Probabilistic Modelling and Reasoning
Reference Solution for Assignment 2014

Instructor: Prof. Amos Storkey

March 12, 2014

1. Inference in a Belief Network

(a) Joint distribution `(8 marks)`

$$P(D,C,S,J,I,V,T) = P(D)P(C|D)P(S)P(J|D)P(I|C)P(V|C,S,J)P(T|J) \tag{1}$$

(b) Computation of $P(S = \boldsymbol{stuck}|V = \boldsymbol{static}, T = \boldsymbol{cold})$ `(8 marks)`

Using the product rule, we have:

$$P(S = \textit{stuck}|V = \textit{static}, T = \textit{cold}) = \frac{P(S = \textit{stuck}, V = \textit{static}, T = \textit{cold})}{P(V = \textit{static}, T = \textit{cold})} \tag{2}$$

We can use the sum rule to expand the denominator:

$$P(S = \textit{stuck}|V = \textit{static}, T = \textit{cold}) = \frac{P(S = \textit{stuck}, V = \textit{static}, T = \textit{cold})}{\sum_S P(S, V = \textit{static}, T = \textit{cold})} \tag{3}$$

Therefore to find $P(S = \textit{stuck}|V = \textit{static}, T = \textit{cold})$ we need to evaluate the distribution in the denominator, $P(S, V = \textit{static}, T = \textit{cold})$. Using the sum rule, we have:

$$P(S, V = \textit{static}, T = \textit{cold}) = \sum_D \sum_C \sum_J \sum_I P(D, C, S, J, I, V = \textit{static}, T = \textit{cold}) \tag{4}$$

Plugging (1) into (4), we get:

$$P(S, V = \textit{static}, T = \textit{cold}) = \\ \sum_D \sum_C \sum_J \sum_I P(D)P(C|D)P(S)P(J|D)P(I|C)P(V = \textit{static}|C, S, J)P(T = \textit{cold}|J) \tag{5}$$

Notice that the object defined by (5) is a table of values defined over S. To efficiently compute the elements of this table, we re-arrange the terms and push the sums to the right:

$$\sum_D \sum_C \sum_J \sum_I P(D)P(C|D)P(S)P(J|D)P(I|C)P(V = \mathit{static}|C, S, J)P(T = \mathit{cold}|J) \tag{6}$$

$$= \sum_D \sum_C \sum_J P(D)P(C|D)P(S)P(J|D)P(V = \mathit{static}|C, S, J)P(T = \mathit{cold}|J) \underbrace{\left(\sum_I P(I|C)\right)}_{=1} \tag{7}$$

$$= \sum_D \sum_C \sum_J P(D)P(C|D)P(S)P(J|D)P(V = \mathit{static}|C, S, J)P(T = \mathit{cold}|J) \tag{8}$$

$$= \sum_D \sum_C \sum_J P(S)P(D)P(J|D)P(T = \mathit{cold}|J)P(C|D)P(V = \mathit{static}|C, S, J) \tag{9}$$

$$= P(S) \sum_D P(D) \sum_J P(J|D)P(T = \mathit{cold}|J) \sum_C P(C|D)P(V = \mathit{static}|C, S, J) \tag{10}$$

If we make use of the distributivity of multiplications over additions, then by nesting and caching the sums we can reduce the number of operations required to evaluate the distribution defined by (10). Specifically, using the notation $m_X(Y) = \sum_X f(X, Y)$ define the intermediary 'messages' as follows:

$$P(S) \sum_D P(D) \sum_J P(J|D)P(T = \mathit{cold}|J) \underbrace{\sum_C P(C|D)P(V = \mathit{static}|C, S, J)}_{\overset{\text{def}}{=} m_C(D,S,J)} \tag{11}$$

$$= P(S) \sum_D P(D) \underbrace{\sum_J P(J|D)P(T = \mathit{cold}|J)m_C(D, S, J)}_{\overset{\text{def}}{=} m_J(D,S)} \tag{12}$$

$$= P(S) \underbrace{\sum_D P(D)m_J(D, S)}_{\overset{\text{def}}{=} m_D(S)} \tag{13}$$

$$= P(S)m_D(S) \tag{14}$$

To recap, we have:

$$P(S, V = \mathit{static}, T = \mathit{cold}) = P(S)m_D(S) \tag{15}$$

$$P(S = \mathit{stuck}|V = \mathit{static}, T = \mathit{cold}) = \frac{P(S = \mathit{stuck})m_D(S = \mathit{stuck})}{\sum_S P(S)m_D(S)} \tag{16}$$

Now we recursively compute the tables $m_C(D, S, J)$, $m_J(D, S)$ and $m_D(S)$:

D	**S**	**J**	$\mathbf{m_C(D, S, J) = \sum_C P(C\|D)P(V = \mathit{static}\|C, S, J)}$
high	stuck	true	$0.5 \times 1 + 0.5 \times 1 = 1$
high	stuck	false	$0.5 \times 1 + 0.5 \times 0.9 = 0.95$
high	free	true	$0.5 \times 1 + 0.5 \times 1 = 1$
high	free	false	$0.5 \times 0 + 0.5 \times 0.8 = 0.4$
low	stuck	true	$0.7 \times 1 + 0.3 \times 1 = 1$
low	stuck	false	$0.7 \times 1 + 0.3 \times 0.9 = 0.97$
low	free	true	$0.7 \times 1 + 0.3 \times 1 = 1$
low	free	false	$0.7 \times 0 + 0.3 \times 0.8 = 0.24$

