CS340 Fall 2006: Homework 4

Out Mon 2 Oct, back Wed 11 Oct

1 Bernoulli distributions

Let $X \in \{0, 1\}$ be a binary random variable (e.g., a coin toss). Suppose $p(X = 1) = \theta$. Then

$$p(x|\theta) = \text{Be}(X|\theta) = \theta^x (1-\theta)^{1-x} \tag{1}$$

is called a **Bernoulli** distribution. Prove the following facts:

$$\begin{aligned} E[X] &= p(X=1) = \theta & (2) \\ \text{Var}\,[X] &= \theta(1-\theta) & (3) \end{aligned}$$

2 Setting hyper-parameters for the beta distribution

1. Let $\theta \sim Be(a, b)$. (In class, we called $a = \alpha_1$ and $b = \alpha_0$.) Sometimes our prior knowledge is not in the form of pseudo counts, so it is not immediately clear how to set a and b. For example, suppose you believe that $E\theta = m$ and Var $\theta = v$. Use the following properties of the Beta distribution to solve for a and b in terms of m and v.

$$\begin{aligned} E\theta &= m = \frac{a}{a+b} & (4) \\ \text{Var}\,\theta &= v = \frac{m(1-m)}{a+b+1} = \frac{ab}{(a+b)^2(a+b+1)} & (5) \end{aligned}$$

2. Let θ represent the proportion of adults in New York who support the death penalty. Suppose θ is beta with mean 0.7 and standard deviation 0.2. What are the values of the hyper-parameters a and b that correspond to this?

3. A random sample of 1000 adults in New York is taken, and 62% support the death penalty. Plot the prior $p(\theta)$ and posterior $p(\theta|D)$. What is the posterior mean and variance? What is the **95% posterior credible interval**? i.e., find values $\theta_{2.5}$ and $\theta_{97.5}$ such that

$$p(\theta_{2.5} < \theta < \theta_{97.5}|D) = 0.95 \tag{6}$$

(This is a Bayesian version of a confidence interval.) Hint: use the function `betainv` in the statistics toolbox.

3 Marginalizing a Dirichlet

Suppose

$$\begin{aligned} p(\theta_1, \theta_2, \theta_3) &= Dir(\alpha_1, \alpha_2, \alpha_3) & (7) \\ &\propto \theta_1^{\alpha_1 - 1}\theta_2^{\alpha_2 - 1}(1 - \theta_1 - \theta_2)^{\alpha_3 - 1} & (8) \end{aligned}$$

since $\theta_1 + \theta_2 + \theta_3 = 1$. Derive an expression for

$$p(\theta_1) = \int_0^{1-\theta_1} p(\theta_1, \theta_2) d\theta_2 \tag{9}$$

You may ignore (cancel) any normalizing constants in your final answer. But you should identify the functional form of the marginal, along with its parameters.
Hint 1: make the substitution $u = \frac{\theta_2}{1-\theta_1}$ so $1 - u = \frac{1-\theta_1-\theta_2}{1-\theta_1}$. The answer should be another Dirichlet. (In fact, it will be a beta distribution, which is just a special case of Dirichlet.)
Hint 2: Use the fact that

$$\int_0^1 x^{a-1}(1-x)^{b-1} dx = B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{10}$$

since this is the normalization constant of the beta distribution.
Hint 3: Using the above substitutions, rewrite the integral as

$$p(\theta_1) \propto \theta_1^a (1-\theta_1)^b \int_0^1 u^c (1-u)^d du \tag{11}$$

for suitably chosen a, b, c, d. (You must derive this equation and work out what these values are!)

4 Naive Bayes for document classification

Consider the problem of classifying email messages posted to online discussion boards into one of two classes, one for users of X Windows (class 1) and another for users of microsoft Windows (class 2). (This is analogous to spam filtering.) There are 900 documents from each class; we divided them into training and text sets of equal size. To save space (and time), we ran word detection on the documents, and the data available to you consist of binary feature vectors for each document. Upon loading `docdata.mat` (on the web page), the Matlab environment will contain variables `xtrain,xtest,ytrain,ytest`. You can visualize this data as an image as follows

```
load docdata
spy(xtrain);colormap(gray);xlabel('words');ylabel('documents');title('training set')
```

The result is shown in Figure 1. (The command `spy` is useful for plotting **sparse matrices**. One can also use `imagesc` for plotting matrices.) The first 450 lines correspond to the class $y = 1$, the second to $y = 2$ (this information is stored in the `ytrain` vector). Can you see a difference in the patterns between the first 450 rows and the second 450 rows?
To identity of the 600 words is stored in the file 'words.txt' (on the web); you can load it into matlab by saying

```
vocab = textread('words.txt','%s');
```

`vocab` is a **cell array**, so `vocab{t}` is the t'th word. You can print out the first 10 words using

```
for t=1:10
   fprintf(2,'%2d %20s\n', t,  vocab{t});
end
```

which produces the list in Figure 2.

