FINAL PROJECT

STATE UNIVERSITY OF NEW YORK - UNIVERSITY AT ALBANY

PHY/CSI/INF 451/551 - BAYESIAN DATA ANALYSIS

Predicting Keywords and Tags of Questions: A Bayesian Data Analysis Approach

Students: Sibelius Seraphini and Xiaohu Nian

Professor: Kevin H. Knuth

2013

Contents

1 Introduction **2**

2 Methods and Materials **3**

2.1 Preprocessing the Data 3

2.2 The Bayesian Data Analysis Approach 3

2.3 From the Trainset to the Probabilities 4

3 Results **6**

4 Conclusion **8**

1 Introduction

StackExchange is a collection of 110 questions and answer sites on diverse topics from programming to cooking to photography and gaming. (StackExchange, 2013a). Regarding this, Facebook (Facebook, 2013a) has created an online recruiting competition in Kaggle (**?**) that has the aim of "Identify keywords and tags from millions of text questions" (Facebook, 2013b). Facebook provided a Train and a Test dataset with real questions from the Stack Overflow (StackExchange, 2013b), a StackExchange's site that contain topics from programming languages and computer science. The task is to predict the tags (a.k.a. keywords, topics, summaries) given only the question text and its title.

Accordingly, a Bayesian Data Analysis was proposed in this project to predict the tags. With this in mind, the train dataset was used to calculate the prior information, the likelihood values, and the evidence of a given question.

2 Methods and Materials

The Train set contains 4 columns for each question: Id,Title,Body,Tags. Id is the unique identifier for each question. Title is the question's title. Body is the body of the question. Tags is the tags associated with the question. The Test set contains the same columns but without the Tags, which we hava to predict.

Given that our data is text, we employ some text mining techniques (Feldman and Sanger, 2007) to transform the data into the prior information, likelihood, and evidence necessary for the bayesian data analysis.

2.1 Preprocessing the Data

The first step to analyse this data is to perform some preprocessing (e.g., remove punctuation symbols, all characters to lowercase, and so on). The steps of our preprocessing approach is showed below:

1. One record per line;
2. Concatenate title and body of questions;
3. Transform to lowercase;
4. Remove HTML tags and entities;
5. Remove Stop Words;
6. Strip Whitespaces.

The first step aims to remove newlines inside the body questions become only one record per line. After, we join the title and body of questions to have more change to find keywords. Next, we transform all the characters to lowercase. Subsequently, we remove the HTML tags and entities (e.g., <br />, &lt;), and stop words. Stop words are common words as *the*, *is*, *at*, *which*; they should be removed to improve the text categorization (Silva and Ribeiro, 2003). Afterwards, we strip the whitespaces, i.e, remove duplicates spaces. Finally, we reduce our Train set from 6.8 Gigabyte to only 3.9 Gigabyte (%57); and we reduce our Test set from 2.2 Gigabyte to only 1.3 Gigabyte (%59).

Preprocessing the data used the *tr* command (SS64, 2013), python and perl.

2.2 The Bayesian Data Analysis Approach

Our Bayesian Data Analysis Approach uses the Bayes' Theorem in formula 1:

$$P(Tag = t|Keyword = k, I) = \frac{P(Tag = t|I) \times P(Keyword = k|Tag = t, I)}{P(Keyword = k|I)} \quad (1)$$

1. $P(Tag = t|I)$ is the Prior Information about the Tag = t, i.e., what is the probability of a question has a Tag=*t* without any information about this question.
2. $P(Keyword = k|Tag = t, I)$ is the Likelihood Probability, that is, the probability of a question has a Keyword = *k* given that question has a Tag = *t*.
3. $P(Keyword = k|I)$ is the Evidence, in other words, it is the probability of a Keyword = *k* appear in a question with any Tag.
4. $P(Tag = t|Keyword = k, I)$ is the Posterior Probabiity, i.e, the probability of a question has a Tag = *t* given a Keyword = *k*.

The formula 1 can be expanded for any number of keywords, as shows in formula 2:

$$\begin{multline}P(Tag = t|Keywords = \{k_1, k_2, \ldots, k_n\}, I) = \\ \frac{P(Tag = t|I) \times P(Keyword = \{k_1, k_2, \ldots, k_n\}|Tag = t, I)}{P(Keyword = \{k_1, k_2\}, \ldots, k_n}|I) \quad (2)\end{multline}$$

2.3 From the Trainset to the Probabilities

This subsection describe the approach to calculate the Prior Information, the Likelihood, and the Evidence using the dataset provided in the Trainset.

The Prior Information of a given Tag=*t* was defined as the frequency of the Tag in the trainset, i.e., the number of questions that has the Tag=*t* divided by the total number of question in the trainset; the formula 3

$$P(Tag = t|I) = \frac{\text{Number of questions of Tag=t}}{\text{Total number of questions in the trainset}} \quad (3)$$

The Likelihood Probability of a Keyword=*k* given a Tag=*t* is the importance value of the keyword divided by the sum of importance of the other keywords in questions that have Tag=*t*. The importance of a keyword is calculate using the Tf-Idf, a numerical statistic that reflects how important a word is to a document in a collection of documents (Aizawa, 2003). Keywords are the words that best describe a Tag (a.k.a. keywords, topics, summaries). It was selected the 20 most important word to be the keywords of a given tag. The formula 4 illustrate this calculation. Additionally, the formula 5 show how to expand it to a set of keywords.

$$P(Keyword = k|Tag = t, I) = \frac{\text{Importance value of the Keyword } k}{\text{Sum of Importance of the other Keywords of Tag=}t} \quad (4)$$

$$P(Keyword = \{k_1, k_2, \ldots, k_n\}|Tag = t, I) = \prod_{k \,\in\, \{k_1, k_2, \ldots, k_n\}} P(Keyword = k|Tag = t, I) \quad (5)$$

The Evidence is calculate using the Prior Information of all the Tags and the Likelihood of all the combination of keywords and tags. The formula 6 show that how to calculate the Evidence given the Prior Information and the Likelihood. The formula 7 expands to a set of keywords.

$$P(Keyword = k|I) = \sum_{t \,\in\, Tags} P(Keyword = k|Tag = t, I) \times P(Tag = t|I) \quad (6)$$

$$P(Keyword = \{k_1, k_2, \ldots, k_n\}|I) = \sum_{t \,\in\, Tags} \prod_{k \,\in\, \{k_1, k_2, \ldots, k_n\}} P(Keyword = k|Tag = t, I) \times P(Tag = t|I) \quad (7)$$

In summary, this subsection described how to calculate the Prior Information, the Likelihood, and the Evidence using the Trainset. Hence, it is possible to calculate the Posterior Probability using the keywords of a question.

3 Results

The evaluation of the Bayesian Data Analysis approach proposed used a small set of the trainset. This small set contains 20000 questions preprocessed of the tags: *java*, *android*, *php*, and *c++*. 5000 questions of each tag.

The Prior Information calculated for the four tags is shown in Figure 1. The most probable tag is *java* with 0.31.

The Figure 2 contains four figures that show the likelihood for the first five keywords for the four tags. It is worth mentioning, there are keywords that appear in more than one tag, for instance, the keyword *file* is one of the top five keywords for *c++*, *java*, and *php*.

For the evaluation with consider that one question has only one tag. The experiment used from 1 top keyword to 20 top keywords. The Figure 3 shows the accuracy for these experiments. The best result using the Bayesian Data Analysis approach was using 1 top keyword (73% of hits), and using 5 top keywords (61 % of hits).