D	**S**	$\mathbf{m_J(D, S) = \sum_J P(J\|D)P(T = \mathit{cold}\|J)m_C(D, S, J)}$
high	stuck	$0.4 \times 0.1 \times 1 + 0.6 \times 0.9 \times 0.95 = 0.5530$
high	free	$0.4 \times 0.1 \times 1 + 0.6 \times 0.9 \times 0.4 = 0.2560$
low	stuck	$0.3 \times 0.1 \times 1 + 0.7 \times 0.9 \times 0.97 = 0.6411$
low	free	$0.3 \times 0.1 \times 1 + 0.7 \times 0.9 \times 0.24 = 0.1812$

S	$\mathbf{m_D(S) = \sum_D P(D)m_J(D, S)}$
stuck	$0.7 \times 0.5530 + 0.3 \times 0.6411 = 0.57943$
free	$0.7 \times 0.2560 + 0.3 \times 0.1812 = 0.23356$

Finally, we compute $P(S = \mathit{stuck}|V = \mathit{static}, T = \mathit{cold})$ using (16):

$$P(S = \mathit{stuck}|V = \mathit{static}, T = \mathit{cold}) = \frac{P(S = \mathit{stuck})m_D(S = \mathit{stuck})}{\sum_S P(S)m_D(S)} \tag{17}$$

$$= \frac{0.5 \times 0.57943}{0.5 \times 0.57943 + 0.5 \times 0.23356} = \mathbf{0.71271} \tag{18}$$

Comments

In general, the elimination order of variables is important. When summing out a variable, all factors that contain that variable — either to the right or left of the condition bar — must be one of the summands. The question asks the student to compute the probability efficiently. Students who chose to use auxiliary functions such as $m_C(\cdot, \cdot, \cdot)$ to "cache" partial results were rewarded.

(c) Conditional independence (8 marks)

To determine if S is conditionally dependent of any other node given V and T, we see if we can find any $X \in \{D, C, J, I\}$ which satisfies the following condition:

$$S \perp\!\!\!\perp X|\{V, T\} \tag{19}$$

This condition holds only if all paths between between S and X are blocked. A path is said to be blocked when any of the following conditions holds:

1. There is a node ω in the evidence set which is head-to-tail with respect to the path,
2. There is a node ω in the evidence set which is tail-to-tail with respect to the path,

3. There is a node that is head-to-head and neither the node, nor any of its descendants are in the evidence set.

Here, the evidence set is $\{V = static, T = hot\}$. It can be shown that none of the conditions hold. We consider all paths from S to all nodes $X \in \{D, C, J, I\}$:

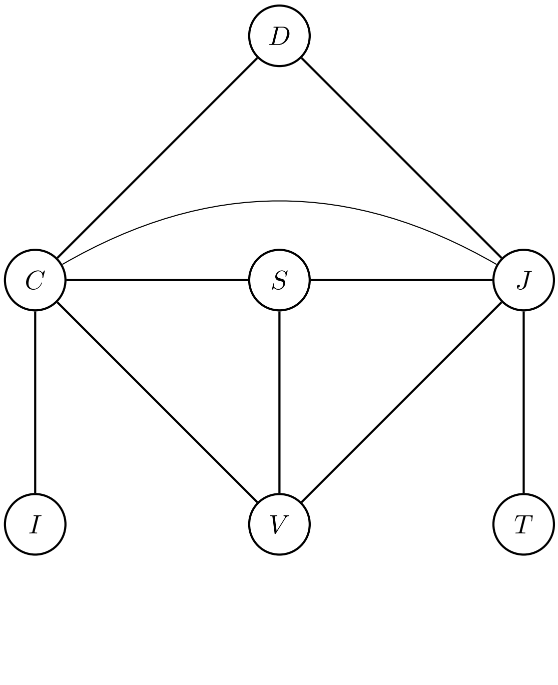
1. The evidence nodes (V and T) are not head-to-tail for any such path.
2. The evidence nodes (V and T) are not tail-to-tail for any such path.
3. Every possible head-to-head node on such a path is in the evidence set, since the only head-to-head node is V.

Therefore, no path between S and X is blocked for any $X \in \{D, C, J, I\}$. This means that S is conditionally dependent on every other node given V and T, and that the observation of any other variable would lead to extra information.

Comments

For a strict application of the rule to determine if a path is blocked, it is required to also state that the descendants (if any) are not in the conditioning set. Many students reason that since a single node on a path is unblocked, the entire path is unblocked too. However, a path is unblocked only if none of its nodes are blocked. In other words, a path is unblocked only if **every node** on the path is unblocked.

