ECE521: Inference Algorithms and Machine Learning
University of Toronto

Assignment 4:
Inference and Learning on Graphical Models

TA: Jimmy Ba jimmy@psi.toronto.edu
Due date: April 7 midnight, 2016

General Note:

- In this assignment, you will implement learning and inference procedures for some of the probabilistic models described in class, apply your solutions to some simulated datasets, and analyze the results.
- Full points are given for complete solutions, including justifying the choices or assumptions you made to solve the question. Both complete source code and program outputs should be included in the final submission.
- Homework assignments are to be solved in the assigned groups of two or three. You are encouraged to discuss the assignment with other students, but you must solve it within your own group. Make sure to be closely involved in all aspects of the assignment.

1 Graphical Models [20 pt.]

1.1 Graphical models from factorization [6 pt.]

Consider a joint distribution that factors in the following form:

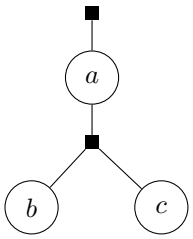
$$P(a, b, c, d, e, f) = P(a|b)P(b)P(c|a, b)P(d|b)P(e|c)P(f|b, e)$$

1. Sketch the corresponding Bayesian network (BN). [2 pt.]
2. Sketch the factor graph representation and label the factors with corresponding distributions. [2 pt.]
3. Sketch the Markov random field (MRF) representation and label the cliques with corresponding distributions. [2 pt.]

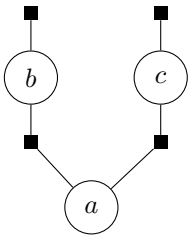
1.2 Conversion between graphical models [10 pt.]

1.2.1 [4 pt.]

a)



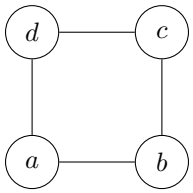
b)



1. For both factor graphs (a) and (b), if it exists, show the equivalent BNs that implies the **same conditional independence properties** as the factor graphs? [2 pt.]
2. For both factor graphs (a) and (b), if it exists, show the equivalent MRFs that implies the **same conditional independence properties** as the factor graphs? [2 pt.]

1.2.2 [4 pt.]

Consider the following MRF:

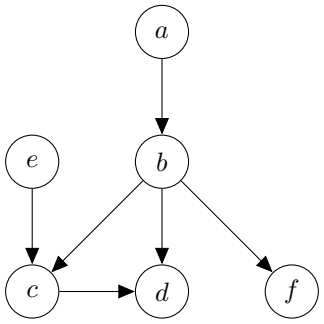


1. If it exists, show the equivalent factor graph representation. [2 pt.]
2. If it exists, show the equivalent BN. [2 pt.]

1.2.3 [2 pt.]

1. Construct an example of a BN that does not have any equivalent MRF with the same conditional independence properties. [2 pt.]

1.3 Conditional Independence in Bayesian Networks [4 pt.]



1. Express the joint probability $P(a, b, c, d, e, f)$ in a factorized form corresponding to the BN shown. [1 pt.]
2. Determine whether each of the followings statements is TRUE or FALSE and provide your explanations: $a \perp\!\!\!\perp c$? $a \perp\!\!\!\perp c|b$? $e \perp\!\!\!\perp b$? $e \perp\!\!\!\perp b|c$? $a \perp\!\!\!\perp e$? $a \perp\!\!\!\perp e|c$? [3 pt.]

2 Factor Analysis [8 pt.]

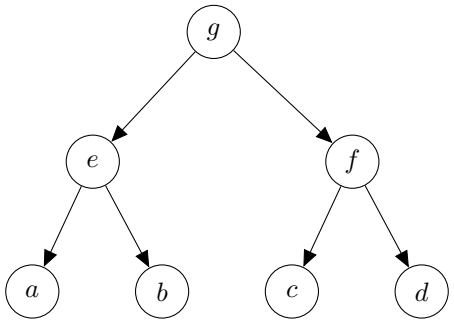
A Mixture of Gaussians (MoG) can be interpreted as a probabilistic treatment to K-means clustering. Factor analysis is a probabilistic model describe variability among observed data, that is an analogous probabilistic treatment to principal component analysis (PCA). In factor analysis, the D-dimensional observed variable $\mathbf{x} \in \mathbb{R}^D$ is modeled as a linear combination of the potential factors $\mathbf{z} \in \mathbb{R}^K$. Intuitively, it is a way of determining underlying variance in a given dataset.

