CS464 Machine Learning, Spring 2017: Homework 1

Due: March 15 17:00 pm

Instructions

- Submit a hard copy of your homework of all questions. Add the print out of your code at the end of the your submission and upload the code online. You may hand in the hard copy in classes to the instructor or may drop it in the box in room EA525 (please stick to this submission routes) and please STAPLE your write up.
- You may code in any programming language you would prefer. In submitting the code on Moodle, please package your code as a gzipped TAR file or a ZIP file with the name `CS464_HW1_Halil_brahim_Kuru`, where you substitute in your first and last names into the filename in place of "Halil_brahim_Kuru". The code you submit should be in a format easy to run, main script to call other functions, please seet the intructions for code submissions in the Moodle main page. You must also provide us with a README file that tells us how we can execute/call your code.
- If you are submitting the homework late (see the late submission policy in syllabus), prepare a **soft copy** for all the parts of the homework, submit everything online on Moodle. Moodle will allow late submissions until after 4 days of submission, but we will grade your homework based on the time stamp and your remaining late days.
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.

1 Slot Machine [10 pts]

You and your friend are playing the slot machine in a casino. Having played on two separate machines for a while, you decide to switch machines to measure for differences in luck. The wins/losses of you and your friend for each machine are tabulated below.

Machine 1	Wins	Losses	Machine 2	Wins	Losses
You	40	60	You	212	828
Friend	25	75	Friend	18	72

Assuming that the outcome of playing the slot machine is independent of its history and that of the other machine, answer the following questions:

Question 1.1 [2 points] If the outcome of a game is a success, what's the probability that it was played using machine 1?

Question 1.2 [5 points] What is the winning probability of you and your friend for each of the machines? Compare your winning probability with your friend's on different machines, who is more likely to win on each machine?

Question 1.3 [3 points] Suppose you did not keep track of the wins/losses for each machine, but only of the total number of wins/loses for the two machines. In this case, estimate the overall winning probability of you and your friend in the casino (assume that there are only two slot machines in the casino). Who is more likely to win?

2 Conditional Independence [10 pts]

Let's say we have two dice: one is blue and the other one is red. You roll the dices and try to predict the outcome. Let's say the outcome of the blue die is b, and the outcome of the red is r. You can understand that the outcome of the dices are independent of each other.
Let's say we have an oracle who helps you to predict the probability of any outcome.

Question 2.1 **[4 pts]** Oracle gives you the following information about the outcomes:

C = 'b is not equal to 6, and r is not equal to 1 '

What is the probability of b = 1 and r = 3, which is `P(b=1,r=3 | C)`?

Question 2.2 **[3 pts]** Independent from the previous information, oracle gives you another information in this time:

D = 'sum of the outcomes (b+r) is an even number '

What is the probability of b = 3 and r = 5, which is `P(b=3,c=5 | D)` ?

Question 2.3 **[3 pts]** What is the difference between the information C and D above? Explain your reasoning in terms of conditional probability.

3 Packets [10 pts]

The Poisson distribution is a useful discrete distribution which can be used to model the number of occurrences of something per unit time. For example, in networking, the number of packets to arrive in a given time window is often assumed to follow a Poisson distribution. If X is Poisson distributed, i.e. $X \sim Poisson(\lambda)$, its probability mass function takes the following form:

$$\mathbf{P}\left(X = x \mid \lambda\right) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Assume now we have n identically and independently drawn data points from $Poisson(\lambda) : \mathcal{D} = \{x_1, \ldots, x_n\}$. Assume that prior distribution for λ is $N(0, \beta^2)$, derive an expression for maximum a posterior (MAP) estimate of λ.

4 Building a Spam Classifier with Naive Bayes [50 pts]

Your job is to build a spam classifier that can accurately predict whether an email is spam or not. The questions summarizes the model, therefore, please read all questions before starting coding.

Dataset

Your dataset is a preprocessed and modified subset of the Ling-Spam Dataset [1]. It is based on 960 real email messages from a linguistics mailing list. Emails have been preprocessed in the following ways:

- **Stop word removal:** Words like "and","the", and "of", are very common in all English sentences and are therefore not very predictive in deciding spam/nonspam status. These words have been removed from the emails.

- **Lemmatization:** Words that have the same meaning but different endings have been adjusted so that they all have the same form. For example, "include", "includes," and "included," would all be represented as "include." All words in the email body have also been converted to lower case.
- **Removal of non-words:** Numbers and punctuation have both been removed. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character

The data has been already split into two subsets: a 700-email subset for training and a 260-email subset for testing (consider this as your validation set and imagine there is another test set which is not given to you). The features have been generated for you. You will use the following files:

- `question4-train-features.txt`
- `question4-train-labels.txt`
- `question4-test-features.txt`
- `question4-test-labels.txt`

The files that ends with `features.txt` contains the features and the files ending with `labels.txt` contains the ground truth labels.

