

# 10-701 Machine Learning, Fall 2011: Homework 3

Due 10/31 at the beginning of class.

There are 4 problems in this homework, the first 3 problems are mandatory, and the last problem is optional.

## 1 Hidden Markov Model [25 points, Bin]

### 1.1 General Questions

- [4 points] For each of the following data sets, is it appropriate to use HMM? Provide a one sentence explanation to your answer.
  - Stock market price data
  - Collaborative filtering on a database of movie reviews: for example, Netflix challenge: predict about how much someone is going to enjoy a movie based on their and other users' movie preferences
  - Daily precipitation data in Pittsburgh
  - Optical character recognition
- [2 points] True or false: (if true, give a 1 sentence justification; if false, give a counter example.) When learning an HMM for a fixed set of observations, assume we do not know the true number of hidden states (which is often the case), we can always increase the training data likelihood by permitting more hidden states.
- [4 points] Show that if any elements of the parameters  $\pi$  (start probability) or  $A$  (transition probability) for a hidden Markov model are initially set to zero, then those elements will remain zero in all subsequent updates of the EM algorithm.

### 1.2 HMM for DNA Sequence

In this problem, you will use HMM to decode a simple DNA sequence. It is well known that a DNA sequence is a series of components from  $\{A, C, G, T\}$ . Now let's assume there is one hidden variable  $S$  that controls the generation of DNA sequence.  $S$  takes 2 possible states  $\{S_1, S_2\}$ . Assume the following transition probabilities for HMM  $M$

$$P(S_1|S_1) = 0.8, P(S_2|S_1) = 0.2, P(S_1|S_2) = 0.2, P(S_2|S_2) = 0.8$$

emission probabilities as following

$$P(A|S_1) = 0.4, P(C|S_1) = 0.1, P(G|S_1) = 0.4, P(T|S_1) = 0.1$$
$$P(A|S_2) = 0.1, P(C|S_2) = 0.4, P(G|S_2) = 0.1, P(T|S_2) = 0.4$$

and start probabilities as following

$$P(S_1) = 0.5, P(S_2) = 0.5$$

Assume the observed sequence is  $x = CGTCAG$ , calculate:

- [5 points]  $P(x|M)$  using the forward algorithm. Show your work to get full credit.
- [5 points] The posterior probabilities  $P(\pi_i = S_1|x, M)$  for  $i = 1, \dots, 6$ . Show your work to get full credit.
- [5 points] The most likely path of hidden states using the Viterbi algorithm. Show your work to get full credit.

## 2 Bayesian Network [25pt, Nan Li]

Bayesian networks provide an efficient way to encode causality relationships among variables in a DAG. Figure 1 is a Bayesian network. Assume that all the variables are boolean. We will explore the probability distribution encoded in this graph in this problem.

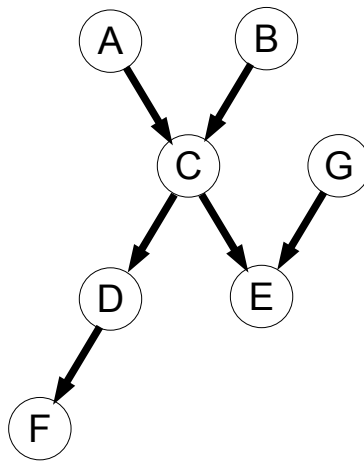


Figure 1: A Bayesian Network

### 2.1 True or False [4 pt]

Please answer whether the following statement is true or not. If not, briefly explain why.

- [2 pt] We can encode any probability distribution with a Bayesian network.
- [2 pt] For each Bayesian network  $G$ , it represents a family of all distributions,  $\mathcal{D}$ , that satisfy  $I(G)$ . Since D-separation is sound and complete, any independence assumption that cannot be derived from the graph will not hold for any distribution in  $\mathcal{D}$ .

### 2.2 Joint Probability [3 pt]

Please write down the formula to calculate the joint probability of all variables in factored form based on the conditional independence assumptions shown above.

### 2.3 Number of Parameters [7 pt]

- (a) [2 pt] If we do not make any conditional independence assumption on the variables, how many parameters do we need to store the complete probability distribution?
- (b) [5 pt] If the distribution is consistent with the ones expressed in the Bayesian network, how many parameters do we need to store the distribution?