Formally, the prior distribution over the latent factors is a K-dimensional Gaussian distribution $P(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$, where $\mathbf{I}$ denote an identity matrix. Given the latent factors $\mathbf{z}$, the observed variable $\mathbf{x}$ are modeled using a D-dimensional multivariate Gaussian distribution $P(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(\mathbf{x}; W\mathbf{x}, \sigma^2\mathbf{I})$, where σ is a scalar. The matrix $W \in \mathbb{R}^{D\times K}$ relates the latent factors to the observed data. The motivation here is that, for $K << D$, the latent factors will offer a more compact explanation of the dependencies between the observations. (You can learn more about Factor Analysis at https://www.metacademy.org/graphs/concepts/factor_analysis.) You may find *Chap. 12, Pattern Recognition and Machine Learning, Bishop 2007* useful for this question.

1. Sketch a BN representing the factor analysis model. [1 pt.]
2. Derive the multivariate posterior distribution over the latent factors $P(\mathbf{z}|\mathbf{x})$ in terms of $\mathbf{z}, \mathbf{x}, W, \sigma^2$. Which family of distributions does the posterior distribution belong to? [3 pt.]
3. Derive the multivariate marginal likelihood over the observed variable $P(\mathbf{x})$ in terms of $\mathbf{z}, \mathbf{x}, W, \sigma^2$. Which family of distributions does the marginal likelihood belong to? [4 pt.]

3 Message-Passing [8 pt.]

Consider a tree structure BN below, which represents a crude statistical model of genes mutations from the parent nodes to the children nodes.



We will focus on the probability of observing two specific genes. The presences of the two genes are denoted using a pair of binary random variables. Thus, the sample space or the set of the possible observations is $\{00, 01, 10, 11\}$ for all the random variables node in the BN. Let $P(g = 00) = 0.5$ and $P(g = 11) = 0.5$. Each gene has 10% chances of changing its state when passed down from a parent node. For example, $P(f = 11 \,|\, g = 00) = 0.01$ and $P(f = 01 \,|\, g = 11) = 0.09$.

1. Sketch the factor graph representing the BN. [1 pt.]
2. Calculate the numerical values of the probabilities $P(e)$ and $P(e|a = 01)$ using sum-product message-passing rules. Show the intermediate steps for computing the local messages. [7 pt.]

4 Hidden Markov Models [22 pt.]

Hidden Markov Model (HMM) is a class of probabilistic generative models for sequence data. Similar to Mixtures of Gaussians, HMM also has a set of latent mixture components. In addition, the latent states in HMM evolve over time to capture the temporal dependencies in the observed sequences. Consider a dataset of B sequences of observations and each has length T. For the n^{th} data point, the observed variable $x^{(n)} = \{x_1^{(n)}, x_2^{(n)}, x_3^{(n)}, \cdots, x_T^{(n)}\}$ is a sequence of T elements. HMM uses the latent variables $z_t^{(n)}$ to represent the hidden state assignments for each time t. An important difference between HMMs and MoGs is that the latent states in HMMs are related through shared transition probabilities, $P(z_t^{(n)} \,|\, z_{t-1}^{(n)})$ for each time t. The HMM represents the factorized joint distribution: $P(x_1^{(n)}, x_2^{(n)}, \cdots, x_T^{(n)}, z_1^{(n)}, z_2^{(n)}, \cdots, z_T^{(n)}) = P(z_1^{(n)}) \prod_{t=2}^{T} P(z_t^{(n)} \,|\, z_{t-1}^{(n)}) \prod_{t=1}^{T} P(x_t^{(n)} \,|\, z_t^{(n)})$. We will implement the learning and the inference algorithms for HMMs, and train HMM models on a toy DNA dataset. (You can learn more about HMMs at https://www.metacademy.org/graphs/concepts/hidden_markov_models.)

4.1 Factor graph representation [2 pt.]

1. Sketch a BN representing a HMM of over five observed variables in a sequence $\{x_1, x_2, x_3, x_4, x_5\}$. Label the latent state variables $\{z_1, z_2, z_3, z_4, z_5\}$. [1 pt.]

2. Sketch the factor graph representation for the BN above. Annotate each of the factors using either the prior $P(z_1)$, the transition probabilities $P(z_t \mid z_{t-1})$ and the likelihoods $P(x_t \mid z_t)$. [1 pt.]

4.2 Inference through message-passing [6 pt.]

We define the factor between variable nodes a and b as $f_{ab}(a, b)$, e.g. the factor between x_3 and z_3 is $f_{x_3 z_3}(x_3, z_3)$. In a message-passing scheme, messages from node c to node d are written as $\mu_{c \to d}$. The local messages of a node in a factor graph include all of its incoming and outgoing messages from the neighboring nodes. We continue the example from Section 4.1 and derive the sum-product algorithm for the HMM:

1. Write down the message-passing rule to compute the message from the variable node z_4 to the factor node $f_{z_3 z_4}$, that is $\mu_{z_4 \to f_{z_3 z_4}}(z_4)$, in terms of other local messages at z_4. for the posterior distribution $P(z_3 \mid x_1, x_2, x_3, x_4, x_5)$. [2 pt.]