In the feature files each row contains the feature vector for an email. The j-th term in a row i is the number of occurrences of the j-th vocabulary word in the i-th email. The size of the vocabulary is 2500. The label files include the ground truth label for the corresponding email, the order of the emails (rows) are the same as the features file. That is the i-th row in the files corresponds to the same email document. The labels are indicated by 1 or 0, 1 stands for a spam email and 0 stands for the nonspam email.

Bag-of-Words Representation and Multinomial Naive Bayes Model

Recall the bag-of-words document representation makes the assumption that the probability that a word appears in email is conditionally independent of the word position given the class of the email. If we have a particular email document D_i with n_i words in it, we can compute the probability that D_i comes from the class y_k as:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \mathbf{P}\left(X_1 = x_1, X_2 = x_2, .., X_{n_i} = x_{n_i} \,|\, Y = y_k\right) = \prod_{j=1}^{n_i} \mathbf{P}\left(X_j = x_j \,|\, Y = y_k\right) \tag{4.1}$$

In Eq. (4.1), X_j represents the j^{th} position in email D_i and x_j represents the actual word that appears in the j^{th} position in the email, whereas n_i represents the number of positions in the email. As a concrete example, we might have the first email document (D_1) which contains 200 words ($n_1 = 200$). The document might be of spam email ($y_k = 1$) and the 15^{th} position in the email might have the word "office" ($x_j =$ "office").

In the above formulation, the feature vector $\vec{X}$ has a length that depends on the number of words in the email n_i. That means that the feature vector for each email will be of different sizes. Also, the above formal definition of a feature vector $\vec{x}$ for a email says that $x_j = k$ if the j-th word in this email is the k-th word in the dictionary. This does not exactly match our feature files, where the j-th term in a row i is the number of occurrences of the j-th dictionary word in that email i. As shown in the lecture slides, we can slightly change the representation, which makes it easier to implement:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{4.2}$$

,where V is the size of the vocabulary, X_j represents the appearing of the j-th vocabulary word and $t_{w_j,i}$ denotes how many times word w_j appears in email D_i. As a concrete example, we might have a vocabulary of size of 1309, $V = 1309$. The first email (D_1) might be spam ($y_k = 1$) and the 80-th word in the vocabulary, w_{80}, is "click" and $t_{w_{80},1} = 2$, which says the word "click" appears 2 times in email D_1. Contemplate on why these two models (Eq. (4.1) and Eq. (4.2)) are equivalent.

In the classification problem, we are interested in the probability distribution over the email classes (in this case Spam and Nonspam) given a particular email D_i. We can use Bayes Rule to write:

$$\mathbf{P}\left(Y = y_k | D_i\right) = \frac{\mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}}}{\sum_k \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}}} \tag{4.3}$$

Note that, for the purposes of classification, we can actually ignore the denominator here and write:

$$\mathbf{P}\left(Y = y_k | D_i\right) \propto \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}} \tag{4.4}$$

$$\hat{y}_i = \arg\max_{y_k} \mathbf{P}\left(Y = y_k \,|\, D_i\right) = \arg\max_{y_k} \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{4.5}$$

Question 4.1 [2 points] Why it is that we can ignore the denominator?

Probabilities are floating point numbers between 0 and 1, so when you are programming it is usually not a good idea to use actual probability values as this might cause numerical underflow issues. As the logarithm is a strictly monotonic function on [0,1] and all of the inputs are probabilities that must lie in [0,1], it does not have an affect on which of the classes achieves a maximum. Taking the logarithm gives us:

$$\hat{y}_i = \arg\max_{y} \left(\log \mathbf{P}\left(Y = y_k\right) + \sum_{j=1}^{V} t_{w_j,i} * \log \mathbf{P}\left(X_j \,|\, Y = y_k\right) \right) \tag{4.6}$$

, where $\hat{y}_i$ is the predicted label for the i-th example.

Question 4.2 [3 points] If the the ratio of the classes in a dataset is close to each other, it is a called "balanced" class distribution if not it is skewed. What is the percentage of spam emails in the `train.labels.txt`. Is the training set balanced or skewed towards a one of the classes?

The parameters to learn and their MLE estimators are as follows:

$$\theta_{j \,|\, y=0} \equiv \frac{T_{j,y=0}}{\sum_{j=1}^{V} T_{j,y=0}}$$

$$\theta_{j \,|\, y=1} \equiv \frac{T_{j,y=1}}{\sum_{j=1}^{V} T_{j,y=1}}$$

$$\pi_{y=1} \equiv \mathbf{P}\left(Y = 1\right) = \frac{N_1}{N}$$

- $T_{j,0}$ is the number of occurrences of the word j in nonspam emails in the training set including the multiple occurrences of the word in a single email.
- $T_{j,1}$ is the number of occurrences of the word j in spam emails in the training set including the multiple occurrences of the word in a single email.
- N_1 is the number of spam emails in the training set.
- N is the total number of emails in the training set.
- $\pi_{y=1}$ estimates the probability that any particular email will be a spam email.
- $\theta_{j \,|\, y=0}$ estimates the probability that a particular word in a nonspam email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y = 0\right)$
- $\theta_{j \,|\, y=1}$ estimates the probability that a particular word in a spam email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y = 1\right)$