### 2.4 Markov Blanket [3 pt]

What is the Markov blanket of variable  $C$ ?

### 2.5 D-Separation [8 pt]

Please answer whether the following conditional independence statements are true or not based on the Bayesian network using the Bayes-ball algorithm. Briefly explain why if two variables are d-separated, i.e. point out the variable that prevent the ball to get through.

- (a)  $A \perp B | C$
- (b)  $A \perp G | E$
- (c)  $B \perp G | C, E$
- (d)  $F \perp G$

## 3 Conditional Random Fields [Suyash, 25 points]

### 3.1 CRFs and HMMs [6 points]

Explain the difference between Conditional Random Fields and Hidden Markov Models with respect to the following factors. Please give only a one-line explanation.

- [2 points] Type of model - generative/discriminative
- [2 points] Objective function optimized
- [2 points] Require a normalization constant

### 3.2 Features in CRFs [8 points]

Consider the standard CRF discussed in class (Lecture 12, slide 22). For each of the following feature functions, explain whether they can be represented by the CRF probability distribution? Briefly explain your answer.

1. [2 points]  $m_k = \mathbb{I}[y_i = y_{i+1}]$
2. [2 points]  $n_k = \mathbb{I}[\text{tag}(y_i) = \text{“Proper noun” AND } X_i \text{ is uppercase}]$  (Assume that the  $X$ 's are words and  $Y$ 's are the part-of-speech tags).
3. [2 points]  $o_k = \mathbb{I}[y_i = y_{i+2}]$
4. [2 points]  $n_k = \mathbb{I}[\text{tag}(y_i) = \text{“Proper noun” AND } X_{i-1} \text{ is an article}]$  (Assume that the  $X$ 's are words and  $Y$ 's are the part-of-speech tags).

### 3.3 Complex CRFs [7 points]

Conditional random fields can have more complex structures than the one described in class. Figure 2 shows a (part of a) CRF called the skip chain CRF. Assume that we have a single feature  $f$  for the edges between the  $Y$  variables and single feature  $g$  for the edges between the  $Y$  and  $X$  variables (in the notation of Lecture 12, slide 20, assume  $k = 1$ ).

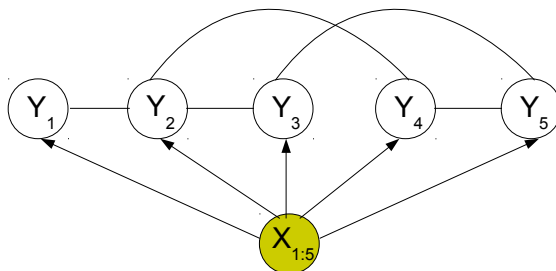


Figure 2: Skip chain CRF

- [5 points] Write down the expression for the conditional distribution of this skip chain CRF in expanded form (no summations). You can use  $Z(x)$  to represent the normalization constant in this expression.
- [2 points] Write down the expression for  $Z(x)$  in expanded form. Your expression may involve a summation over  $y$ .

### 3.4 Computing the normalization constant [4 points]

We will examine how expensive computing the normalization constant in a CRF is. Consider the normalization constant for the CRF in the previous problem. For this problem, we will assume that each  $Y$  variable can only take two values.

- [1 points] Assuming you compute  $Z(x)$  naively, how many terms would you have to sum over?
- [2 points] In general, if there were  $n$  variables  $Y_1, \dots, Y_n$ , how many terms would the summation involve?
- [1 points] What does that tell you about the amount of time required to compute  $p(y|x)$  for a new  $x$  (not seen before) and a given  $y$  in a CRF?

## 4 Extra Credits: Gibbs Sampling for an Infinite Gaussian Mixture Model [Qirong Ho, 10 points]

In this question, we're going to show that the Gibbs sampler for a  $K$ -Gaussians mixture model can be easily extended to accommodate infinite Gaussian centers, by way of the Chinese Restaurant Process. We shall see that the resulting infinite Gaussian mixture model automatically selects the number of Gaussians relevant to the data, thus obviating the need to choose  $K$ .

