6.867: Recitation Handout (Week 10)

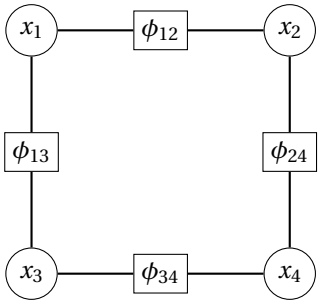
November 10, 2016

Contents

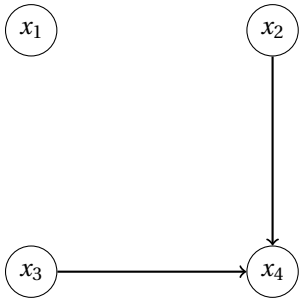
1 Learning graphical models using co-occurence counts

We wanted to estimate the parameters of several different graphical models from medical data. For privacy reasons, we weren't allowed access to the actual records but only statistics computed from the records. We requested all pairwise counts involving four binary $(0/1)$ variables $x_1, ..., x_4$. In other words, we have co-occurence counts $\hat{n}_{ij}(x_i, x_j)$ for $i \neq j$ and $x_i \in \{0, 1\}, x_j \in \{0, 1\}$.

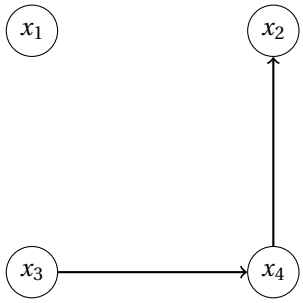
(a) Consider the graphical models shown below. Check which of them we could estimate based on the pairwise counts. By estimation we mean finding the same maximum likelihood estimates of the parameters as we would if we had access to the full records. In answering this question, make no additional assumptions about the models above and beyond the graph structure.



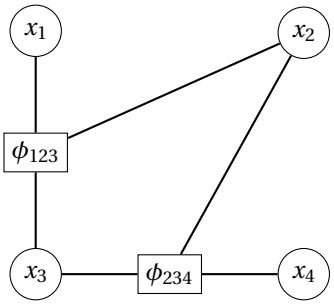
(a) **a ( )**



(b) **b ( )**



(c) **c ( )**



(d) **d ( )**

> **Solution:** a and c, since they only involve pairwise dependences.

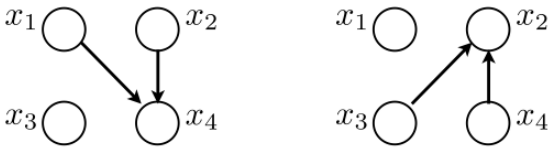
(b) Provide a brief justification for your answer to model (b) above

> **Solution:** The associated distribution includes a term $P(x_4|x_2, x_3)$. To estimate this conditional probability, we would need counts over three variables, i.e., $\hat{n}_{234}(x_2, x_3, x_4)$.

(c) We also considered building a Bayesian network model from expert knowledge. Based on our discussions with a prominent physician, we were able to extract the following properties concerning the four binary variables.

- x_1 and x_2 are independent
- x_1 and x_3 are independent
- x_3 and x_4 are independent

Draw two NOT equivalent Bayesian network models consistent with these properties avoiding (if possible) any additional assumptions not already implied by the above properties.



2 EM for word counts

Consider a 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 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.

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

Solution:

The posterior probability (our "belief") in class c, after observing w_1 and w_2, is found using Bayes' theorem.

$$\begin{aligned} p(c \mid w_1, w_2) &= \frac{p(c, w_1, w_2)}{\sum_{c'} p(c', w_1, w_2)} \\ &= \frac{p(w_1)p(w_2, c \mid w_1)}{\sum_{c'} p(w_1)p(w_2, c' \mid w_1)} \\ &= \frac{q(c \mid w_1)q(w_2 \mid c)}{\sum_{c'} q(c' \mid w_1)q(w_2 \mid c)} \end{aligned}$$

(b) Say for each pair of words w_1, w_2, $\text{count}(w_1, w_2)$ 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 $\text{count}(c \mid w_1)$ to be expected number of times that w_1 generates class c, and define $\text{count}(w_2 \mid c)$ 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 $\text{count}(c \mid w_1)$ and $\text{count}(w_2 \mid c)$ can be calculated as a function of the q parameters:

Solution:

To calculate $\text{count}(c \mid w_1)$ based on our beliefs $p(c \mid w_1, w_2)$ from the previous question, we estimate: of all occurrences of word w_1, how many do we believe generated class c? This involves marginalizing w_2.

$$\begin{aligned}\text{count}(c \mid w_1) &= \sum_{w_2} \text{count}(w_1, w_2) p(c \mid w_1, w_2) \\ &= \sum_{w_2} \text{count}(w_1, w_2) \frac{q(c \mid w_1) q(w_2 \mid c)}{\sum_{c'} q(c' \mid w_1) q(w_2 \mid c')}\end{aligned}$$

Similarly, we marginalize w_1 to estimate: of all believed occurrences of class c, how many were associated with word w_2?

$$\begin{aligned}\text{count}(w_2 \mid c) &= \sum_{w_1} \text{count}(w_1, w_2) p(c \mid w_1, w_2) \\ &= \sum_{w_1} \text{count}(w_1, w_2) \frac{q(c \mid w_1) q(w_2 \mid c)}{\sum_{c'} q(c' \mid w_1) q(w_2 \mid c')}\end{aligned}$$

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

Solution: In several instances of EM, we have found that the M-step update rules for frequency parameters involving the hidden data (e.g., Gaussian mixture proportions) work out to the weighted average frequency in the training data, where the weighting is by our beliefs about the hidden data for each training example.

For example, $\text{count}(c \mid w_1)$ from the last part tells us how many times we believe w_1 generated class c. We get a weighted frequency estimate of $q(c \mid w_1)$ by dividing this by the total number of observations of w_1.

$$q(c \mid w_1) \leftarrow \frac{\text{count}(c \mid w_1)}{\sum_{c'} \text{count}(c' \mid w_1)}$$

The denominator could be written in several equivalent ways. Similarly we divide count(w_2 | c), the number of times we believe class c generated w_2, by the total number of times we believe class c occurred.

$$q(w_2 \mid c) \leftarrow \frac{\text{count}(w_2 \mid c)}{\sum_{w_2'} \text{count}(w_2' \mid c)}$$

(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?

Solution: $q(c \mid w_1)$ will not change, simply because there is nothing to distinguish c from any other class – count($c \mid w_1$) is the same for all c. However, count($w_2 \mid c$) will vary based on how many times we observe w_2 in the training data. Therefore, we will set $q(w_2 \mid c)$ to the frequency of w_2 in the training data:

$$q(w_2 \mid c) \leftarrow \frac{\sum_{w_1} \text{count}(w_1, w_2)}{\sum_{w_1, w_2} \text{count}(w_1, w_2)}$$

("Of all training examples, what fraction had w_2?" Again, this could be written in several equivalent ways.) After this update, there is still nothing distinguishing any class from any other, so we'll be stuck after one iteration.

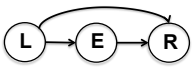
3 Buying widgets

You need widgets as components in a large system you are building. You have bought a large number of them, and you find that there is considerable variability in their longevity (length of time before failure), energy consumption, and reliability (correctness of output). You've studied the data and can find no independence relations among any of the variables.

(a) Using three variables (L, E, and R), draw a Bayesian network model of their relationships.

Solution:

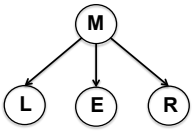
Several possibilities exist, but one solution is given below.



Now, you discover that the widgets are actually manufactured by three different suppliers, and you have reason to believe that longevity, energy consumption, and reliability are independent given the manufacturer. You are unable to tell which widget came from which manufacturer, but you can observe L, E, and R.

(b) Draw a new Bayesian network to describe this situation.

Solution:



(c) How would you learn the parameters for each model?

Solution:

The model in (a) can be learned using parameter estimation for fully observed data. In this case, the joint distribution can be estimated using the frequency of the values in the data.

The model in (b) can be learned with latent variable methods. Because M is not observed, this corresponds to the case where a whole column of data (the values corresponding to M) is missing. Methods such as EM can be used in this case.

Additional details are in the lecture notes for both of these cases of parameter estimation.

(d) Which learned model would assign higher likelihood to the data? Why?

Solution:

In general, model (a) would assign higher likelihood to the data due to the larger number of parameters; model (a) can represent any joint distribution on L, E and R.

(e) Why might you prefer each of these models?

Solution:

In general, the model in (b) will have fewer parameters than the model in (a), but it may be more difficult to estimate its parameters.

4 Missing data

We'll start with a very simple problem, in which 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 $\mathcal{L}(\widehat{\theta}^1)$?

Solution: If we do that, then

$$\begin{aligned} \mathcal{L}(\widehat{\theta}^1) &= \log \left(\Pr(00\ ;\ \widehat{\theta}^1) \prod_{i \neq 6} \Pr(x^i\ ;\ \widehat{\theta}^1) + \Pr(01\ ;\ \widehat{\theta}^1) \prod_{i \neq 6} \Pr(x^i\ ;\ \widehat{\theta}^1) \right) \\ &= 3 \log 0.429 + 2 \log 0.143 + 2 \log 0.285 + \log(0.429 + 0.143) \\ &= -9.498 \end{aligned}$$

(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?

Solution: That value is 0. So, then we'd have

$$\widehat{\theta}^2 = \begin{pmatrix} .5 & .125 \\ .125 & .25 \end{pmatrix}$$

and

$$\mathcal{L}(\widehat{\theta}^2) = -9.481 \quad .$$

That's a little better!

(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?

Solution: Evelyn is absolutely right about (if at all possible!) optimizing the thing we want to optimize.

We can do this with gradient methods, but it gets tricky because of constraint that $\widehat{\theta}$ be a valid probability distribution; that constraint is *not* maintained by our basic gradient descent code. So, we'd have to investigate constrained optimization algorithsm, or try to formulate the problem using Lagrange multipliers.

(d) Ariel was paying close attention in lecture and thinks this problem is an example of estimation in the presence of a latent variable and that we should use EM.

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?

Solution:

$$\tilde{P}(H = 1) = \Pr(H = 1 \mid \mathcal{D} \; ; \; \theta_0) = \Pr(H = 1 \mid x^{(6)} \; ; \; \theta_0) = \Pr(B = 1 \mid A = 0 \; ; \; \theta_0) = 0.5$$

(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?

Solution:

$$\begin{aligned} \theta_1 &= \arg\max_{\theta} \left(0.5 \log \Pr(\mathcal{D}, H = 0 \; ; \; \theta) + 0.5 \log \Pr(\mathcal{D}, H = 1 \; ; \; \theta)\right) \\ &= \begin{pmatrix} 7/16 & 3/16 \\ 2/16 & 4/16 \end{pmatrix} \end{aligned}$$

This step is not immediately obvious: to derive it, we need to take the derivative with respect to each of the parameters, set to 0, and solve for θ. We find that we can treat the estimation problem as one in which we have a data item for each possible value of H, weighted by the probability that H has that value. We get such a decomposition because, for each particular value of H, only one of the parameter estimates is affected.

We get the same result by doing estimation as usual, but treating $\Pr(H)$ as giving us fractional counts on both data cases:

```
i     A   B    count = P~(H)
1     1   1
2     1   1
3     0   0
4     0   0
5     0   0
6a    0   0     0.5
6b    0   1     0.5
7     0   1
8     1   0
```

On subsequent EM iterations, we have $\mathcal{L}(\theta) = -10.39, -9.47, -9.4524, -9.4514, \ldots$.

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

Solution: No. It will converge monotonically to a *local optimum* but it may not be a global optimum, and will depend, in general, on your initial guess.

5 Mixed-up mixture

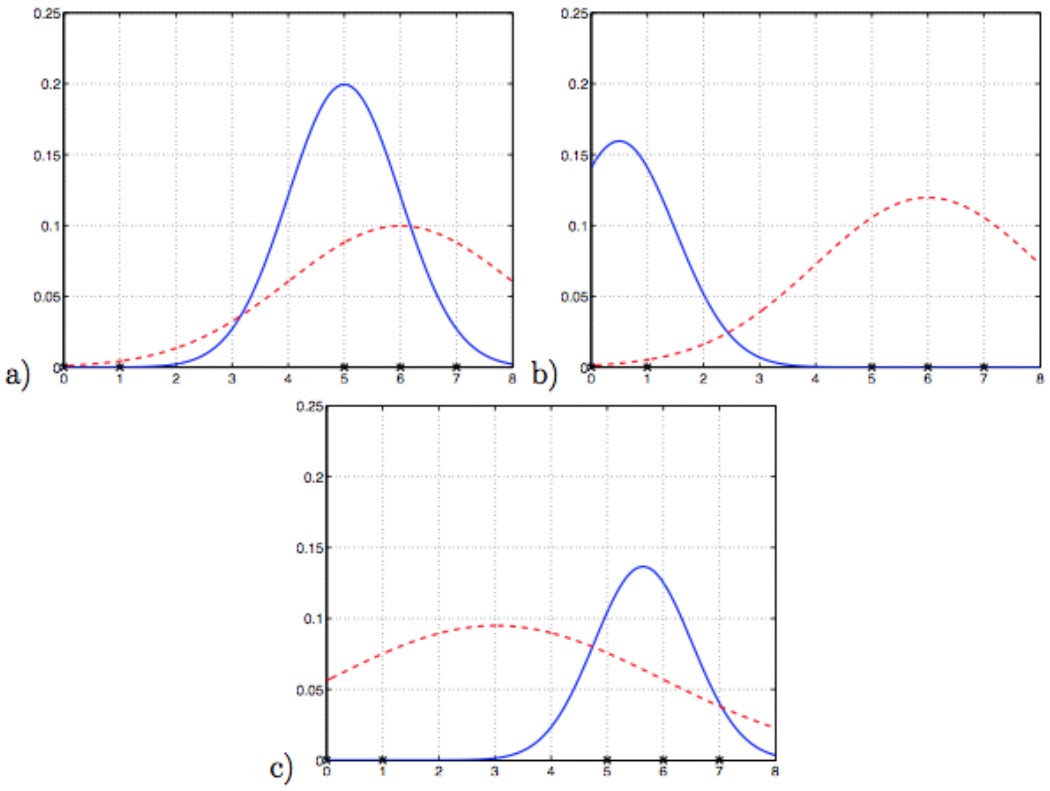
Here we are estimating a mixture of two Gaussians via the EM algorithm. The mixture distribution over x is given by

$$P(x;\theta) = P(1)N(x;\mu_1,\sigma_1^2) + P(2)N(x;\mu_2,\sigma_2^2)$$

Any student in this class could solve this estimation problem easily. Well, one student, devious as they were, scrambled the order of figures illustrating EM updates. They may have also slipped in a figure that does not belong. Your task is to extract the figures of successive updates and explain why your ordering makes sense from the point of view of how the EM algorithm works. All the figures plot $P(1)N(x;\mu_1,\sigma_1^2)$ as a function of x with a solid line and $P(2)N(x;\mu_2,\sigma_2^2)$ with a dashed line.

(a) (True/False) In the mixture model, we can identify the most likely T posterior assignment, i.e., j that maximizes $P(j \mid x)$, by comparing the values of $P(1)N(x;\mu_1,\sigma_1^2)$ and $P(1)N(x;\mu_2,\sigma_2^2)$

Solution: True



(b) Assign two figures to the correct steps in the EM algorithm.
- Step 0: ( ) initial mixture distribution
- Step 1: ( ) after one EM-iteration

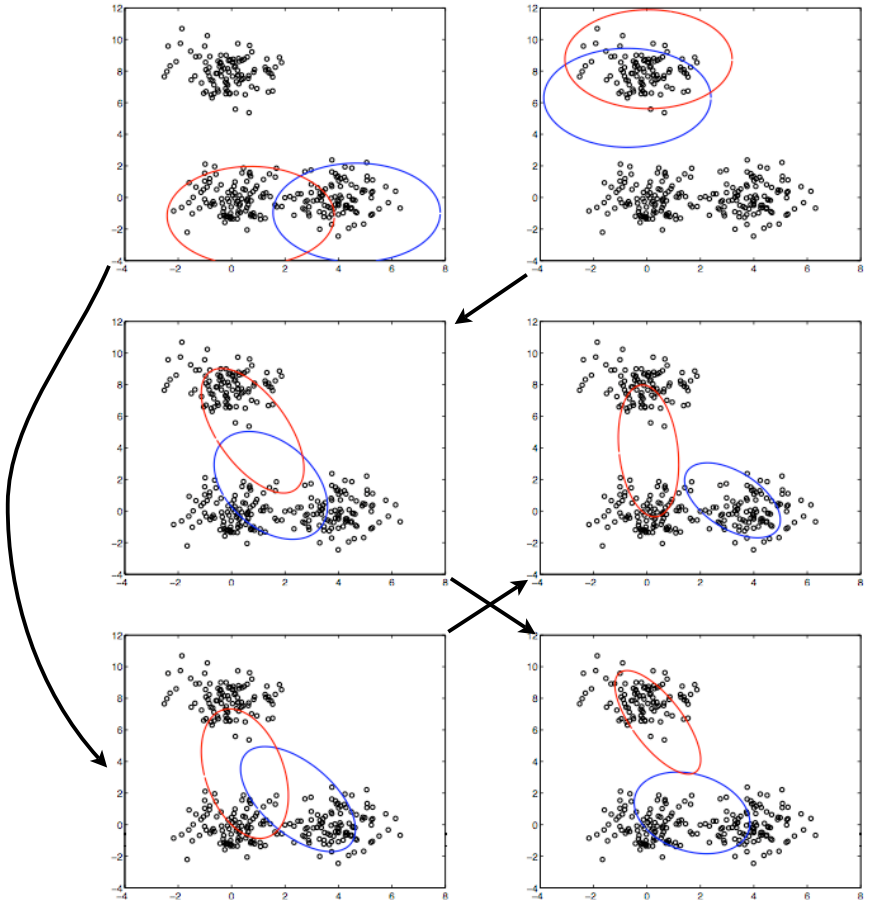
Solution: - Step 0: a
- Step 1: c

(c) Briefly explain how the mixture you chose for "step 1" follows from the mixture you have in "step 0".

Solution: The two points on the left will be assigned more to the second (red) Gaussian since $P(1)N(x;\mu_1,\sigma_1^2) < P(2)N(x;\mu_2,\sigma_2^2)$ for those points. The points on the right, except for the very last one, will be assigned mostly to the first (blue) Gaussian. As a result, the first Gaussian will become more concentrated around the two points on the right, while the second (red) Gaussian will move to the left and will have a higher variance as, in the M-step, it is estimated essentially on the basis of the spread out points $x = 0$, $x = 1$, and $x = 7$.

6 More mixture

We estimated a two Gaussians mixture model based on two-dimensional data shown in the figure below. The mixture was initialized randomly in two different ways and run for three iterations based on each initialization. However, the figures got mixed up (yes, again!). Please draw an arrow from one figure to another to indicate how they follow from each other (you should draw only four arrows).

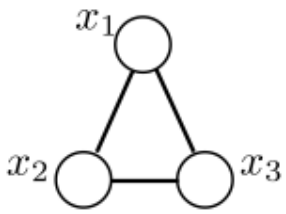


7 NB: Nota bene

Consider a Naive-Bayes model over three binary (0/1) features x_1, x_2, and x_3 and two classes $y = 0, 1$. By marginalizing over the class variable y, we obtain a two component mixture distribution

$$P(x_1, x_2, x_3; \theta) = \sum_{y=0,1} q(y) \prod_{j=1}^{3} q_j(x_j \mid y)$$

(a) If we wanted to represent this mixture distribution as a graphical model over x_1, x_2, and x_3, what would the graph look like? Draw the necessary edges or arrows in the figure below.



(b) We will use the EM algorithm to learn the parameters of the Naive Bayes model from feature observations (x_1, x_2, x_3). We initialized the parameters as follows:

$q(y = 1) = 1/2$, the class conditional feature distributions are randomized but with the constraint that $q_j(x_j = 1 \mid y = 0) = q_j(x_j = 1 \mid y = 1)$, for all j.

What can you say about $P(y = 1 \mid x_1, x_2, x_3; \theta)$ as a function of (x_1, x_2, x_3) given our initialization?

Solution: The class-conditional feature distributions agree and $q(y = 1) = q(y = 0)$. Thus $P(y = 1 \mid x_1, x_2, x_3; \theta) = 0.5$ for any configuration (x_1, x_2, x_3).

(c) Let $\hat{n}_j(x_j)$ denote the number of times the j^{th} feature has value x_j in the data. Given our initialization above, what are the expected counts $\hat{n}_j(x_j, y)$ we would evaluate in the first step of the EM-algorithm?

Solution: The expected counts $\hat{n}_j(x_j, y)$ are evaluated from data where each feature combination (x_1, x_2, x_3) is completed with the posterior $P(y \mid x_1, x_2, x_3; \theta)$. Thus $\hat{n}_j(x_j, y) = \hat{n}_j(x_j) \cdot 0.5$

(d) Check all the statements below that are consistent with running the EM algorithm with our initialization.
( ) The parameters $q(y)$ would remain as initialized.
( ) None of the model parameters would change after the first M-step.
( ) The property $q_j(x_j = 1 \mid y = 0) = q_j(x_j = 1 \mid y = 1)$, for all j, holds throughout the EM iterations

Solution: Check all 3 brackets

8 Parameterized CPT

Consider the Bayesian network $X \to Y$.

Instead of the usual CPT's we have decided to use the following parametric model with two parameters for this problem:

$$\begin{aligned}\Pr(X=1) &= \alpha\end{aligned}$$

$$\begin{aligned}\Pr(Y=0|X=0) &= \theta\\ \Pr(Y=1|X=0) &= 1-\theta\\ \Pr(Y=0|X=1) &= 1-\theta\\ \Pr(Y=1|X=1) &= \theta\end{aligned}$$

Given the following data $\mathcal{D}$:

X	Y	#occ
0	0	3
0	1	5
1	0	6
1	1	2

(a) Give an expression for $\Pr(\mathcal{D}|\alpha,\theta)$.

Solution: The likelihood of the data is:

$$\Pr(\mathcal{D};\alpha,\theta) = \Pr(X=0,Y=0)^3 \Pr(X=0,Y=1)^5 \Pr(X=1,Y=0)^6 \Pr(X=1,Y=1)^2$$

Taking the log and expanding the definition of the terms (we have that $\Pr(X,Y) = \Pr(X)\Pr(Y \mid X)$) we get

$$\begin{aligned}\log\Pr(\mathcal{D};\alpha,\theta) &= 3\log\Pr(X=0,Y=0) + 5\log\Pr(X=0,Y=1)\\ &\quad + 6\log\Pr(X=1,Y=0) + 2\log\Pr(X=1,Y=1)\\ &= 3[\log(1-\alpha)+\log\theta] + 5[\log(1-\alpha)+\log(1-\theta)]\\ &\quad + 6[\log\alpha+\log(1-\theta)] + 2[\log\alpha+\log\theta]\\ &= 8\log(1-\alpha) + 8\log\alpha + 11\log(1-\theta) + 5\log\theta\end{aligned}$$

(b) For what value of α is it maximized?

Solution:

$$\alpha = 8/16 = 1/2$$

Take the derivative of the log likelihood wrt to α, set to zero and solve for α.

This is also clear from counting: $\alpha = \Pr(X=1)$ and half the data has $X=1$.

(c) For what value of θ is it maximized?

Solution:

$$\theta = 5/16$$

Take the derivative of the log likelihood wrt to θ, set to zero and solve for θ.

(d) Give a numerical example of a probability distribution that can't be well modeled with this parameterization.

Solution: Note that the network structure does not imply any conditional independence assumptions. $\Pr(X, Y) = \Pr(X)\Pr(Y \mid X)$ is always true. However, we have specified $P(Y|X)$ using only one parameter when, in general, it would take two parameters to specify an arbitrary CPT with one parent. So, for example, $P(Y = 1|X = 0) = P(Y = 1|X = 1)$ isn't well modeled with the given parameterization (unless $\theta = 0.5$).

9 What's up, doc?

Consider the simple topic model in which:

- Words are drawn from a vocabulary $\mathcal{W}$;
- Each document d is a sequence of words $w_1^d, \ldots w_{n_d}^d$;
- The distribution of words depends on the topic; for a particular topic $z = \{1 \ldots k\}$, the word distribution is a multinomial: $\Pr(w \mid z, \theta) = \theta_{w|z}$, $\sum_{w \in \mathcal{W}} \theta_{w|z} = 1$ for all z;
- The distribution of topics is a multinomial : $\Pr(z \mid \theta) = \theta_z$, $\sum_{z=1}^{k} \theta_z = 1$;
- The document can be viewed as being drawn from a mixture

$$\Pr(d \mid \theta) = \sum_{z=1}^{k} \Pr(z \mid \theta)\Pr(d \mid z, \theta) = \sum_{z=1}^{k} \theta_z \prod_{w \in d} \theta_{w|z} \ ,$$

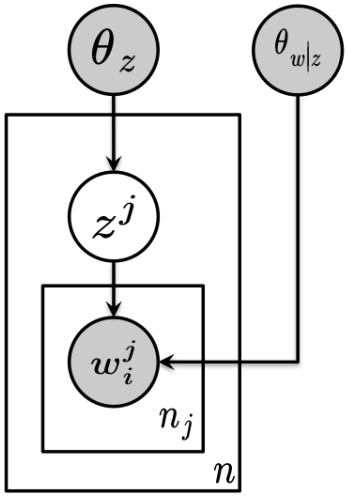
in which there is a single topic per document.

(a) Draw the graphical model corresponding to this model. Here as it is inconvenient to write down all z^j and w_i^j in a graphical model, we therefore introduce a notation called *plate*. For more reference about *plate*, please look at Section 8.1.1 in Bishop book.

Solution: Given the parameters, namely the topic distribution θ_z for $z = 1, \ldots, k$, and the topic-specific word distributions $\theta_{w|z}$ for $w \in \mathcal{W}$ and $z = 1, \ldots, k$. The generative process for a document (say d^j) with n_d words is formalized as

$$z^j \sim \theta_z, \text{ and } w_i^j \sim \theta_{w|z^j}, \ \forall i = 1, \ldots, n_d.$$

The graphical model is shown in the following figure:



(b) What is the complete log likelihood, $\Pr(D, z \mid \theta)$, for a set of documents $D = \{d^1 \ldots d^n\}$?

Solution: The complete log-likelihood is given by

$$\begin{aligned} p(D, Z \mid \theta) &= \sum_{j=1}^{n} \left(\log p(z^j \mid \theta_z) + \sum_{i=1}^{n_j} \log p(w_i^j \mid z^j; \theta_{w|z}) \right) \\ &= \sum_{j=1}^{n} \log \theta_{z^j} + \sum_{j=1}^{n} \sum_{i=1}^{n_j} \log \theta_{w_i^j|z^j}. \end{aligned}$$

Let h_w^j be the number of times the word w appears in the document d^j. We can write

$$\sum_{i=1}^{n_j} \log \theta_{w_i^j|z^j} = \sum_{w \in \mathcal{W}} h_w^j \log \theta_{w|z^j}.$$

As a result, we can rewrite the complete log-likelihood as

$$p(D, Z \mid \theta) = \sum_{j=1}^{n} \log \theta_{z^j} + \sum_{j=1}^{n} \sum_{w \in \mathcal{W}} h_w^j \log \theta_{w|z^j}.$$

This form is easier to work with in deriving the E-M steps.

(c) Describe the E and M steps for an EM algorithm that estimates the θ values given D. Give these expressions explicitly.

Solution: In E-steps, we derive the posterior distribution of the latent variable z given both the parameters θ and the documents in D. From the graphical model, it is not difficult to see that z^j are conditionally independent from the topic labels of other documents,

and thus its posterior distribution is

$$p(z^j = z \mid d^j; \theta) \propto \theta_z \prod_{w \in \mathcal{W}} (\theta_{w|z})^{h_w^j}.$$

Let $q^j(z) \triangleq p(z^j = z \mid d^j; \theta)$. Then their values need to be normalized such that $\sum_{z=1}^{k} q^j(z) = 1$.

In M-steps, given Z and D, we are to solve the parameters θ by maximizing the expected complete log-likelihood, given by

$$L(\theta) = \sum_{j=1}^{n} \sum_{z=1}^{k} q^j(z) \log \theta_z + \sum_{j=1}^{n} \sum_{w \in \mathcal{W}} \sum_{z=1}^{k} q^j(z) h_w^j \log \theta_{w|z}.$$

Then, θ_z and $\theta_{w|z}$ can be solved under the following constraints:

$$\sum_{z=1}^{k} \theta_z = 1, \quad \sum_{w \in \mathcal{W}} \theta_{w|z} = 1, \quad \theta_z \geqslant 0, \quad \text{and} \quad \theta_{w|z} \geqslant 0.$$

The solution is given by

$$\hat{\theta}_z = \frac{1}{n} \sum_{j=1}^{n} q^j(z), \quad z = 1, \ldots, k.$$

and

$$\hat{\theta}_{w|z} = \frac{\sum_{j=1}^{n} q^j(z) h_w^j}{\sum_{j=1}^{n} q^j(z) n_j}, \quad z = 1, \ldots, k, \;\; w \in \mathcal{W}.$$