(d) Markov Network (8 marks)



2. Mixtures of Multivariate Bernoullis

(a) EM (12 marks)

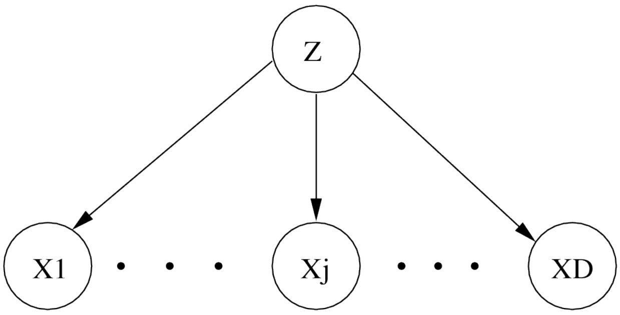


Figure 1: Graphical model of a mixture of Bernoullis

The graphical model of a mixture of Multivariate Bernoullis is shown in Figure 1 and the joint probability of the visible variables is given by:

$$P(\mathbf{x}^i) = \sum_{m=1}^{M} P(Z=m) \prod_{j=1}^{D} P(x_j^i|Z=m)$$

where $\mathbf{x}^i$ is the ith data vector for $i = 1, \ldots n$, D is the number of dimensions (variables), M the number of components of the mixture and n the number of data-points. The parameters of the model are:

$$\begin{aligned} p_j^m &= P(x_j = 1|Z=m), \qquad m = 1, \ldots, M \qquad (20) \\ \pi^m &= P(Z=m), \qquad m = 1, \ldots, M \qquad (21) \end{aligned}$$

which we aim to learn via EM algorithm, where $\sum_{m=1}^{M} \pi^m = 1$. We denote all the parameters as θ.

The *expected complete data log-likelihood* can be expressed as:

$$Q(\theta|\theta^{old}) = \sum_{i=1}^{n}\sum_{k=1}^{M} P(Z^i = k|\mathbf{x}^i, \theta^{old})[\log P(\mathbf{x}^i|Z=k,\theta) + \log \pi^k]. \qquad (22)$$

- **Estimating π^m:** For this purpose we have to maximise Equation 22 wrt π^m subject to the constraint $\sum_{k=1}^{M} \pi^k = 1$. Using Lagrange multipliers we define the function:

  $$\sum_{i=1}^{n}\sum_{k=1}^{M} P(Z^i = k|\mathbf{x}^i, \theta^{old}) \log \pi^k + \lambda(\sum_{k=1}^{M} \pi^k - 1)$$

  Differentiating this equation wrt a specific π^m and equating the result to zero we have:

  $$\frac{\sum_{i=1}^{n} P(Z^i = m|\mathbf{x}^i, \theta^{old})}{\pi^m} + \lambda = 0 \qquad (23)$$

  Summing over all possible $k = 1, \ldots, M$ we obtain:

  $$\sum_{i=1}^{n}\sum_{k=1}^{M} P(Z^i = k|\mathbf{x}^i, \theta^{old}) = -\lambda \sum_{k=1}^{M} \pi^k$$

but $\sum_{k=1}^{M} P(Z^i = k|\mathbf{x}^i, \theta^{old}) = 1$ and $\sum_{k=1}^{M} \pi^k = 1$, thus we find that:

$$-n = \lambda$$

Replacing this value back in Equation 23 we finally obtain:

$$\hat{\pi}^m = \frac{1}{n}\sum_{i=1}^{n} P(Z^i = m|\mathbf{x}^i, \theta^{old}). \tag{24}$$

- **Estimating $\mathbf{p}^m$:** Considering

$$P(\mathbf{x}^i|Z = k, \theta) = \prod_{d=1}^{D} (p_d^k)^{x_d^i}(1 - p_d^k)^{1-x_d^i}$$

where $p_d^k = P(x_d = 1|Z = k)$, x_d^i is the dth component of $\mathbf{x}^i$ and D is the dimensionality of the data, equation 22 leads to:

$$\sum_{i=1}^{n}\sum_{k=1}^{M} P(Z^i = k|\mathbf{x}^i, \theta^{old}) \log\left(\prod_{d=1}^{D} (p_d^k)^{x_d^i}(1 - p_d^k)^{1-x_d^i}\right)$$

$$= \sum_i \sum_k P(Z^i = k|\mathbf{x}^i, \theta^{old})\left(\sum_d x_d^i \log p_d^k + (1 - x_d^i)\log(1 - p_d^k)\right). \tag{25}$$

Differentiating wrt p_j^m and equating the results to zero we have:

$$\sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})\left[\frac{x_j^i}{p_j^m} - \frac{1 - x_j^i}{1 - p_j^m}\right] = 0 \qquad \Rightarrow \qquad \sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})[x_j^i - p_j^m] = 0$$

and finally, we obtain:

$$\hat{p}_j^m = \frac{\sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})x_j^i}{\sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})} \tag{26}$$

or in vector notation:

$$\hat{\mathbf{p}}^m = \frac{\sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})\mathbf{x}^i}{\sum_i P(Z^i = m|\mathbf{x}^i, \theta^{old})}.$$

In order to complete and EM iteration it is necessary to define $P(Z^i = m|\mathbf{x}^i, \theta^{old})$. We do this by simply applying Bayes' rule:

$$P(Z^i = m|\mathbf{x}^i) = \frac{P(\mathbf{x}^i|Z = m)P(Z = m)}{P(\mathbf{x}^i)} = \frac{P(Z = m)\prod_j P(x_j^i|Z = m)}{\sum_{k=1}^{M} P(Z = k)\prod_j P(x_j^i|Z = k)} \tag{27}$$

and

$$\prod_j P(x_j^i|Z = m) = \prod_j (p_j^m)^{x_j^i}(1 - p_j^m)^{1-x_j^i}.$$

The parameters π^m and $\boldsymbol{p}^m$ are initialised at random and equations 27, 24 and 26 are iterated until convergence.

Comments: A derivation of the general form of the expected complete data log-likelihood for a mixture model was not expected. However, some students did not explicitly state $Q(\theta|\theta^{old})$ and therefore the estimation of the parameters of the model was not clear. Additionally, students were expected to show the expression for a multivariate Bernoulli distribution and those who did this and those who gave a derivation of p_j^m were rewarded. A common mistake was to try to express the likelihood in terms of quantities like $(\mathbf{p}^m)^{\mathbf{x}^m}$; this kind of expression is best avoided— what is the meaning of raising a vector to the power of a vector? (If it is defined as an elementwise product this works, but it needs to be properly defined.) A small group of students have used a very different notation and did not specify what this notation meant. This makes it very hard to follow nay argument. Other students did not show in detail the full derivation of the parameters of the model based on the EM algorithm. Note that the final parameters were provided both cases. Those that did show the full deviation were rewarded.

(b) Fitting Mixtures of Bernoullis to data (10 marks)

In order to evaluate the number of components of the mixture one can apply the EM algorithm with the default values of the parameters, i.e. `tol` = `0.0001`, `max_it` = `400`, `P0` initialised at random and `Pi0` initially 1/M. However, as the results vary depending upon the initialisation of `P0`, it is better to execute the algorithm a certain number of times R. The likelihood of the data for $M = 1, 2, .., 5$ is shown in Figure 2 for $R = 10$. It can be observed in this figure that the likelihood increases as we augment the number of components, which seems not to fit with the true generative model. Nonetheless, one must be cautious with this interpretation because it may be a sign of *overfitting* rather than an indication of that the best model occurs when $M = 5$. Indeed, as we increase the number of components, it is necessary to estimate a greater number of parameters. For example, for $M = 3$ we need to estimate 21 parameters while for $M = 5$ the number of estimations needed are 35. It stands to reason that $n = 200$ data-points might be insufficient for estimating this number of parameters without the risk of overfitting.

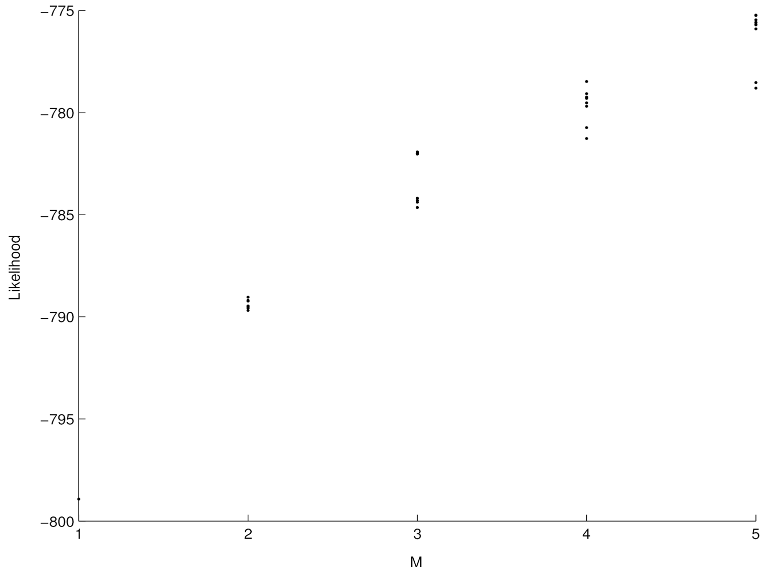


Figure 2: Likelihood of the data for different number of mixtures

An interesting fact to notice at this point is that when using the default parameters the algorithm terminates due to `code` = `0`, i.e. the tolerance has been reached for all of them.