First, let us review the  $K$ -Gaussians mixture model. Assume we have  $N$  observed data points in  $D$  dimensions, denoted by the vectors  $x_1, \dots, x_N$ . We also assume that there are  $K$  Gaussian with

unknown means  $\mu_1, \dots, \mu_K$ , and for simplicity, we shall assume that all  $K$  Gaussians have known covariances equal to the identity matrix  $I$ . Finally, we let  $z_1, \dots, z_N$  represent the (unknown) Gaussian that each data point belongs to. The variables  $z$  are discrete, for example,  $z_1 = 1$  says that data point  $x_1$  comes from the Gaussian with mean  $\mu_1$ . For convenience, we let  $\mathbf{x} = \{x_1, \dots, x_N\}$  denote the set of all data points  $x_i$ ,  $\mathbf{z} = \{z_1, \dots, z_N\}$  denote the set of all variables  $z_i$ , and  $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_K\}$  denote the set of all Gaussian means  $\mu_k$ . Under this model, the data points  $x_i$  have the following probability distributions:

$$p(x_i = x \mid z_i, \boldsymbol{\mu}) = (2\pi)^{-D/2} \exp\left\{-\frac{1}{2} \|x - \mu_{z_i}\|_2^2\right\},$$

where  $\|\cdot\|_2^2$  is the squared Euclidean distance.

Thus far, we have assumed that only the  $\mathbf{x}$  (and  $\mathbf{z}$ ) are random variables, distributed according to  $x_i \sim \text{Normal}(\mu_{z_i}, I)$ . In class, you learnt how to use the EM algorithm to find the MLE values of  $\boldsymbol{\mu}$ , which are simply unknown, non-random parameters. After learning  $\boldsymbol{\mu}$ , we could then find the MAP values of  $\mathbf{z}$ , i.e. the most likely Gaussian for each data point. However, in order to derive a Gibbs sampler, we must assume that the  $\boldsymbol{\mu}$  are also random variables, distributed according to some *prior* distribution. For this question, we shall assume  $\mu_k \sim \text{Normal}(0, I)$ , implying that the prior distribution of each  $\mu_k$  is given by

$$p(\mu_k = u) = (2\pi)^{-D/2} \exp\left\{-\frac{1}{2} \|u\|_2^2\right\}.$$

All that remains is to define a prior over  $\mathbf{z}$ .

#### 4.1 Uniform discrete prior for $\mathbf{z}$

Let's start with the simplest possible prior for  $\mathbf{z}$ , a uniform discrete distribution

$$p(z_i = k) = \frac{1}{K} \quad \text{for } k \in \{1, \dots, K\}.$$

Notice that we are still assuming  $K$  Gaussians, since  $z_i$  can only take integer values from 1 through  $K$ .

- **1. [2 points]** It can be shown that the probability of  $z_i$  conditioned on all other random variables is

$$p(z_i = k \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\}) \propto p(x_i \mid z_i = k, \boldsymbol{\mu}) p(z_i = k),$$

where the notation  $\mathbf{z} \setminus \{z_i\}$  represents the set of all variables  $\mathbf{z}$  except  $z_i$ . Rewrite the RHS by substituting in the appropriate expressions for each  $p(x_i \mid z_i, \boldsymbol{\mu})$  and  $p(z_i)$ , and drop any constant multiplicative factors that do not depend on  $k$ . Since this conditional distribution is discrete, the normalization constant is simply computed by summing this expression over all values of  $k \in \{1, \dots, K\}$  (you don't have to compute this).