2. Write down the message-passing rule to compute the posterior distribution $P(z_3 \mid x_1, x_2, x_3, x_4, x_5)$ in terms of the local messages at z_3. [2 pt.]

3. We expand the HMM to a new variable x_6 in the sequence that has not been observed yet. x_6 has its latent state z_6. We can make predictions about x_6 given the past observations by computing the predictive distribution $P(x_6 \mid x_1, x_2, x_3, x_4, x_5)$. Write down the message-passing rule for the predictive distribution in terms of $P(z_3 \mid x_1, x_2, x_3, x_4, x_5)$ [2 pt.]

4.3 Message-passing as bi-direction RNNs [4 pt.]

Consider an HMM with K latent states and observed variables may take on D discrete values $\{1, 2, \cdots, D\}$. Under a particular latent state $z_t = k$, each observation value may occur with $P(x_t = d \mid z_t = k)$. We represent the likelihood function $P(x_t \mid z_t)$ concisely in a $D \times K$ matrix W, where $P(x_t = d \mid z_t = k)$ is the element on the d^{th} row and the k^{th} column. Furthermore, we define a transition matrix $T \in \mathbb{R}^{K \times K}$ contains the transition probability $P(z_t = n \mid z_{t-1} = m)$ on its m^{th} row and n^{th} column. The prior distribution over the initial latent state z_1 is then defined as a vector $\pi = [P(z_1 = 1), \cdots, P(z_1 = K)]^\top$. Notice that the same matrices W and T are used for all time t in the sequence $\{x_t\}$, $\{z_t\}$.

In a factor graph without loops, there are two messages at each edge of the graph. They are sent by the nodes connected by the edge. In order to compute all the messages, a message-passing algorithm has to visit each node twice in both forward and backward directions. In

particular, running message-passing algorithms on HMMs can be thought of as running a bi-directional recurrent neural network (bi-RNN) from both ends of the sequence, where W are the bottom up input weights and T are the recurrent weights in an RNN with linear hidden units.

1. We continue the example from Section 4.1. Assume the observed variables x_t are one-hot vectors. Write down the expression to compute a vectorized message $\mu_{f_{z_2z_3}\to z_3} = [\mu_{f_{z_2z_3}\to z_3}(z_3 = 1), \cdots, \mu_{f_{z_2z_3}\to z_3}(z_3 = K)]^\top$ in terms of x_1, x_2, W, T, π. [2 pt.]

2. Write down the expression for $\mu_{z_3\to f_{z_2z_3}}$ in terms of x_3, x_4, x_5, W, T, π. [2 pt.]

4.4 Learning the HMM [10 pt.]

Similar to the MoG model in Assignment 3, we can learn HMMs through optimizing its marginal log likelihood $\log P(X) = \sum_{n=1}^{B} \log P(x_1^{(n)}, x_2^{(n)}, \cdots, x_T^{(n)})$ over the set of model parameters: W, T, π

1. Complete the *hmm.py* script that defines two RNN cells for the forward and the backward computation. [2 pt.]

2. Write a Tensorflow function that computes the likelihood function $P(x_t \,|\, z_t)$ for all sequences and all time steps using likelihood matrix W. The output should be a $B \times T \times K$ tensor (You may find *tf.nn.embedding_lookup* useful) [2 pt.]

3. Write a Tensorflow function that uses the custom recurrent layer *HMMCellFW/BW* with *tensorflow.models.rnn.bidirectional_rnn* function to compute posterior distributions for each latent state variable $P(z_t \,|\, x_1, x_2, \cdots, x_T)$. The function output should be a 3D tensor for B data points, T sequence length and K latent states. [2 pt.]

4. Implement the marginal log likelihood of the HMM and find the HMM parameters W, T, π by optimizing the marginal log likelihood. Note that π is constrained on a simplex, $\sum_k \pi^k = 1, \pi^k \geq 0$. In addition, all the columns of matrix W and the rows of matrix T are constrained on simplexes. You may optimize the unconstrained parameters through softmax transformation using *tf.nn.softmax*. For the dataset *dataDNA_raw.txt*, set K = 2, hold out 1/3 of the data for validation and report the best model parameters it has learnt. Include a plot of the loss vs the number of updates. [2 pt.]

5. For the dataset *dataDNA_raw.txt*, infer the posterior distributions of the latent variables on a few holdout test data sequences, $P(z_t \,|\, x_1, \cdots, x_T)$ using the best learnt model parameters. Comment on what pattern you think the HMM has learnt from this dataset. One approach you may consider is to examine the latent states that has the highest posterior probability at each time step independently. One can study $\text{argmax}_k P(z_t = k \,|\, x_1, \cdots, x_T)$ over the sequences and observe how they change with different observations. Include the model inference results for the data point in the dataset. [2 pt.]