1. Implement the following function

```
function theta = NB_train(X,Y)
% Bayes estimate of Naive Bayes parameters
% Input:
% X(n,d) = 1 if word d appears in document n, otherwise X(n,d)=0
% Y(n) = class label of doc n (assumed to be 1 or 2)
% Output:
% theta(c,d) = P(Wd=1|Y=c) = probability of word d appearing in class c
```

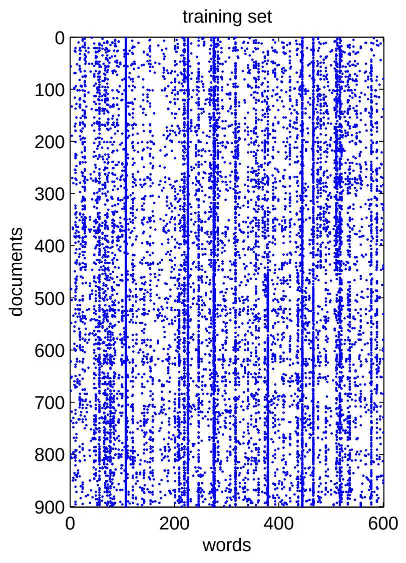

Figure 1: Training data for document classification visualized using matlab's `spy` command, which is useful for sparse matrices. A black dot in row i column j means document i contains word j at least once.

1	straight
2	magazines
3	issues
4	ray
5	enabled
6	head
7	improved
8	thread
9	libs
10	working

Figure 2: First 10 words in the vocabulary used for the NB exercise.

which computes the following posterior mean estimate

$$\hat{\theta}_{cd} = \frac{N_{dc} + 1}{N_c + 2} \tag{12}$$

where N_{dc} counts the number of times word d appears in class c, N_c is the total number of documents in class c), and we have assumed a Beta(1,1) prior. Turn in your code.

2. Implement a function to classify each document, assuming uniform class priors $p(Y = 1) = p(Y = 2) = 0.5$.

```
function y = NB_apply(X,theta)
% X(n,:) is a bit vector for document n
% theta(c,d) = prob of word d in class c
% y(n) = most probable class for X(n,:)
```

   Here $y(n) = \arg\max_y p(Y = y|X(n,:))$ is the most probable class label for document n. Since $p(Y = y|\vec{x}) \propto p(\vec{x}|Y = y)$ is a small number, you will need to use logs to avoid underflow. (You don't necessarily need the logsumexp trick, because it suffices to compute the log likelihood $p(\vec{x}|y)$ rather than the normalized posterior $p(y|\vec{x})$, but you will need to use logs somehow!) Turn in your code.

3. Use NB_train on the data in `xtrain,ytrain`. Compute the misclassification rates (i.e., the number of documents that you mis-classified) on the training set (by using NB_apply on `xtrain,ytrain`) and on the test set (by using NB_apply on `xtest,ytest`). Sanity check: You should get test error of **0.1867**.

4. The provided function `er = cv(X,Y,K)` computes the K-fold cross-validation error. (This calls your functions `NB_train` and `NB_apply`.) $K = 1$ means no cross-validation, that is error is simply computed on the whole training set. Use this to compute the 10-fold error rate on the training set. How does this compare to the (non cross validated) training and test error?

5. Plot (as histograms) the class-conditional densities $p(x_d = 1|y = c, \theta_c)$ for classes $c = 1, 2$ and words $d = 1 : 600$.

6. What are the 5 most likely words in each class?

7. It is clear that the most probable words are not very discriminative. One way to measure how much information a word (feature) $X_d \in \{0, 1\}$ conveys about the class label $Y \in \{1, 2\}$ is by computing the **mutual information** between X_d and Y, denoted $I(X_d, Y)$, and defined as

$$mi(d) = I(X_d, Y) = \sum_{x=0}^{1}\sum_{c=1}^{2} p(X_d = x, Y = c) \log \frac{p(X_d = x, Y = c)}{p(X_d = x)p(Y = c)} \tag{13}$$

   If we assume equal class priors, $p(Y = 1) = p(Y = 2) = 0.5$, then

$$p(X_d = 1, Y = c) = p(X_d|Y = y)p(Y = c) = \frac{\theta_{cd}}{2} \tag{14}$$

   Use the provided function `MI.m` to compute the 5 words with the highest mutual information with the class label. (Use the θ's estimated on `xtrain,ytrain`.) List the words along with the corresponding values of `MI`. (As a sanity check, the first word should be "windows" with an MI of 0.2150.)