- **2. [2 points]** Also, it can be shown that the probability of  $\mu_k$  conditioned on all other random variables is

$$p(\mu_k = u \mid \mathbf{x}, \mathbf{z}, \boldsymbol{\mu} \setminus \{\mu_k\}) \propto \left[ \prod_{i=1}^N p(x_i \mid z_i, \mu_k = u, \boldsymbol{\mu} \setminus \{\mu_k\})^{\delta(z_i=k)} \right] p(\mu_k = u),$$

where  $\boldsymbol{\mu} \setminus \{\mu_k\}$  represents the set of all variables  $\boldsymbol{\mu}$  except  $\mu_k$ , and the indicator function  $\delta(z_i = k)$  is defined to be 1 if  $z_i = k$ , and 0 otherwise. Notice that the indicator functions  $\delta(z_i = k)$  simply “pick out” the probabilities of the data points belonging to the Gaussian  $\mu_k$ . Rewrite the RHS by substituting in the appropriate expressions for each  $p(x_i | z_i, \boldsymbol{\mu})$  and  $p(\mu_k)$ , and drop any constant multiplicative factors that do not depend on  $u$ . Note that this conditional distribution can be shown to be Gaussian, i.e. proportional to  $\exp\left\{-\frac{1}{2}\|a - u\|_2^2\right\}$  for some  $a$ . We won’t require you to express the RHS as this form, but you should know that in practice this is required to compute the normalization factor.

These two expressions define a Gibbs sampler algorithm. By randomly drawing each  $z_1, \dots, z_N$  and  $\mu_1, \dots, \mu_K$  in sequence according to the above expressions, we perform Gibbs sampling for the  $K$ -Gaussians mixture model.

## 4.2 An infinite prior over $\mathbf{z}$

In order to extend our model to handle an infinite number of Gaussians, we shall use a special prior for  $\mathbf{z}$  called the Chinese Restaurant Process. Under this prior, the  $\mathbf{z}$ ’s are no longer independent, and we define their prior conditional probabilities as

$$p(z_i = k | \mathbf{z} \setminus \{z_i\}) = \begin{cases} \frac{\#\{\mathbf{z} \setminus \{z_i\} = k\}}{N + \alpha} & \text{if } \#\{\mathbf{z} \setminus \{z_i\} = k\} > 0 \\ \frac{\alpha}{N + \alpha} & k \text{ is the smallest positive integer such that } \#\{\mathbf{z} \setminus \{z_i\} = k\} = 0. \end{cases}$$

The notation  $\#\{\mathbf{z} \setminus \{z_i\} = k\}$  is shorthand for the number of random variables in the set  $\mathbf{z} \setminus \{z_i\}$  that have the value  $k$ . Essentially, the probability that we sample  $z_i = k$  (i.e. we choose the Gaussian  $\mu_k$ ) is proportional to the number of other  $\mathbf{z}$ ’s already equal to  $k$  — unless there are no other  $\mathbf{z}$ ’s equal to  $k$ , in which case the probability is proportional to  $\alpha$ . Notice that the second case is analogous to creating a new Gaussian cluster, and by convention we number this cluster using the *smallest positive integer* not already used by some other Gaussian. Finally, the parameter  $\alpha$  controls the probability of creating new Gaussian centers — observe that as  $\alpha$  increases, so does the prior probability of drawing a new  $k$ .

Let’s see how this Chinese Restaurant Process prior changes our Gibbs sampler algorithm.

- **1. [1.6 points]** It can be shown that the probability of  $z_i$  conditioned on all other random variables now becomes

$$p(z_i = k | \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\}) \propto p(x_i | z_i = k, \boldsymbol{\mu}) p(z_i = k | \mathbf{z} \setminus \{z_i\}) \quad \text{if } \#\{\mathbf{z} \setminus \{z_i\} = k\} > 0.$$

Note that, for now, we are only considering cases where  $\#\{\mathbf{z} \setminus \{z_i\} = k\} > 0$  (i.e. the Gaussian  $\mu_k$  is associated with other data points). Rewrite the RHS by substituting in the appropriate expressions for each  $p(x_i | z_i, \boldsymbol{\mu})$  and  $p(z_i | \mathbf{z} \setminus \{z_i\})$ , and drop any constant multiplicative factors that do not depend on  $k$ .