The results of using `tol` = `10`$^{-6}$ and selecting the best of $R = 10$ runs are shown in Table 1. Although for $M = 4$ and $M = 5$ there is an increase in the log-likelihood of the data the number of iterations required is larger than for $M = 3$. It is also possible to compare the parameters of the models found by EM algorithm with the actual parameters of the true model that generated the data. We might expect these parameters

M	Log-likelihood	iterations
1	-798.92	0
2	-788.45	153
3	-781.26	100
4	-777.39	176
5	-774.04	160

Table 1: Log-likelihood and number of iterations for different number of mixtures

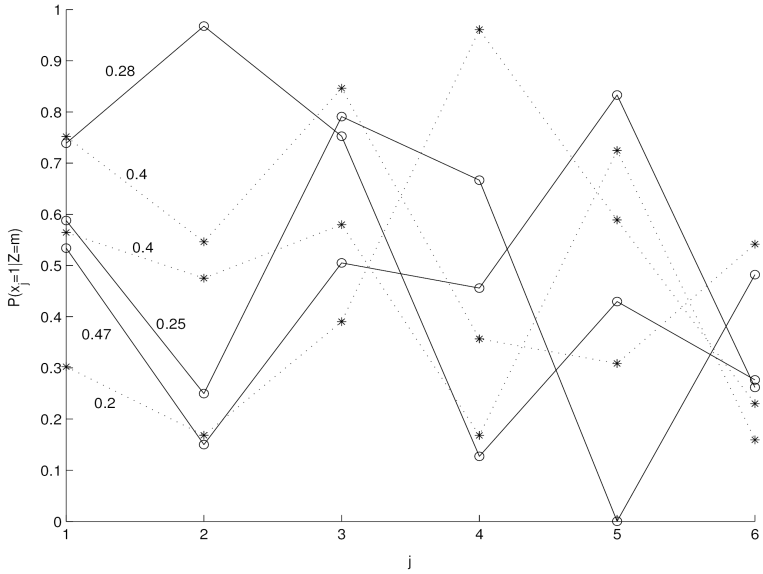


Figure 3: Parameter vectors for EM solution 'o' and true model '*' $M = 3$.

to be similar since after each EM iteration we are maximising the log-likelihood of the data. However, they do not look quite similar even when decreasing the tolerance of the algorithm and augmenting the number of iterations. Figure 3 shows the parameter vectors for the EM solution and the true model in the case of $M = 3$, where each line describes a parameter vector and is labelled with the corresponding probability of the component. The agreement between the parameters found by EM and the true generative parameters is not good. Additionally, the priors found by the algorithm [0.47 0.25 0.28] do not accurately correspond to those of the real model [0.4 0.2 0.4]. There are two possible reasons that can explain these differences. First, the weakness of EM algorithm of getting trapped in a local maxima. Indeed, there is no guarantee that EM can obtain the solution that in fact corresponds to the global maxima of the likelihood. However, the second reason can be that the data sample available may be insufficient to accurately reflect the underlying mixture model. As the log likelihood of the 200 datapoints is less under the true generative model (-790) than under the learned model it seems that the second reason applies here.

Comments: Students were expected to run the algorithm several times for each value of M because different answers may be obtained each time. It is not correct to compare the results based on a single execution of the program for each M. A plot like Figure 2 is an easy and illustrative way of showing the results and their variability. It is important to recognise the potential of *overfitting* and it should have been observed through the analysis of the results that if we increase the number of components overfitting is taking place. Some students used a chi-squared test to analyse the results, though in most cases did not justify this.

Finally, some students tried the nice idea of generating more data from the given mixture model using `mix_bernoulli_sample`; this allowed them to verify that overfitting was taking place for higher values of M.

(c) Naïve Bayes vs. Mixtures of Bernoullis `(10 marks)`

- **Estimating the parameters for Naïve Bayes**: Based on the principle of maximum likelihood the parameters can be estimated as follows:

$$\theta_k^c = P(x_k = 1|c) = \frac{\#(x_k = 1)}{N_c} \tag{28}$$

$$P(class = c) = \frac{N_c}{N} \tag{29}$$

where $\#(x_k = 1)$ stands for the number of times attribute x_k is equal to 1, N_c is the number of data-points belonging to class c and N is the total number of data-points.

- **Parameter vectors for each class:** The probability of each class ham or spam is 0.5 given that they are in the same proportion in the data. The following table shows the estimators $P(x_k = 1|c)$ for each class.

θ^{ham}	0.00	0.00	0.00	0.00	0.28	0.00	0.32	0.22	0.26	0.22
θ^{spam}	0.56	0.44	0.40	0.22	0.08	0.26	0.00	0.00	0.12	0.18

- **Computing $\boldsymbol{P(Z^i = ham|x^i)}$:** Let us denote class 1 as ham, $\theta_d^{c1} = P(x_d = 1|c_1)$ and $p_1 = P(class = ham)$. Similarly let class 2 be spam, $\theta_d^{c2} = P(x_d = 1|c_2)$ and $p_2 = P(class = spam)$. Simple application of Bayes' rule produces:

$$P(c = 1|X) = \frac{p_1 \prod_{d=1}^{D} (\theta_d^{c1})^{x_d} (1 - \theta_d^{c1})^{1-x_d}}{p_1 \prod_{d=1}^{D} (\theta_d^{c1})^{x_d} (1 - \theta_d^{c1})^{1-x_d} + p_2 \prod_{d=1}^{D} (\theta_d^{c2})^{x_d} (1 - \theta_d^{c2})^{1-x_d}}$$

The corresponding matlab code is given by:

```
for i=1:N
   pham_X  =  prod( (theta1.^X(i,:)) .* ((1-theta1) .^ (1-X(i,:))) )*p1;
   pspam_X =  prod( (theta2.^X(i,:)) .* ((1-theta2) .^(1-X(i,:))) )*p2;
   pham(i)=pham_X/(pham_X+pspam_X);
end
```

The posterior probabilities are shown in Figure 4.

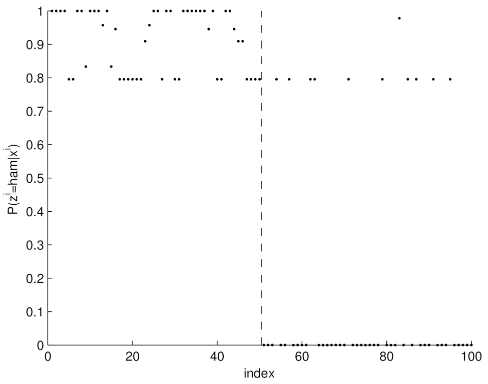


Figure 4: Classification with Naive Bayes

If we did not have the labels we can fit a Mixture of Bernoullis to the data with $M = 2$ and expect the data to be naturally grouped in two clusters. As before, several runs, say $R = 10$ should be used and the parameters found by the one with maximum likelihood should be chosen. The calculation of the posterior probabilities is the same as in the previous case. The matlab code is shown below.

```
for i=1:N
   p1_X  =  prod( (P(1,:).^X(i,:)) .* ((1-P(1,:)) .^(1-X(i,:))) )*Pi(1,1);
   p2_X  =  prod( (P(2,:).^X(i,:)) .* ((1-P(2,:)) .^(1-X(i,:))) )*Pi(2,1);
   p_c(i)=p_1_X/(p_1_X+p_2_X);
end
```

Figure 5 shows the posterior probabilities for the Mixture of Bernoullis, where it is clear that these probabilities correspond to $P(Z^i = spam|\mathbf{x}^i)$.

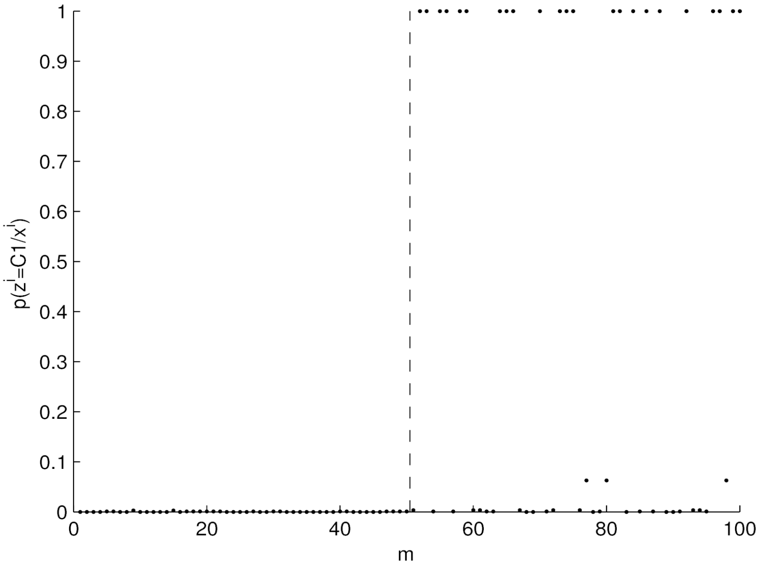


Figure 5: Classification with Mixture of Bernoullis

- **Goodness of fit**
  As can be seen in the plots both methods correctly classify all the ham. However, some misclassifications occur when deciding about spam. The following table summarises the log-likelihood and the error rate for each method. It is not surprising that the error rate for the Mix-Bernoulli method is greater than for Naïve Bayes as the learning for the former was completely unsupervised i.e. without knowledge of the labels. In fact, it is doing quite well considering that none of the labels were used during training. It is also not surprising that the Mix-Bernoulli method exhibits a better log-likelihood than Naïve Bayes. This is because the EM algorithm is completely driven by what the data says regardless of the labels each example has and tries to maximise the likelihood at each iteration.

Method	Log-likelihood	error rate
Naïve Bayes	-400.25	11/100
Mix-Bernoulli	-369.15	27/100

Comments: There were some misapplications of Bayes' rule. When calculating the posterior probabilities the prior over the class was neglected or the normalisation by dividing over $P(\mathbf{x}^i)$ was not done. Those who applied Bayes rule correctly where rewarded. A misapplication of Bayes rule is fairly problematic, and so unfortunately did not attract many marks. Other students recognised that they were not normalising as $P(\mathbf{x}^i)$ is the same for both classes but the question asks to see plots of the actual posterior probabilities. When comparing the methods, at least the error rate should have been mentioned. The calculation of the log-likelihood (also important and a matter of discussion), was used only by a few number of people. Finally, some students analysed the results further and found out that several data-points were having zero for all the variables and that indeed they were misclassified by both algorithms.

3. Gaussians

$$P(\mathbf{x},\mathbf{y}) = \frac{1}{Z(W)} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x} - \frac{1}{2}\mathbf{y}^T\mathbf{y} - \mathbf{x}^T W \mathbf{y}\right) \tag{30}$$

where $\mathbf{x}$ is a real vector of dimension m_x, $\mathbf{y}$ is a real vector of dimension m_y and $m_x = m_y$.

3a: Let $\mathbf{r} = (\mathbf{x}^T, \mathbf{y}^T)^T$. What is the inverse covariance matrix (called the precision) of $\mathbf{r}$?

Note that $\mathbf{r}$ is zero mean. How do we know? If we transform $\mathbf{x} \to -\mathbf{x}$ and $\mathbf{y} \to -\mathbf{y}$ we get the same form. So the distribution is symmetric about zero and so must have zero mean.

So we are looking for a distribution of the form

$$\frac{1}{Z}\exp\left(-\frac{1}{2}\mathbf{r}^T\Sigma^{-1}\mathbf{r}\right)$$

because if we have that form we can just read off the precision matrix (inverse covariance).

Write this out with $\mathbf{r} = (\mathbf{x}^T, \mathbf{y}^T)^T$:

$$\frac{1}{Z}\exp\left(-\frac{1}{2}\begin{pmatrix} \mathbf{x}^T & \mathbf{y}^T \end{pmatrix}\begin{pmatrix} \Sigma^{-1}_{xx} & \Sigma^{-1}_{xy} \\ \Sigma^{-1}_{yx} & \Sigma^{-1}_{yy} \end{pmatrix}\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}\right)$$

where we have split Σ^{-1} up into blocks.

Now we can expand the matrix bits out

$$\frac{1}{Z}\exp\left(-\frac{1}{2}\left[\mathbf{x}^T\Sigma^{-1}_{xx}\mathbf{x} + \mathbf{x}^T\Sigma^{-1}_{xy}\mathbf{y} + \mathbf{y}^T\Sigma^{-1}_{yx}\mathbf{x} + \mathbf{y}^T\Sigma^{-1}_{yy}\mathbf{y}\right]\right)$$

Now we can simply match terms in this to terms in the given form ensuring symmetry (each term is a linear parameter of a different set of variables). So $\Sigma^{-1}_{xx} = I$, $\Sigma^{-1}_{yy} = I$, $\Sigma^{-1}_{xy} = W$, $\Sigma^{-1}_{yx} = W^T$, giving

$$\Sigma^{-1} = \begin{pmatrix} I & W \\ W^T & I \end{pmatrix}$$

3b: Write down an expression for $Z(W)$ in terms of W. Note that for determinants $|.|$, we have $|A^{-1}| = 1/|A|$. You may wish to look up the "determinant of block matrices".

The determinant of a block matrix

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(AD - BC)$$

if $CD = DC$.

$$Z(\mathbf{W}) = (2\pi)^{d/2}|\Sigma|^{1/2} = \frac{(2\pi)^{d/2}}{|\Sigma^{-1}|^{1/2}} = \frac{(2\pi)^{d/2}}{|I - W^T W|^{1/2}} = \frac{(2\pi)^{d/2}}{|I - WW^T|^{1/2}}$$

3c: Write out the forms for the distributions $P(y_i|\mathbf{y}_{rest}, \mathbf{x})$ and $P(x_i|\mathbf{x}_{rest}, \mathbf{y})$, where the $rest$ suffix denotes the set of all the other nodes apart from the i-th node.

$$P(y_i|y_{rest}, x) = \frac{1}{P(y_{rest}, x)} P(y_i, y_{rest}, x)$$

where $P(y_{rest}, x) = \sum_{y_i} P(y_i, y_{rest}, x)$.

$$P(y_i, \mathbf{y}_{rest}, x) = \frac{1}{Z} \exp\left(-\frac{1}{2}\left[y_i^2 + \mathbf{y}_{rest}^T \mathbf{y}_{rest} + 2y_i \mathbf{w}_i^T \mathbf{x} + 2y_{rest}\mathbf{W}_{rest}\mathbf{x} + \mathbf{x}^T\mathbf{x}\right]\right) \tag{31}$$

$$= \frac{1}{Z} \exp\left(-\frac{1}{2}(y_i - \mathbf{w}_i^T \mathbf{x})^2 - f(\mathbf{x}, \mathbf{y}_{rest})\right) \tag{32}$$

where we use $\mathbf{w}_i$ to denote the ith column of $\mathbf{W}$ and $\mathbf{W}_{rest}$ to denote the remainder of the matrix. Note the terms collected into f don't depend on y_i and so they (and Z) cancel when the normalisation $P(y_{rest}, x)$ is computed. So

$$P(y_i|\mathbf{y}_{rest}, x) \propto \exp\left(-\frac{1}{2}(y_i - \mathbf{w}_i^T \mathbf{x})^2\right) \tag{33}$$

What is the normalisation constant? Well it is just a Gaussian distribution so we know the constant...

$$P(y_i|\mathbf{y}_{rest}, x) = \frac{1}{(2\pi)^{1/2}} \exp\left(-\frac{1}{2}(y_i - \mathbf{w}_i^T \mathbf{x})^2\right) \tag{34}$$

3d: Draw the Markov Network for this distribution. How does the Markov Network relate to the precision matrix?

