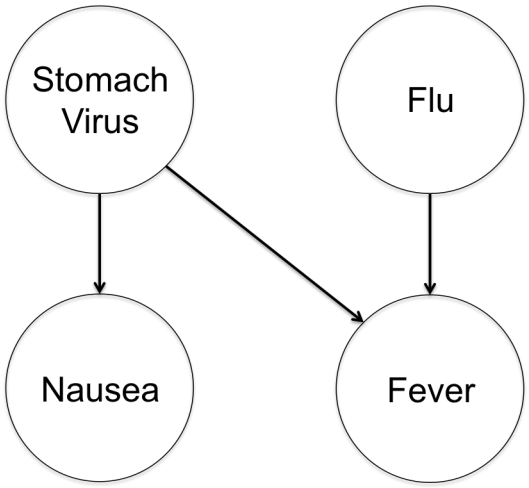
6.867: Exercises (Week 9)

November 10, 2017

Contents

1 Bayesian Network Parameter Estimation

You are a doctor working at a clinic and want to model the relationship between common diseases and symptoms. Drawing on your vast knowledge of the human body you decide to use the following Bayesian network:



Given data on previous patients, you now want to determine the parameters of this network. Let X_1, X_2, X_3, and X_4 be binary random variables representing the event of a given patient having a stomach virus, having the flu, having nausea and having a fever, respectively. In other words, these variables denote the nodes in the network.

Your dataset D consists of the patient profiles of n past patients. The dataset is complete, meaning that you know the values of $X_1^{(i)}$, $X_2^{(i)}$, $X_3^{(i)}$, and $X_4^{(i)}$ for all patients i. We will first find the maximum likelihood parameter estimates for the network.

(a) Provide an expression for the log-likelihood of the data given this network structure in terms of parameterized conditional probability distributions that shows that each conditional probability distribution can be optimized independently.

(b) Assume that the conditional distributions are fully parameterized (ie. in $p(y_1|y_2)$ the distribution over y_1 can be set independently for each value of y_2). Provide expressions for the maximum likelihood parameters in terms of patient frequency counts (eg. $N(X_1 = 1)$ is the number of patients with a stomach virus).

(c) We have been assuming that the conditional probability distributions are fully parameterized. Under this assumption, how many parameters must be learned for the conditional probability distribution of X_4?

To decrease the number of parameters we can make some assumptions about the structure of our conditional probability distributions. In practice, for discrete variables a Noisy-OR distribution is commonly used. Here, any individual parent being true can independently

cause the child to be true but with some error rate p_i for parent i. This means that for a node x_0 with n parents, where all variables take values in $\{0, 1\}$ we get:

$$p(x_0 = 1|x_1, ..., x_n) = 1 - \prod_{i=1}^{n} p_i^{x_i}$$

(d) In our disease example, if we know $p(X_4 = 1|X_1 = 0, X_2 = 1) = 0.9$ and $p(X_4 = 1|X_1 = 1, X_2 = 0) = 0.7$, what is $p(X_4 = 1|X_1 = 1, X_2 = 1)$?

(e) Using a Noisy-OR distribution, how many parameters must be learned for the conditional probability distribution of X_4? Why might this generalize better than a fully parameterized distribution?

2 K-Means Clustering

This question will explore some cases in which the k-means clustering algorithm may or may not give ideal results. All parts of this question will use squared Euclidean distance as a distance metric.

(a) First provide an expression for the objective function being minimized by a k-means algorithm using Euclidean distance squared.

(b) Consider a 1D dataset generated from the two Gaussian distributions $N_1(\mu_1, \sigma_1^2)$ and $N_2(\mu_2, \sigma_2^2)$. Assume the dataset contains n points from each distribution for some arbitrarily large n.

Let's define the optimal classifier as the classifier that assigns each point to the most likely cluster given knowledge of the generating distribution. Consider the case where $\sigma_1^2 = \sigma_2^2$. Would you expect a 2-means clustering algorithm to approximate the optimal classifier for large n? Explain.

(c) Now consider the case where $\sigma_1^2 \gg \sigma_2^2$. Would you expect a 2-means clustering algorithm to approximate the optimal classifier for large n? Explain.

(d) What might be a better way to cluster the data in the case where $\sigma_1^2 \gg \sigma_2^2$?