- **2. [0.8 points]** Now we consider the case where  $k$  is the smallest positive integer such that  $\#\{\mathbf{z} \setminus \{z_i\} = k\} = 0$  (i.e. we’re trying to create a new Gaussian  $\mu_k$ ). The conditional probability of  $z_i$  is now

$$p(z_i = k | \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\}) \propto p(x_i | z_i = k, \boldsymbol{\mu}) p(z_i = k | \mathbf{z} \setminus \{z_i\})$$

$k$  is the smallest positive integer such that  $\#\{\mathbf{z} \setminus \{z_i\} = k\} = 0.$

Rewrite the RHS by substituting in the appropriate expressions for each  $p(x_i | z_i, \boldsymbol{\mu})$  and  $p(z_i | \mathbf{z} \setminus \{z_i\})$ , and drop any constant multiplicative factors that do not depend on  $u$ . Let’s assume for now that the value of  $\mu_k$  is known, even though it’s a newly-created Gaussian.

Finally, the probability of  $\mu_k$  conditioned on all other random variables,  $p(\mu_k = u \mid \mathbf{x}, \mathbf{z}, \boldsymbol{\mu} \setminus \{\mu_k\})$ , remains the same (so we don't have to re-derive it). Therefore, we have completely derived a Gibbs sampler for the infinite Gaussian mixture model... or have we?

### 4.3 A few subtleties

There is a problem with the above Gibbs sampling equations: when we compute the probability  $p(z_i = k \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\})$  for the “new Gaussian” case (i.e.  $k$  is the smallest positive integer such that  $\#[\mathbf{z} \setminus \{z_i\} = k] = 0$ ), we assumed that  $\mu_k$  is already known. This isn't true! We don't know the value of  $\mu_k$  precisely because it's a new Gaussian (so our Gibbs sampler wouldn't have sampled its value yet).

- **1. [1.6 points]** In fact, the correct thing to do is to marginalize (integrate out) the unknown new value of  $\mu_k$ . Thus, the corrected expression of the conditional probability of  $z_i$  is

$$p(z_i = k \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\}) \propto \left[ \int_u p(x_i \mid z_i = k, \mu_k = u, \boldsymbol{\mu} \setminus \{\mu_k\}) p(\mu_k = u) du \right] p(z_i = k \mid \mathbf{z} \setminus \{z_i\})$$

$k$  is the smallest positive integer such that  $\#[\mathbf{z} \setminus \{z_i\} = k] = 0$ .

The prior for the Gaussian means  $p(\mu_k = u)$  comes in because marginalizing out  $\mu_k$  puts a dependency on its prior, which is  $p(\mu_k = u)$ . Rewrite the RHS by substituting in the appropriate expressions for each  $p(x_i \mid z_i, \boldsymbol{\mu})$ ,  $p(z_i \mid \mathbf{z} \setminus \{z_i\})$  and  $p(\mu_k)$ , and drop all constant multiplicative factors that do not depend on  $k$ . You do not have to evaluate the integral (for your information, it actually evaluates to a Gaussian form).

With this corrected expression, we can now sample  $z_i$  from its conditional distribution  $p(z_i = k \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\})$ . Note that the normalization factor is computed by summing this expression over all  $k$  such that  $\#[\mathbf{z} \setminus \{z_i\} = k] > 0$ , and adding the “new Gaussian” case to that sum. If we do end up assigning  $z_i = k$  where  $k$  is the “new Gaussian”, then we immediately draw a value for  $\mu_k$  from the conditional distribution  $p(\mu_k = u \mid \mathbf{x}, \mathbf{z}, \boldsymbol{\mu} \setminus \{\mu_k\})$  (which will depend only on the data point  $x_i$  and the prior  $p(\mu_k)$ ). Conversely, when sampling the  $\mathbf{z}$ 's, if some Gaussian  $\mu_k$  ends up with zero assigned data points, then we must delete that  $\mu_k$ . These additional but important details complete the infinite Gaussian mixture model Gibbs sampler algorithm.

With all that said and done, answer the following questions:

- **2. [0.8 points]** Compare your expressions for  $p(z_i = k \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{z} \setminus \{z_i\})$  in the  $K$ -Gaussians and infinite Gaussians case. Where are they the same, and where do they differ?
- **3. [1.2 points]** For the infinite Gaussian mixture model, the Chinese Restaurant Process prior over  $\mathbf{z}$  is clearly not uniform — whereas the  $K$ -Gaussians model used a uniform prior. Could we design a uniform infinite prior for  $\mathbf{z}$ ? Explain why or why not.