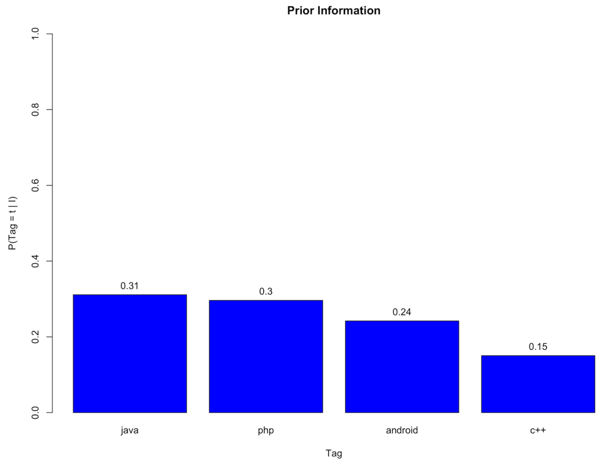
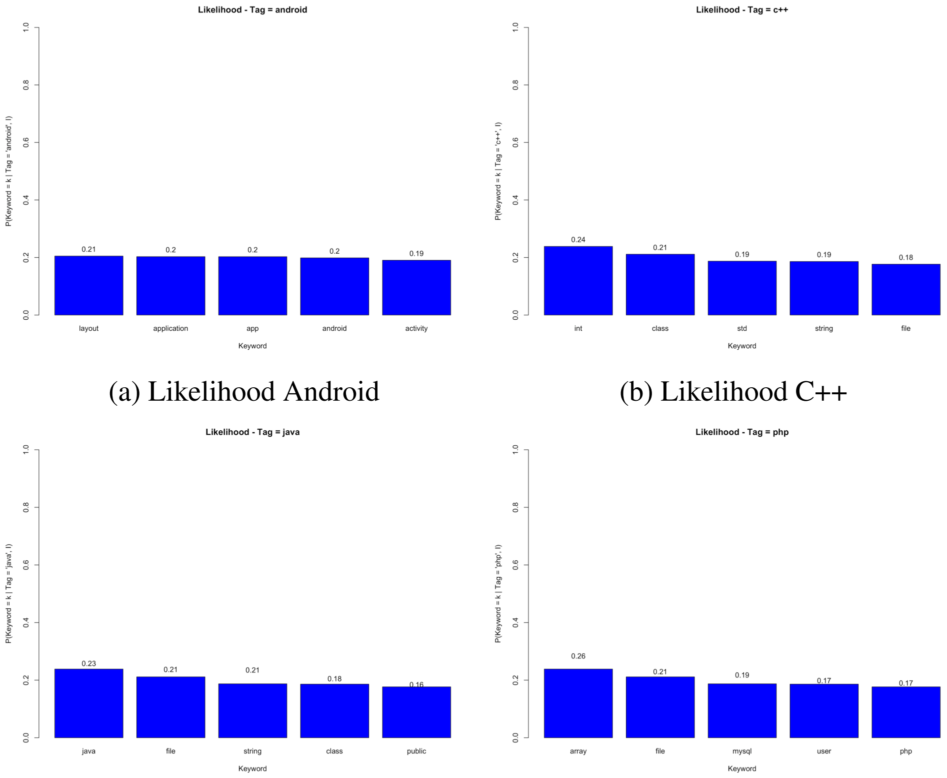


Figure 1: Prior Information



(a) Likelihood Android

(b) Likelihood C++

(c) Likelihood Java

(d) Likelihood Php

Figure 2: Likelihood

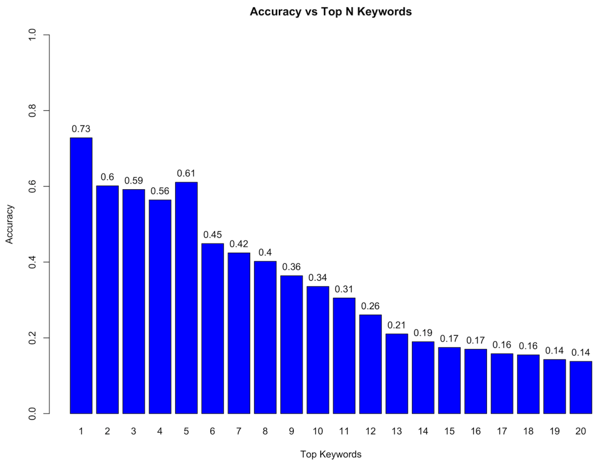


Figure 3: Comparison between difference optimization approaches

4 Conclusion

The Bayesian model assumes the independence between keywords. However, in most cases keywords happen to cooccur. The model we use may lead to some error induced by the assumption of independence. In order to correct this type of error, future work can be done about how to use train dataset to deduce the cooccurrence and incorporate this type of information into the prior knowledge to get more accurate predication of tags. Besides that, the model is limited because we already chose the tags. In order to make the model work in practice, we need to figure out a method to choose the set of the most possible candidate tags, and thus reduce the computation cost.

As for the reason why we choose R over Matlab lies on the fact that R is open source and it has a richer text mining ability to further the future work.

References

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 40(1):45–65.

Facebook (2013a). Facebook. http://www.facebook.com.

Facebook (2013b). Facebook recruiting iii - keyword extraction. https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction.

Feldman, R. and Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.

Silva, C. and Ribeiro, B. (2003). The importance of stop word removal on recall values in text categorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 1661–1666. IEEE.

SS64 (2013). tr man page. http://ss64.com/bash/tr.html.

StackExchange (2013a). Stack exchange - free, community-powered q & a. http://stackexchange.com.

StackExchange (2013b). Stack overflow. http://stackoverflow.com.