3 EM for word counts

You want to create a language model that models text. Instead of using a simple bi-gram model, you observe that the grammatical word class of a word can usually be predicted by the previous word and decide to explicitly model this. Consider the probabilistic model of the following form:

$$p(w_2, c \mid w_1) = q(c \mid w_1)q(w_2 \mid c)$$

Here w_1 and w_2 are words, drawn from some set of possible words $\mathcal{V}$. c is a word class, which can take any value in the set $\{1, 2, \ldots, k\}$ for some integer k. $q(c \mid w_1)$ and $q(w_2 \mid c)$ are the parameters of the model. We can interpret this as a model where: (1) a class c is generated by word w_1; (2) w_2 is then generated by the class c chosen in step 1.

Under this model, we can derive

$$p(w_2 \mid w_1) = \sum_{c=1}^{k} q(c \mid w_1) q(w_2 \mid c)$$

This will be a model of the conditional probability of seeing the word w_2 given that the previous word in a sentence was w_1. We want to derive a method to compute the q parameters for this model.

(a) Write down an expression for $p(c \mid w_1, w_2)$ as a function of the q parameters.

(b) Say for each pair of words w_1, w_2, count(w_1, w_2) is the number of times w_1 is followed by w_2 in our training data. We are going to derive an EM algorithm for optimization of the following log likelihood function:

$$L(\theta) = \sum_{w_1, w_2} \text{count}(w_1, w_2) \log p(w_2 \mid w_1)$$

For given parameter values q, define count$(c \mid w_1)$ to be expected number of times that w_1 generates class c, and define count$(w_2 \mid c)$ to be the expected number of times w_2 is generated by class c. (Here expectation is taken with respect to the distribution defined by the q parameters.) State how count$(c \mid w_1)$ and count$(w_2 \mid c)$ can be calculated as a function of the q parameters:

(c) Now describe how the q parameters are recalculated in the EM algorithm, based on the count$(c \mid w_1)$ and count$(w_2 \mid c)$ counts derived in the previous part.

(d) Say we initialize the parameters to be $q(c \mid w_1) = 1/k$ for all w_1,c, and $q(w_2 \mid c) = 1/ \mid \mathcal{V} \mid$ for all w_2,c. Given these initial parameter values, what parameter values will the EM algorithm converge to?

4 Missing data

In most real world datasets we do not have the privilege of complete data. Certain observations may be incorrect or never taken in the first place. For example, in survey data you may not have answers to every question for every person because they chose not to answer some questions. Due to this being able to make predictions despite missing data is very important in practice.

To gain intuition about how to handle missing data we'll start with a very simple problem, in which a single attribute of a single data set is missing. There are two attributes, A and B, and this is our data set, $\mathcal{D}$:

```
i    A  B
1    1  1
2    1  1
3    0  0
4    0  0
5    0  0
6    0  H ***missing **
7    0  1
8    1  0
```

Assume the data is *missing completely at random* (MCAR): that is, that the fact that it is missing is independent of its value.

Our goal is to estimate $\Pr(A, B)$ from this data. We'd really like to find the maximum-likelihood parameter values, if we can. The likelihood is

$$\mathcal{L}(\theta) = \log \Pr(\mathcal{D}; \theta) = \log \left(\Pr(\mathcal{D}, H = 0; \theta) + \Pr(\mathcal{D}, H = 1; \theta) \right) \quad .$$

(a) Kim is lazy and decides to ignore $x^{(6)}$ all together, and estimate the parameters:

$$\widehat{\theta}^1 = \begin{pmatrix} 3/7 & 1/7 \\ 1/7 & 2/7 \end{pmatrix} = \begin{pmatrix} .429 & .143 \\ .143 & .285 \end{pmatrix}$$

What is $\mathcal{L}(\widehat{\theta}^1)$?

(b) Jan thinks we should let H be the 'best' value it could have, that is to make the log likelihood as large as possible, and so tries setting $H = 0$ and then $H = 1$ and computes the log likelihood of the complete data in both cases. What value gives the highest complete-data log likelihood? What is the likelihood value?

(c) Evelyn thinks this is all unprincipled messing around and says we should optimize the thing we want to optimize! That is,

$$\widehat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta) \quad .$$

Evelyn also thinks we can just use the code for gradient descent that we already built in 6.867 to do this job. Is Evelyn right?

(d) Ariel was paying close attention in lecture and thinks EM is good to use for estimation with missing data because the missing data is effectively just a latent variable.

Let's start with the guess

$$\theta_0 = \begin{pmatrix} .25 & .25 \\ .25 & .25 \end{pmatrix}$$

What is the formula for the E step in this problem? What is the numerical result in this particular case?

(e) Ariel's roommate Angel joins in the EM game and computes the M step, to get θ_1. What is the numerical value in this case, and why?

(f) Will EM always find a solution that maximizes $\mathcal{L}$?

5 Factor Analysis

You are given a dataset $X \in \mathbb{R}^{n \times k}$ of gene expression data where k is the number of genes and n is the number of observations. But $n < k$, posing a challenge for many modelling approaches (eg. fitting a Gaussian). You believe patterns in gene expression data can be explained by less variables

than the number of genes so instead you decide to use the following factor analysis model where $z \in \mathbb{R}$ is your latent variable:

$$z \sim N(0,\ 1)$$
$$x|z \sim N(wz,\ I)$$

where $w \in \mathbb{R}^{k\times 1}$ and I is the $k \times k$ identity matrix. We are modelling the variations in the k-dimensional variable x as a function of variations in the lower dimensional variable z. Here we have chosen z to be 1D to simplify the calculations, but in general this method can be used with multivariate z.

In this question we will derive the EM update steps for determining the maximum likelihood estimate of w. For the following questions it will be useful to note that:

$$p(z, x; w) \sim N(\begin{bmatrix}0\\0\end{bmatrix},\ \begin{bmatrix}1 & w^T\\ w & ww^T + I\end{bmatrix})$$

And if y_1 and y_2 are Gaussian random variables, we have:

$$\mu_{y_1|y_2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_2 - \mu_2)$$
$$\Sigma_{y_1|y_2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

where $\Sigma_{ij} = cov(y_i, y_j)$.

(a) First we will compute the E-step. Provide an expression for $p(z|x; w^{(t)})$.

(b) In the M-step we want to find $w^{(t+1)}$ that maximizes the expected complete log likelihood $\sum_{i=1}^{n} \mathbb{E}[\log p(x^{(i)}, z^{(i)}; w^{(t+1)})]$ where the expectation is taken with respect to $p(z|x; w^{(t)})$. Show that this is equivalent to maximizing $L = \sum_{i=1}^{n} \mathbb{E}[\log p(x^{(i)}|z^{(i)}; w^{(t+1)})]$.

(c) Provide an expression for L.

(d) Find the update step by solving for the maximizing $w^{(t+1)}$ in terms of $\mu_{z|x}$ and $\Sigma_{z|x}$.

6 Time Varying Topic Model

In recent years, social media outlets represent a wealth of data related to people's opinions and world events. As such, tools for understanding this data have grown increasingly important. Intrigued by this trend, you decide you want to perform topic modelling on a corpus of tweets collected over some time interval. Your dataset consists of M tweets (or documents) at each time step t, with tweet i at time t denoted by $D_{i,t}$, for T total time steps. Tweet $D_{i,t}$ is composed of $L_{i,t}$ words where the j^{th} word is denoted by $w_{i,t,j}$.

You believe that any given tweet can be decomposed as a mixture of K topics. However, as is the nature of Twitter, you believe the topic proportions and the nature of the topics themselves are changing over time. Because of this you need to figure out a way to handle the time varying nature of the topics in your topic modelling. Here is the generative process for a static topic model you are using as a starting point:

1. For each document i choose $\theta_i \sim \mathrm{Dir}(\alpha)$
2. For each topic k choose $\phi_k \sim \mathrm{Dir}(\beta)$
3. For each word i, j in document i and position j:
   (a) Choose a topic $z_{i,j} \sim \mathrm{Multi}(\theta_i)$
   (b) Choose a word $w_{i,j} \sim \mathrm{Multi}(\phi_{z_{i,j}})$

where $\mathrm{Dir}()$ is a Dirichlet distribution and $\mathrm{Multi}()$ is a multinomial distribution with 1 trial.

To gain intuition about time varying topics, let's first consider a simplified case. We want to see which distributional assumptions are good for modelling our time varying latent variables. To do this assume we know the true value of the topic proportions for the documents at each of T time steps (assuming there is a single topic proportion at each time step). For further simplicity we only care about modelling the topic proportions (step 1 above). We will see how different modelling approaches perform on this data.

(a) First you consider only using the data at a specific time step t for training. What issues might this approach have?

(b) Next you consider ignoring the time altogether and using the static topic model shown above on the pooled data. What issues might this approach have?

(c) Now let's assume a topic proportion at time t is randomly selected from the vicinity of the topic proportion at time $t-1$. What distribution could be used to model this? Does this seem like a better approach?

(d) Moving back to the original problem without our simplifying assumptions, provide a generative process for tweets analogous to that of the static topic model for time varying topics. Recall we are assuming both topic proportions and the topics themselves (defined as their corresponding distributions over words) are changing with time. In place of a Dirichlet distribution you can use a normal distribution where the samples are then passed through a softmax function $\pi()$ in order to achieve probability distributions.