants $h \in \mathbb{R}$, $\beta \in \mathbb{R}_+$, and $\nu \in \mathbb{R}_+$.

Report the error rate (fraction of pixels recovered incorrectly) that you achieve by comparing your denoised image to the original image that we provide, for three different settings of the three constants. Include a figure of your best denoised image in your writeup. Also make sure to submit a zipped copy of your code. The TAs will give a "special prize" to the student who is able to achieve the lowest error on this task.

Hint 1: When evaluating whether +1 or −1 is a better choice for a particular pixel Z_i, you do not need to evaluate the entire energy function, as this will be computationally very expensive. Instead, just compute the contribution by the terms that are affected by the value of Z_i.

Hint 2: If you'd like to try and compete to achieve the best performance, you can work to find good parameters, or even modify the algorithm in an intelligent way (be creative!). However, if you come up with a modified algorithm, you should separately report the new error rate you achieve, and also turn in a second .m file (placed in your zipped code directory) with the modified algorithm.