Question 4.3 (Coding) [20 points] Train a Naive Bayes classifier using all of the data in the training set (`train-features.txt` and `train-labels.txt`). Test your classifier on the test data (`test-features.txt` and `test-labels.txt`, and report the testing accuracy as well as how many wrong predictions were made. In estimating the model parameters use the above MLE estimator. If it arises in your code, define $0 * \log 0 = 0$

(note that $a * \log 0$ is as it is, that is -inf). In case of ties, you should predict "non-spam". In the written part of your report what your test set accuracy is? What did your classifier end up predicting? Why is using the MLE estimate is a bad idea in this situation?

Question 4.4 (Coding) [5 points] Extend your classifier so that it can compute an MAP estimate of θ parameters using a fair Dirichlet prior. This corresponds to additive smoothing. The prior is fair in the sense that it "hallucinates" that each word appears additionally α times in the train set.

$$\theta_{j \mid y=0} \equiv \frac{T_{j,y=0}+\alpha}{\sum_{j=1}^{V} T_{j,y=0+\alpha*V}}$$
$$\theta_{j \mid y=1} \equiv \frac{T_{j,y=1}+\alpha}{\sum_{j=1}^{V} T_{j,y=1+\alpha*V}}$$
$$\pi_{y=1} \equiv \mathbf{P}\left(Y=1\right) = \frac{N_1}{N}$$

For this question set $\alpha = 1$. Train your classifier using all of the training set and have it classify all of the test set and report test-set classification accuracy.

Question 4.5 (Coding) [10 points] Calculate Mutual information and rank features. Write indices and mutual information scores of top 10 best features from highest mutual information to the lowest. (Hint: You may want to refer mutual information estimation model provided in Stanford book chapter provided in reference 3 below or provided in lecture slides.)

Question 4.6 (Coding) [10 points] Remove features one-by-one from the least informative one. Keep removing until all features are removed. Plot test-set accuracy as a function of removed number of features.

5 Gausian Naive Bayes Classifier [20 pts]

In this question, you will implement the a Gausian Naive Bayes classifier when the features. The data is consists of continious expression values of 5 different genes measured for each cancer patient. Class labels are three different classes, which are kidney, breast and colon cancers. To ease the notation, we denoted the class labels with discrete numbers as follows:

$$1 = \text{Kidney Cancer}, \; 2 = \text{Breast Cancer}, \; 3 = \text{Colon Cancer}$$

- `question5-train.txt`
- `question5-test.txt`

The first two columns of the data consists of the features, and the third column corresponds to class label.

Fitting Gaussian Distribution to Data

In this question, you will fit Gaussian distributions to each class conditional probabilities. The class conditional Gausian distributions for each feature i are as follows:

$$p(X_i = x | Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-(x-\mu_{ik})^2/2\sigma_{ik}^2}$$

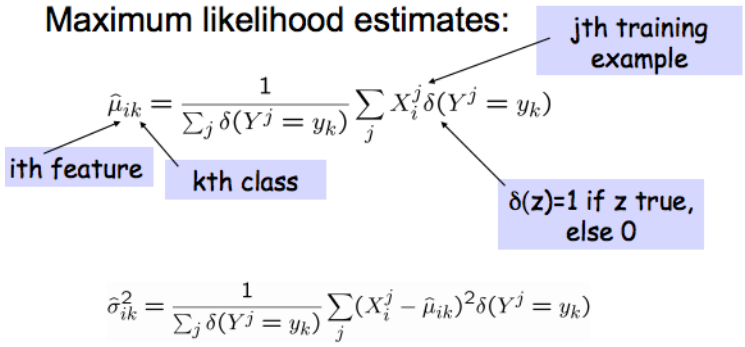

Figure 1: Maximum Likelihood Estimates of Parameter, Source:[4]

Maximum Likelihood Estimation of Parameters

To use the Gaussian distribution for class conditional probabilities, you need to find the parameters of the Gaussian. To ease your work in this question, MLE estimation of μ_{ik} and σ_{ik} are provided below:
Question 5.1 **[10 pts]** Train your Gaussian Naive Bayes classifier, and estimate the parameters using MLE and report them.

Question 5.2 **[10 pts]** In both train and test data, report the confusion matrices for the three different cancer types.

References

1. Liang Spam dataset. http://csmining.org/index.php/ling-spam-datasets.html
2. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes" by Andrew Ng and Michael I. Jordan.
3. Manning, C. D., Raghavan, P., and Schutze, H. (2008). Introduction to information retrieval. New York: Cambridge University Press.
http://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html
4. CMU Lecture Notes.
http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf