DCP1206 Fall 2019: Probability **(Due: 2020/01/03 in class)**

Homework 6, Part II: Parameter Estimation

Problem 5 (Maximum Likelihood Estimation) (10+15=25 points)

Let $X_1, X_2, \cdots, X_n$ be a sequence of i.i.d. *log-normal* random variables with CDF given by $P(X_i \leq x) = \Phi(\frac{(\ln x)-\mu}{\sigma})$, where $\Phi(\cdot)$ is the CDF of a standard normal random variable and μ, σ are the parameters of a log-normal distribution.

(a) Show that the PDF of X_i is given by

$$f(x) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} \cdot \exp(-\frac{((\ln x)-\mu)^2}{2\sigma^2}), & \text{if } x > 0 \\ 0, & \text{else} \end{cases}$$

(b) Suppose μ and σ^2 are unknown and we observe the outcomes of these n random variables as $X_i = x_i$, for every $i = 1, \cdots, n$. Find the maximum likelihood estimators of μ and σ^2. (Hint: Slides of Lecture 26)

Problem 6 (Naive Bayes Classifier for Spam Filtering) (15+10=25 points)

Naive Bayes classification is a simple and classic tool for machine learning problems, especially for spam filtering and text classification. In this problem, we will implement a naive Bayes classifier in python with the help of the *scikit-learn* package. The resulting classifier will be trained and tested on a SMS spam collection dataset provided on E3 (Source: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/).

Here is a brief summary of naive Bayes classification in the context of spam filtering:

- Our goal is to learn a classifier that determines whether a piece of SMS is spam or not (i.e. there are only two types of labels, "spam" or "ham") given its raw text.
- Specifically, we would leverage maximum a posteriori (MAP) estimation to determine the label of a SMS: Given a piece of SMS with n words $\{x_1, x_2, \cdots, x_n\}$, we would like to determine whether it is a spam or not. In naive Bayes classification, we typically assume *conditional independence* for the likelihood, i.e.

  $$p(x_1, x_2, \cdots, x_n|\text{label}) = \prod_{i=1}^{n} p(x_i|\text{label}).$$

  Then, we could apply MAP by calculating the following posterior distribution:

  $$p(\text{spam}|x_1, x_2, \cdots, x_n) \propto p(\text{spam}) \cdot p(x_1, x_2, \cdots, x_n|\text{spam})$$
  $$p(\text{ham}|x_1, x_2, \cdots, x_n) \propto p(\text{ham}) \cdot p(x_1, x_2, \cdots, x_n|\text{ham})$$

- To find $p(x_i|\text{label})$, we shall leverage a training dataset to find the empirical frequency of each word for each type of SMS.
- The prior distribution (i.e. $p(\text{spam})$ and $p(\text{ham})$) is to be configured by the designer.

Based on the above summary, in this implementation there are mainly 3 mini-tasks (normally you need no more than 60 lines of code in total):

- **Read and split the dataset**: The SMS spam dataset is provided in "spam.csv", which contains 5572 English text messages. Each message has two fields, namely the label (either "ham" or "spam") and the corresponding raw text. Moroever, we need to divide the dataset into a training set and a testing set (Note: This part has already been done in "naive_bayes.ipynb").
- **Train the classifier**: First, find the empirical frequency of each word for both spam and ham SMS using the training dataset (possibly with the help of some existing parsing/counting functions). Next, implement the MAP estimation (Hint: You may use MultinomialNB in the scikit-learn package)

- **Test the classifier**: Predict the labels of the messages in the testing dataset and find the accuracy of your classifier.

(a) Please finish the remaining parts of "naive_bayes.ipynb". What is the accuracy of your classifier with a 70%/30% partition of the dataset and a uniform prior?

(b) What if we use different prior distributions (please try at least 2 priors other than the uniform prior)? Also, if we change the partition of the dataset, will there be any significant change in the accuracy?

Please briefly summarize your observation in a technical report (no more than 1 page) and turn in your code and the report via E3.