CS 267A Project Proposal

Huo, Jianfan
jihuo1116@g.ucla.edu

Du, Yunhao
yudwn010@g.ucla.edu

Wang, Yuchen
ycw509@g.ucla.edu

May 2023

Abstract

Our project aims to create an efficient and accurate spam classification model that will significantly enhance the filtering capabilities of messaging platforms. With the exponential growth of online communication, the number of spam message has become a serious issue, leading to stress and potential safety for users. To address this challenge, we will use the power of modeling data with a probabilistic programming language. Our model will consider specific words within messages as conditional parameters for classification. We considered putting some specific words within the message to our model as the conditional parameter for classification. The dataset will include approximately 87% non-spam and 13% spam messages. We will carefully split the 80% dataset into training and 20% dataset into test to ensure unbiased evaluation. During the training process, we will construct a vocabulary table, which will include words commonly associated with spam, such as "prize" and "claims." These words will be the crucial features for training our model to accurately identify the spam message. For the creation of this glossary, we are still exploring its feasibility, which is the biggest challenge for our project to consider now. After the training is complete, we will check if it achieves a predetermined baseline of accuracy. And we will also use the Problog to refine the model's predictions and further improve its efficiency.

1 Motivation and Introduction

In the dynamic landscape of today's rapidly evolving information, the alarming prevalence of personal data breaches has reached new heights. Protecting our cherished information against such breaches is of paramount importance. However, it is equally important that once our private information is leaked, we must protect ourselves for dealing with the far-reaching effects. Amidst these perils, the nefarious realms of spam and deceitful information cast an indelible impact on our lives. Though the dangers posed by the private information leaks, fraudulent messages have a huge impact on our lives. While sometimes these dangers may seem insignificant, but if we don't pay attention on them, the consequences can quickly escalate to terrible levels, including finances, time and even our lives. Even if we are careful in our lives to avoid becoming victims of these scams, the presence of these massive scams and spam messages can cause us psychological pain. We are keen to reduce the impact of spam on people's lives by implementing a keyword filtering system. During the project, we will develop deeper into the usage of ProbLog, extensively explore the usefulness of the Probabilistic Programming language, and gain a comprehensive understanding of how it benefits data analysis. This groundbreaking solution is designed to protect individuals from the relentless plague of fraudulent messages.

2 Course relevance

In this project, we will cover the topic of Discrete Probabilistic Program Inference we have studied in class. We will use the Naive Bayes Algorithm to calculate the probability that the given message is spam or non-spam given some of the words. The idea is based on the following equations:

$$Pr(Spam|w_1, w_2, ...) = Pr(Spam) * \prod_{i=1}^{n} Pr(w_i|Spam)$$

$$Pr(Nonspam|w_1, w_2, ...) = Pr(Spam) * \prod_{i=1}^{n} Pr(w_i|Nonspam)$$

where wi represents the word in the message sentence.

To test the accuracy of our model probabilities, we will also explore the use of the probabilistic programming language Problog to calculate these probabilities and see if these two methods would produce similar results.

Before that, we will use Model Counting to calculate $Pr(wi|Spam)$ and $Pr(wi|Nonspam)$ insides the formula above. We need to use the following equations:

$$Pr(w_i|Spam) = \frac{N_{wi|Spam}}{N_{Spam}}$$

$$Pr(w_i|Nonspam) = \frac{N_{wi|Nonspam}}{N_{Nonspam}}$$

$N_{wi|Spam}$ represents the number of times the words wi occurs in spam messages
N_{Spam} represents the total number of words in spam messages
$N_{wi|Nonspam}$ represents the number of times the words wi occurs in non-spam messages
$N_{Nonspam}$ represents the total number of words in non-spam messages

Due to the limitation of our training dataset, it is impossible for us to explore all combinations of a group of words. So, we might encounter the zero probability problem when calculating the above probabilities. Therefore, we decide to use Lapace Smoothing to help us address this problem and our updated formulas are:

$$Pr(w_i|Spam) = \frac{N_{wi|Spam} + \alpha}{N_{Spam} + \alpha \cdot N_{Words}} \tag{1}$$

$$Pr(w_i|Nonspam) = \frac{N_{wi|Nonspam} + \alpha}{N_{Nonspam} + \alpha \cdot N_{Words}} \tag{2}$$

where α will be a small integer for avoiding the zero probability issue.
where N_{Words} represents the number of unique words in all the messages of the training dataset

On the basis of calculating these probabilities, we can further explore the relationship between these words and scam information. In other words, we can try to find out which words are dependent on scam messages in order to help us more accurately identify the spam message in our training model.

3 Background

For exploring the dataset and creating the spam filter model, we mainly use a Python environment and a Probabilistic Programming language ProbLog which we don't know yet. Therefore, we will consider downloading the Problog package in our Python environment and studying ProbLog based on the following relevant references which provide the guideline for installation and tutorial:

`https://github.com/ML-KULeuven/problog`
`https://problog.readthedocs.io/en/latest/python.html`

The dataset used was from The UCI Machine Learning Repository. It includes a total of 5574 SMS messages that were collected from free or free for research sources on the Internet. You can download the dataset from the following link:

`https://archive.ics.uci.edu/ml/datasets/sms+spam+collection#`

4 Technical Contribution

The technical contribution of our project lies on the development of probabilistic programming(based on Problog) and the completion of an efficient and accurate spam classification model. The goal of our project is to address the challenges of today's spam nuisance and provide the spam filtering of the message platform. To achieve this, we have considered the following technical aspects:

- Using probabilistic programming languages:
  This is the most basic technical support for our project. In this project, we will use the power of probabilistic programming language to help us analyze the data more quickly and effectively. The main programming language we use for modeling is Python. The advantage of using Python is the huge collection of libraries and frameworks for machine learning and natural language processing tasks. Therefore, we want to implement a combination of Python and probabilistic programming languages in our project to build a Spam filter with higher accuracy.

- Dataset clean and construction:
  As we mentioned in the abstract, we explored a dataset consisting of 87% non-spam and 13% spam. We have now finished cleaning and classifying the data and have obtained 8484 non-duplicate words from all the messages in the training dataset. This allows us to collect various different information about each message, annotate it correctly and avoid the use of duplicate data for the best performance.

- Calculation of probabilities and model building:
  This is the main challenge we are currently experiencing as the project proceeds. We use the Model Counting to calculate the $Pr(wi|Spam)$ and $Pr(wi|Nonspam)$ for our model building by using the formula $Pr(wi|Spam) = \frac{N_{wi|Spam}}{N_{Spam}}$, where N_{spam} represents the number of words in all the spam messages. However, the current model is not as well established as our thought. Since the training set is limited, we might encounter zero probability in the Naïve Bayes. Therefore, We consider adding an α(which is a smoothing parameter) to help us avoid this error during the model building.

- Constructing spam words table:
  During the training process, we can construct a vocabulary table containing words that are commonly associated with spam, such as "prize", "free" and "AD". We can determine whether a word is fraud-related based on its probability of appearing in a spam message and then construct a vocabulary table with associated words of spam that have a high probability of $Pr(Spam|wi)$. This vocabulary table will be used as an important feature set for training our model. The construction of the vocabulary and the identification of associated words of spam will help us to improve the accuracy and reliability of the model.

- Use Problog to test accuracy:
  This is the main challenge we need to face in the rest of the time. We will use the Problog knowledge taught in class and learned outside of class to test our model probabilities and make sure we have achieved the baseline we envisioned. Currently, we have discovered that evidence() and query() methods will help a lot when we try to classify a message based on a certain group of words. We will learn more about Problog techniques as we work through this project.

- Improved model accuracy and more optimization:
  Based on the results obtained from Problog, we will further optimize our model to make it more accurate. If there is enough time, we will consider some other methods to improve the accuracy of our model, such as performing data pre-processing, balance dataset, and updating spam trends.

By integrating these technical components, our project aims to complete a spam classification model to help people reduce the nuisance of spam in their lives. The usage of probabilistic programming languages, dataset cleaning and construction, the construction of vocabulary table, Calculation of probabilities, and the usage of Problog will help us to complete the model, improve its accuracy, and contribute greatly to improving the filtering capability of the information platform.

5 Measures of success

- Baseline:
  Our minimum requirement is to be able to explore the entire dataset at the end of the quarter, calculate all the probabilities we need, and use our model to achieve at least 80% filter accuracy.
- Medium:
  For the higher level, we will adjust our Naive Bayes and model to achieve more than 95% accuracy for spam classification. In addition, we hope that our model not only has high test accuracy in the dataset we trained but also can accurately filter it when we get a brand new message.
- Strech:
  We all know that if you want to spam a message, there must be a reason, such as fraud, false information, or advertising. If we have extra time, we hope that on the existing basis, our model could provide the corresponding reason based on the key words if it predicts a message needed to be spam.

6 Planning and Timeline

Based on the expected output of our project and the remaining time for this quarter, we plan to divide and arrange tasks on a weekly basis.

- Week 7:
  - Learn ProbLog and get familiar with the dataset (everyone)
  - Pre-processing the dataset for future analysis
    - Extract the useful information (messages and their corresponding labels) from the dataset (Yuchen)
    - Split the dataset into the training set and the testing set for building our model and making the predictions (Yunhao)
- Week 8:
  - Train the model
    - Calculate the frequency at which each word appears on two labels after removing the punctuations and capitalizations (Jianfan)
    - For different combinations of words, calculate the probability that they will form a spam message and the probability that they will form a non-spam message based on the Naive Bayes algorithm (Yuchen)
    - For each label, calculate the probability for each word in that label (Yunhao)
- Week 9:
  - Use the probabilities we calculated to predict the label for a new message (Jianfan)
  - Measure the accuracy of our filter and try to improve it (Yuchen)
  - Write Project Check-in (everyone)
- Week 10:
  - Continue improving our filter (Yunhao & Jianfan)
  - Try to classify the spam messages based on the spam reasons (fake news/fraud messages/advertisement) (everyone)
    - Come up with different reasons for labeling a message to be "spam"
    - Group the keywords for each reason
    - Calculate the frequency at which each keyword appears in each reason
    - For each reason, calculate the probability for each keyword in that reason
- Week 11:

- Continue working on classifying the spam messages into different reasons and try to achieve the best accuracy (everyone)
- Final Writeup (everyone)