The terms 'connected' in the Energy Function (the terms in the exponential) turn up in the Markov network. That is if any factor if a (nonadditive) function of two random variables then those are directly dependent and connected in the Markov Network. For a Gaussian, the connected terms are equivalent to terms that are non-zero in the precision matrix. The Markov Network is a bipartite structure with connections only between hidden and visible neurons.

3e: Suppose the x data are observable. What is the form of the joint conditional distribution $P(\mathbf{y}|\mathbf{x})$?

From the Markov Network, conditioned on $\mathbf{x}$ each y_i is independent. So the joint conditional is

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x})$$

where each term is given by the result of 3c.

3f: You observe data $D = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \dots, \mathbf{x}^N\}$. Explain how you could use the EM algorithm to learn the parameters W. Give the form of the *E*-Step and the *M*-Step updates.

The EM algorithm first computes the distribution over the latent space and then uses that distribution as 'surrogate data' for learning the parameters. The E step is given by the 3c.

The M Step then optimizes the expected log likelihood: it optimizes using the gradient

$$-\frac{1}{2}\sum_{n=1}^{N}\sum_{y_j^n} P(y_j^n|\mathbf{x}^n)x_i^n y_j^n + \frac{N}{2}\frac{\partial}{\partial w_{ij}}\log|I - WW^T|$$

That is all that is necessary for a full answer to this question. But we can do the sums:

The derivative of $(\log\det B(A))$ wrt A_{ij} is $Tr(B^{-1}\frac{\partial B}{\partial A_{ij}}$. So the gradient is

$$-\frac{1}{2}\sum_{n=1}^{N}\left[\sum_{y_j^n} P(y_j^n|\mathbf{x}^n)x_i^n y_j^n\right] + N(I - \mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}$$

Note that this is not a closed form solution and so a gradient update is needed for the M step.

3g: Let $\mathbf{v} = \mathbf{W}\mathbf{y}$ and assume that W is invertible. Draw the Markov Network for the variables $\mathbf{x}, \mathbf{v}$ in the case of four x nodes and four v nodes.

Rewrite in terms of $\mathbf{v}$ using $\mathbf{y} = \mathbf{W}^{-1}\mathbf{v}$

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z(W)}\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x} - \frac{1}{2}\mathbf{v}^T(\mathbf{W}^{-1})^T\mathbf{W}^{-1}\mathbf{v} - \mathbf{x}^T\mathbf{v}\right) \quad (35)$$

so now all the $\mathbf{v}$ terms are connected, but y_i is only dependent on v_i. The Markov network has links between all the v nodes, but single links from each v node to the corresponding x node.

3h: Let the Singular Value Decomposition (SVD) of W be $W = U\Lambda V^T$, where $UU^T = I$, $VV^T = I$ and Λ is diagonal. Now let $\mathbf{u} = V^T\mathbf{y}$. What is the Markov Network for u and x?

Rewrite again in terms of $\mathbf{u}$.

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z(W)} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x} - \frac{1}{2}\mathbf{u}^T\mathbf{u} - \mathbf{x}^T\mathbf{U}\boldsymbol{\Lambda}\mathbf{u}\right) \tag{36}$$

3i: Compare this network with the network in question (d). Comment about the uniqueness of the EM solution in (e). Explain your reasoning.

This has the same graphical form as the original graph (bipartite). It also has the same parameter structure, but with a different parameter value $\mathbf{W} \to \mathbf{U}\boldsymbol{\Lambda}$. This means there are no unique parameters and the model is not identifiable. The parameters learnt by EM are just one set of parameters in an equivalence class.

3j: What is the *marginal* distribution over the x variables integrating out the y variables?

As stated in the question, The covariance matrix for $\mathbf{r}$ takes the form:

$$cov(\mathbf{r}) = \begin{bmatrix} [I - WW^T]^{-1} & -W(I - W^TW)^{-1} \\ -W^T(I - WW^T)^{-1} & [I - W^TW]^{-1} \end{bmatrix}$$

This can be derived using the partitioned inverse equations (sometimes called the block inverse equations).

So the marginal distribution over $\mathbf{x}$ is just a Gaussian with the relevant covariance read from the covariance matrix: $N(0, [I - WW^T]^{-1})$

3k: If we just want to learn a model for P(x), and don't care about y, can you suggest a better way of learning this marginal distribution?

$I - WW^T$ (and hence its invers) is a full rank symmetric matrix. So the distribution over the visibles is just a general zero mean Gaussian. We could just learn this distribution directly. I.e. set the covariance matrix to match the empirical covariance.

3l: If y were instead binary variables ($y_i \in \{0, 1\}$), with x and y still following the model described in equation (1), explain why the distribution over x would be a mixture of Gaussians. How many mixture components would there be (in general). What would each component have in common?

Given the state of the hidden units, the visible is a multivariate Gaussian $N(\mu, I)$. Different states of the hidden units lead to differen μ. Hence summing over them results in a mixture. there are 2^{m_y} states of the hidden nodes, resulting 2^{m_y} mixture components, all with unit covariance.