CSCD11 Machine Learning and Data Mining, Fall 2016

Assignment 3: Classification and Bayesian Methods

Written Part (5%): due Friday, November 4th, 11am

1. Bayesian Prediction [18 marks; 2 marks per part]

Suppose you are attending a conference where each person has an ID badge with a number, indicating the order in which they registered. After seeing a number of people walk by and noting the number on their ID badges, you wonder how many people are there in total? What ID number is likely to be seen next?

To formalize the problem, we assume that all cars in the province are numbered from 1 to L, where L is the largest licence plate number. Let M be the largest possible value of L. To make things simple, we'll assume that license plate numbers are three digits, so that $M = 999$. We assume that all values of L are equally likely, so our prior for L is a uniform distribution from 1 to M. Furthermore, we assume that, when we see a new car, we are equally likely to see any of the L cars out there, so the likelihood of seeing licence plate number X is also uniform. Our observations will be the numbers X_i of the N cars we see go by.

To specify the model, we define

$$f(Z, A, B) = \begin{cases} \frac{1}{B-A+1} & A \leq Z \leq B \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$P(L) = f(L, 1, M) \text{ (the prior)} \tag{2}$$

$$P(X|L) = f(X, 1, L) \text{ (the likelihood of a single license plate number } X) \tag{3}$$

$$P(X_{1:N}|L) = \prod_{i=1}^{N} P(X_i|L) \text{ (the likelihood of observing numbers } X_{1:N}) \tag{4}$$

Additionally, define

$$X_{\max} = \max X_{1:N} \tag{5}$$

to be the largest license plate number observed.

(a) Write the posterior distribution $P(L|X_{1:N})$ using Bayes' Rule, in terms of the uniform distributions above. **Hints**: simplify the numerator first. For the denominator, use the sum rule and the product rule: $P(X_{1:N}) = \sum_{i=1}^{M} P(X_{1:N}, L = i) = \sum_{i=1}^{M} P(X_{1:N}|L = i)P(L = i)$. How does the denominator relate to the numerator?

(b) For what range of values of L will the posterior be non-zero?

(c) For the values of L where the posterior is nonzero, simplify the posterior into a function of only L, M, N, $X_{\max}$, and/or the observations. Hint: begin with the numerator.

(d) Let $M = 999$ and $X_{\max} = 700$. On one set of axes, plot $P(L|X_{1:N})$ for $N = 1$, $N = 10$, and $N = 100$. The horizontal axis of the plot should range from 1 to M. Be sure to label your plot with `xlabel`, `ylabel`, `title`, and `legend`. (As a sanity check, make sure that $\sum P(L = i|X_{1:N}) = 1$, otherwise this estimate won't be meaningful.)

(e) Suppose we want to estimate L from this data. What is the MAP estimate L_{MAP}? Does this estimate seem intuitive, i.e., is it what a person might do?

(f) Another way to determine L is to use the Bayes' estimate, i.e., the posterior mean,

$$L_{\text{mean}} = E[L] = \sum_{i=1}^{M} i\, P(L = i|X_{1:N}) \tag{6}$$

Compute this numerically for $N = 1$, $N = 10$, and $N = 100$. Do these estimates seem more or less reasonable than the MAP estimate?

(g) A third option is to *not* estimate L. Suppose we now wish to describe the probability of the next license plate number X_{N+1} that we might see. Derive a formula for this distribution, i.e., $P(X_{N+1}|X_{1:N})$ in terms of the posterior distribution and the distributions from the model. Your derivation should use only the basic rules of probability theory, i.e., Sum Rule, Product Rule, and Bayes' Rule. You will also need to make use of the independence of X's given L: $P(X_{N+1}|L, X_{1:N}) = P(X_{N+1}|L)$.

(h) Plot, on one set of axes: $P(X_{N+1}|L_{\text{MAP}}), P(X_{N+1}|L_{\text{mean}})$, and $P(X_{N+1}|X_{1:N})$, for the case where $N = 10$, $M = 999$, and $X_{\max} = 700$.

(i) Which plot seems most consistent with your intution about the likelihood of the next number, given what you've seen? Suppose you have seen 10 cars, with $X_{\max} = 700$, and your friend bets you \$10 that the next car will have a license plate less than 750. Do you take the bet? Why or why not?

2. Gaussians [12 marks]

(a) Let $\bar{x} = [x_1, ..., x_D]^T$ be a vector composed of D scalar, independent, zero-mean Gaussian variables $\{x_i\}$, with variances $\{\sigma_i^2\}$. So, the joint density over the variables x_i (i.e., the elements of $\bar{x}$) is the product of their individual densities. Show that this joint distribution can also be written as a multidimensional Gaussian on $\bar{x}$ with mean μ and covariance Σ. Give mathematical expressions for μ and Σ (ie for their elements).

(b) Consider the following regression model

$$y = \mathbf{w}^T \mathbf{b}(x) + n, \tag{7}$$

where the observation noise n is Gaussian with mean zero and variance σ^2, which leads to the following likelihood

$$p(y|x, \mathbf{w}, \sigma^2) = G(y; \mathbf{w}^T \mathbf{b}(x), \sigma^2) \tag{8}$$

$$p(y_{1:N}|x_{1:N}, \mathbf{w}, \sigma^2) = \prod_{i=1}^{N} p(y_i|x_i, \mathbf{w}, \sigma^2). \tag{9}$$

We define a Gaussian prior on the parameters, with mean $\mathbf{v}$, covariance $\mathbf{K}$ ($\mathbf{v}$ and $\mathbf{K}$ are *hyperparameters of the model*):

$$p(\mathbf{w}) = G(\mathbf{w}; \mathbf{v}, \mathbf{K}). \tag{10}$$

The hyperparameters $\mathbf{v}, \mathbf{K}, \sigma$ are assumed to have a uniform prior, i.e.,

$$p(\mathbf{v}, \mathbf{K}, \sigma) = c, \tag{11}$$

where c is a constant (we can ignore the finite bounds on their domains here).

It can be shown that the posterior over the weights is given by

$$p(\mathbf{w} \,|\, y_{1:N}, x_{1:N}, \mathbf{v}, \mathbf{K}, \sigma^2) = \frac{\prod_i G(y_i; \mathbf{w}^T \mathbf{b}(x_i), \sigma^2)\; G(\mathbf{w}; \mathbf{v}, \mathbf{K})}{Z}\,, \tag{12}$$

where Z is a constant. Moreover, the posterior is still Gaussian

$$p(\mathbf{w} \,|\, y_{1:N}, x_{1:N}, \mathbf{v}, \mathbf{K}, \sigma^2) = G(\mathbf{w}; \mu, \Sigma)\,. \tag{13}$$

Your task is to derive the mean μ and covariance Σ of the new posterior. **Hint:** (1) A Gaussian can be written $e^{-(\bar{x}-\mu)\Sigma^{-1}(\bar{x}-\mu)/2}/Z$. Work out the terms of the exponent, completing the square to determine μ and Σ. (2) See the Bayesian Methods chapter of the course lecture notes for a similar problem.

Programming Part (8%): due Friday, November 11th, beginning of class

Spam Classification

We will provide a data set of features from email messages (available from the course web site), each of which corresponds to either a spam email or a valid ("ham") email. The input is a list of discrete features. The output is either spam or ham.

The Data: This data set comes from a collection of 5000 personal email messages, 1000 which are used for the training set, and 4000 for the test set. Each spam message was reduced to 185 binary $\{0, 1\}$ features. The text strings associated with these 185 features are included in the `feature_names` variables. Each message is thus represented by a vector of 185 binary values, i.e., a row in the `data_train` and `data_train` vectors. Your goal is to learn two classifiers. Each takes a 185-vector and returns a class label. The `labels_test` and `labels_train` data sets are binary features indicating which of the emails are spam and which are ham. We'll leave it to you to figure out whether 0 or 1 indicates `spam`. (The data are courtesy of Sam Roweis.)

Your Task: Learn classifiers for these datasets using both Naïve Bayes and Logistic Regression. The starter code includes an implementation of Logistic Regression for you to use, but you will need to implement discrete Naïve Bayes yourself. You should implement the regularized form of learning for Naïve Bayes, see course notes Section 8.7.

For each dataset, test with a variety of regularization parameters (e.g., on the weights of the logistic regression, or on the binomial probabilities as explained in the notes) and plot test-set performance as a function of these parameters. Use these plots to try to select the best setting for these regularizatino parameters. Then, for each of the two methods using the best regularization parameter settings, show a list of the 10 features most indicative of a message being spam, and the 10 features most indicative of ham, along with their weights in the classifier.

MATLAB hints: You can use the `find` command to separate the training sets, e.g.,
`data_train(find(labels_train==0),:)`
will give you all the data from class 0. You can use the `sort` command to find the highest and lowest weights, and then get the corresponding indices from the list of labels.

Submission

Hand in all written work and a print out of your Matlab code. For the electronic version of your solution, tar up the files into one file called A3.tar, and submit it according to instructions on the course web site (again, by the due date).