IITM-CS6730 : Probabilistic Reasoning in AI — Given on: Feb 05, 6pm

Programming Assignment #1 — Due on : Feb 15, 10pm

- The goal of this assignment is to gain understanding in Maximum Likelihood Estimation, Bayesian Estimation and sampling from a distribution.
- This is an individual assignment. Collaborations and discussions with others regarding the problems and solutions are strictly prohibited.
- You have to turn in your report and code in the prescribed format in Moodle. Report should be typesetted in Latex only.
- You can use either of the following languages: C, C++, Java, Python, Matlab, Octave, and R.

1. Design and implement a Bayesian Spam Filter that classifies email messages as either spam (unwanted) or ham (useful), i.e $y_i \in \{spam, ham\}$ for the following four scenarios. (Credits: Deepak V)
   (a) Maximum Likelihood Estimation assuming likelihood $\mathcal{L} \sim \text{Multinomial}(n_1, n_2, ..., n_k, N)$ where k is the size of the vocabulary, n_w is the number of times word w appears in the document d and $N = \sum_i n_i$.
   (b) Maximum Likelihood Estimation assuming likelihood $\mathcal{L} \sim \text{Bernoulli}(i, p)$, where p is the parameter of the Bernoulli distribution and $i \in \{0, 1\}$. In our case, we would have k Bernoulli distributions.
   (c) Bayesian Parameter Estimation assuming that prior $p \sim \text{Dir}(\overrightarrow{\alpha})$, where $\overrightarrow{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_k)$, the elements of the vector are the parameters of the Dirichlet distribution.
   (d) Bayesian Parameter Estimation, assuming that prior $p \sim \text{Beta}(\alpha, \beta)$, where α and β are the parameters of the Beta distribution.

Naive Bayesian Classifier:

Naive Bayesian Classifier takes a collection of words as input and predicts the category of the given text. Naive Bayes algorithm for text classfication involves two stages : training and classfication. In the training stage, various probabilities are estimated based on the training examples. In the classfication stage, the estimated probabilities are used to evaluate the likelihood of each class for the given document. The document is then assigned a label with the highest likelihood score.

Let $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..\mathcal{C}_m\}$ be the set of labels and $\mathcal{V} = \{w_1, w_2, ..w_n\}$ be all the words in the vocabulary. You would need to estimate the following propbabilities in the training stage.

- Class priors : $p(C_i)$ for $i = 1, 2, .., m$. Note that $\sum_{i=1}^{k} p(C_i) = 1$.
- Within class word probabilities : $p(w_j|C_i)$ for $i = 1, 2, .., m$ and $j = 1, 2, .., n$. Note that $\sum_{i=1}^{n} p(w_j|C_i) = 1$ for $i = 1, ..., k$.

In the classification stage, given a document d, the estimated probabilities can be used to compute $\widehat{\mathcal{C}}$. Since the estimated probabilities might be too low, we would have to deal with underflow. This can be avoided by considering log-likelihood.

Description of the dataset:

The data set contains a collection of spam and legitimate emails. Each token (word, number, punctuations, etc.) is replaced by a unique number througout the dataset. In order to get the emails to an usable by Naive Bayes, some preprocessing is required. Each document should be represented by a term frequency vector of size k, where the ith element is the frequency of the ith term in the vocabulary (set of all distinct words in any text document from the training set). Do not consider the token "Subject:" as part of the vocabulary. Files whose names have the form **spmsg*.txt** are spam messages. Files whose names have the form ***legit*.txt** are legitimate examples.

Submission

- You are required to implement Naive Bayesian classfier for 4 different scenarios as described previously. In the last two cases you are expected to try out different parameters for the prior distribution. Report the results after performing 5-fold cross validation (80-20 split). You have 10 folders in the dataset. Use 1-8 for training, 9-10 for testing in one fold. In the next fold, use 3-10 for training, 1-2 for tesing and so on.
- Comment on the impact of choice of parameters on the performance. Also comment on the performance of the classifier under different scenarios. Plot a PR-curve (refer Page 145-146 Manning, Raghavan and Schtze) for each scenario and for different parameter setting in the last 2 scenarios.
- Compare your implementation with Naive Bayes Classifier in WEKA. For using WEKA the input should be in ARFF format. Refer Witten and Frank (Chapter 2.4, chapter 10.1, chapter10.2). Also report the performance of Naive Bayes classifier in WEKA.
- Refer to chapter 13 of Manning, Raghavan and Schtze for further reference on implementation of Naive Bayes for text classification.
- Your code should take $trainMatrix_{pXk}$, $train-label_{pX1}$, $testMatrix_{rXk}$ and $test-label_{rX1}$ in the first two scenarios. p and r are number of documents in training and test set respectively and k is the size of the vocabulary. In the last 2 scenarios it should also take the parameters for the prior distribution as input. The code

should output precision, recall, f1-measure for *spam* class and plot PR-Curve for the best model obtained in terms of performance.

2. In the class, I asked you to think about how to sample from an arbitrary distribution. There are multiple ways to do so. In this assignment, we would explore two such ways: Inversive Method and Rejective Method. Refer the following links for the details of these methods.

   - http://www.amstat.org/sections/srms/proceedings/y2008/Files/300875.pdf
   - http://web.mit.edu/urban_or_book/www/book/chapter7/7.1.3.html

   You are expected to write two functions with the following prototype.

   sampled-value $inverse\_method$(inverse-cdf,min,max)

   sampled-value $reject\_method$(fx,min,max)

   I am deliberately not mentioning about the boundary cases that you should consider. Try to explore all possible boundary cases and you will get more credits if your function handles more boundary cases.

Resources

- Data Mining : Practical Machine Learning Tools and Techniques, Ian H. Witten and Ebie Frank, 2nd edition.
- Introduction to Information Retrieval, Christopher D. Manning, Prabhakaran Raghavan, and Hinrich Schtze, 2008.

Submission Instructions

Submit a single tarball/zip file containing the following files in the specified directory structure. Use the following naming convention: 'rollno.tar.gz' with all capital letters. Ex: CS12S043.tar.gz

rollno
    Report
        report.pdf
    Code
        all your code files.
    Data
        all the data that are needed for your code to run.