CS464 Introduction to Machine Learning
Fall 2019
Homework 1

Due: November 10, 11:59 PM

Instructions

- Submit a soft copy of your homework of all questions to Moodle. Add your code at the end of the your homework file and upload it to the related assignment section on Moodle. Submitting a hard copy or scanned files is NOT allowed. You have to prepare your homework digitally(using Word, Excel, Latex etc.).
- This is an individual assignment for each student. That is, you are NOT allowed to share your work with your classmates.
- For this homework, you may code in any programming language you would prefer. In submitting the homework file, please package your file as a gzipped TAR file or a ZIP file with the name `CS464_HW1_Section#_Firstname_Lastname`.

  As an example, if your name is Sheldon Cooper and you are from Section 1 for instance, then you should submit a file with name `CS464_HW1_1_sheldon_cooper`. Please do not use Turkish letters in your package name. The code you submit should be in a format easy to run and must include a main script serving as an entry point. You must also provide us with a README file that tells us how we can execute/call your program.
- You are NOT allowed to use any machine learning packages, libraries or toolboxes for this assignment (such as scikit-learn, tensorflow, keras, theano, MATLAB Statistics and Machine Learning Toolbox functions, e1071, nnet, kernlab etc.).
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.

1 The Monty Hall Problem [10 pts]

You are racing in a TV game show and the host asks you to choose one of 3 doors: Behind one is a car, behind the others there are goats. After you pick a door, the host reveals one of the other two doors, behind which there is a goat. As a second chance, you can now switch your selected door if you want.

As an example, assume that you have selected Door 1 initially. Now the host reveals that Door 3 has a goat behind. Now, you can either stay with Door 1 or switch to Door 2 instead.

Question 1.1 [4 pts] Explain how switching your door affects your chances to win the car.

Question 1.2 [6 pts] Now, assume the TV show has N doors instead of 3 as the other rules are kept the same i.e. only one door has the car and the others have goats. The host opens a door with a goat behind after you choose one of N doors and then you have the chance to switch your door with one of remaining

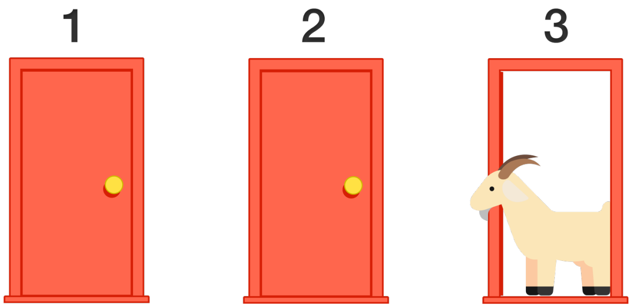

Figure 1: The Monty Hall Problem Example

N-2 doors or you can choose not to switch and go with your original door. In this case, prove or disprove explicitly if switching the original door always increases your winning chance.

2 MLE and MAP [15 pts]

Suppose that 20 identical coins are flipped independently in an experiment where 7 out of 20 coin flips are observed to be heads. Probability of heads is θ.

Question 2.1 [**5 pts**] Find MLE estimate of θ.

Question 2.2 [**7 pts**] Assume that before 20 coin flip experiment we flip $N_1 + N_2$ coins in order to construct prior belief on parameter θ. The outcome of $N_1 + N_2$ flips are given as follows;

- N_1 heads are observed
- N_2 tails are observed

Using Beta(α, β) distribution[1] as prior on θ, find MAP estimate of θ as a function of N_1 and N_2 explicitly.

Question 2.3 [**3 pts**] Use Beta(α, β) distribution as prior on θ to show the relation between MLE and MAP estimates of θ if no experiment is conducted a-priori. Show your steps explicitly.

3 DrugBank Drug Target Identifiers Data Set [25 pts]

Biological sciences are becoming data-rich and information-intensive. Machine learning applications have become very useful and popular tools for resolving important questions in biology by enabling to analyze vast amount of biological information. In this question, you will be introduced to one of the biological data sets, DrugBank, and gain intuition of feasibility of kNN algorithm using this data set.

Data set

The data set contains their 4765 unique proteins and 136 unique drugs that target those proteins. [2] Your job is to predict whether a protein is pharmacologically active, i.e. directly related to the mechanism of action for at least one of the associated drugs, or not by using the target information of drugs.

The data has been split into training and validation subsets with the ratio of 3812/953 i.e. 20% of the proteins are taken for the validation set. You will use the following files:

- `question-3-train-features.csv`

- question-3-train-labels.csv
- question-3-valid-features.csv
- question-3-valid-labels.csv
- question-3-train-protein-index.csv
- question-3-valid-protein-index.csv
- question-3-drug-index.csv

The files that end with `features.csv` contain the features and the files ending with `labels.csv` contain the ground truth labels.

In the feature files, each row contains the feature vector for a protein. The j-th term in a row i, is the target information of the j-th drug for the i-th protein. The label files include the ground truth label for the corresponding proteins where the order of the proteins (rows) are the same as the features file. That is the i-th row in the files corresponds to the same protein. Any protein is labeled as either `active` (1) or `inactive` (0).

The files ending with `index.csv` are the index files in which the first element in j-th is the protein or drug that j-th row or column represents in the feature files.

Question 3.1 [5 pts] Take a quick look to your feature files and decide on a distance metric for your kNN classifier. Briefly explain your motivation for choosing your distance metric. What would happen if you used another metric to calculate distances?

Question 3.2 [12 pts] Train your kNN classifier $\forall\ k \in$ [1, 3, 5, 10, 20, 50, 100, 200]. You do not have to use all values if you observe a trend in the outputs. Report accuracy and precision on validation set for each of your training sessions. You may use plots or tables. Comment on your results. What have you expected and are the result parallel or contradictory to your expectations?

In addition, make note of training time and validation time of your classifier for the next question.

Question 3.3 [8 pts] Comment on the run times that you have noted in the previous question. You may use plots or tables to support your comments. Find the complexity of the brute-force kNN algorithm and report it using big O notation. In which cases using kNN would not be a feasible solution?

4 Sentiment Analysis on Emails [50 pts]

As a computer scientist working for an online science magazine, your job is to analyze online data to classify mails from your subscribers according to their topics.

Data set

Your dataset is a preprocessed and modified version of 20 News Group Data Set [3]. It is based on 4000 real emails about 4 different topics in science. 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. These words have been removed from the emails.
- **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
- **Removal of infrequent words:** Words that occur only once in all data set are removed from emails in order to reduce the size of the data.

The data has been already split into two subsets: a 3200-email subset for training and a 800-email subset for testing (consider this as your validation set and imagine there is another test set which is not given to you). Features have been generated for you. You will use the following files:

- `question-4-train-features.csv`
- `question-4-train-labels.csv`
- `question-4-test-features.csv`
- `question-4-test-labels.csv`
- `question-4-vocab.txt`

The files that end with `features.csv` contain the features and the files ending with `labels.csv` contain 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 occurrence information of the j-th vocabulary word in the i-th email. The size of the vocabulary is 37358. The label files include the ground truth label for the corresponding email (label 0 is medicine, label 1 is space, label 2 is cryptology and label 3 is electronics), 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. Each email is labeled as either `cryptology`, `space`, `medicine` or `electronics`.

The file ending with `vocab.txt` is the vocabulary file in which the j-th word (feature) in the file corresponds to the j-th feature in both train and test sets.

Bag-of-Words Representation and Multinomial Naive Bayes Model

Notice that the bag-of-words document representation assumes that the probability of a word appearing in an 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 (D_1) which contains 200 words ($n_1 = 200$). The document might be of space email ($y_k = 1$) and the 15th position in the email might have the word "saturn" ($x_j =$ "saturn"). 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 an 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 about space ($y_k = 1$) and the 80-th word in the vocabulary, w_{80}, is "planet" and $t_{w_{80},1} = 2$, which says the word "planet" appears 2 times in the 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 medical, space, cryptology and electronics emails) 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] Explain why the denominator can be ignored in Eq. (4.3).

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.

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

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

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

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

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

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

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

- $\theta_{j\,|\,y=electronics}$ estimates the probability that a particular word in an electronics email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y = electronics\right)$
- $\theta_{j\,|\,y=cryptology}$ estimates the probability that a particular word in a cryptology email will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y = cryptology\right)$

Question 4.2 [3 points] How many parameters do we need to estimate for this model?

Question 4.3 (Coding) [25 points] Train a Naive Bayes classifier using all of the data in the training set (`question-4-train-features.csv` and `question-4-train-labels.csv`). Test your classifier on the test data (`question-4-test-features.txt` and `question-4-test-labels.txt`), and report the **testing accuracy** and **confusion matrix** 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 "space". What did your classifier end up predicting? Why is using the MLE estimate 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 assumes that each word appears additionally α times in the train set.

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

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

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

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

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

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 accuracy** and **confusion matrix**. Comment on the results.

Question 4.5 [2 points] What would happen as we increase α in fair Dirichlet prior ? What would you end up predicting when $\alpha = \infty$ is used ? Explain clearly.

Question 4.6 (Coding) [5 points] Using `question-4-vocab.txt` file, find the most commonly used 20 words in each email class in the training set and make comments on them. Do you think the most common words are as expected? Does the model that you have constructed is interpretable? Explain clearly.

Question 4.7 (Coding) [6 points] For each class of emails, find 1 instance in test set that has the highest probability and 1 instance that has the lowest probability. You have to report 8 instances in total (you can simply report the indices of these instances in test set provided to you). Briefly discuss why these instances are easy/hard to predict.

Question 4.8 [2 points] Describe the differences between Bernoulli Naive Bayes model and Multinomial Naive Bayes model. To use Bernoulli model for email class prediction, what would you change in your previous code? Explain clearly.

References

1. Beta distribution. https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading14a.pdf
2. DrugBank - Drug Target Identifiers Data Set https://www.drugbank.ca/releases/latest#protein-identifiers
3. 20 News Group Data Set http://qwone.com/~jason/20Newsgroups/
4. ”On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes” by Andrew Ng and Michael I. Jordan.
5. 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
6. CMU Lecture Notes.
http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf