1 Readme

1.1 Methodology

The problem is to classify the category of a new book based on labeled books. The first file "input1.txt" contains two types of features author and title. According to the training part data , there are 270 books, 476 authors scatter on 9 categories. The average books published by each author is less than 1. Intuitively, if an author published a book in category A, it is more likely his new book will also belong to the same category if he has new publications. However, the statistic shows only few authors will publish more than one books. So, the author may not be the dominated factor to decide whether a book belongs to a category.

The second feature is title. So, this problem is similar to a text classify problem. In general, there are two kinds of approaches: discriminative model and generative model. This program selects generative model and uses navie bayes to do the classification.

The Navie Bayes method for text classification has been studied in [1] and [2]. Name the random variable y for the category, and $x = (x_1 \cdots x_{|D|})$ for the D dimensional features. The classify is to find $\hat{y}$ to have max posterior probability $P(y|x)$. According to Bayes rule:

$$\begin{aligned} \hat{y} &= \underset{y}{\operatorname{argmax}}\, P(y|x) && (1)\\ &= \underset{y}{\operatorname{argmax}}\, \frac{P(x|y)P(y)}{P(x)} && (2)\\ &= \underset{y}{\operatorname{argmax}}\, P(x|y)P(y) && (3) \end{aligned}$$

The reason it is called Navie Bayes is because the method assumes the features are independent such that

$$P(x|y) = P((x_1 \cdots x_{|D|})|y) = \prod_{1 \le i \le |D|} P(x_i|y) \tag{4}$$

thus

$$\begin{aligned} \hat{y} &= \underset{y}{\operatorname{argmax}} \log[\prod_{1 \le i \le |D|} P(x_i|y)P(y)] && (5)\\ &= \underset{y}{\operatorname{argmax}}[\log P(y) + \sum_{1 \le i \le |D|} \log P(x_i|y)] && (6) \end{aligned}$$

Even though this assumption is false, it makes model easy to fit and works well in practice [1]. So the question changes to find the prior probability $P(y)$ and conditional probability $P(x_i|y)$.

Given the categories are defined by C, assume y follows the multinomial distribution, then $P(y = c)$ can derived by the max likelihood estimation, such that:

$$P(y=c)_{MLE} = \frac{N_c}{N} \tag{7}$$

where N_c is the number of books belong to category c, and N is the total number of books.

To derive $P(x_i|y)$, we make another assumption that the positions of each words are independent. This assumption breaks the order of words and treat the document as a bag of words, and x is also treated as a multinomial random variable. Name T for the words set, and $|D| = |T|$.

$$P(x_i = t|y=c)_{MLE} = \frac{N_{tc}}{\sum\limits_{t' \in T} N_{t'c}} \tag{8}$$

where N_{tc} is the times the word t appears in the books of category c.

To deal with the case where $N_{tc} = 0$, 8 is updated as:

$$P(x_i = t|y=c)_{MLE} = \frac{N_{tc}+1}{\sum\limits_{t' \in T} (N_{t'c}+1)} \tag{9}$$

The author information is treated as the text word and processed in the same way.

The algorithm is as follows:

Algorithm Book Classify

Train
$T \leftarrow$ extract words from training books
for c in C
    for t in T
        $N_{tc} \leftarrow$ times word t appears in books of category c
        update $P(x_i = t|y=c)$ as Equation 9
Test
for c in C
    $W \leftarrow$ extract words from testing book
    $pr(c) = \log P(y=c)$
    for w in W
        $pr(c) += \log(P(x=t|y=c))$
return $\text{argmax}_c\, pr(c)$

1.2 Improve Accuracy

The accuracy improvement focuses on feature selection, *i.e.,* which word should be included as feature. A trivial method is to filter *preposition*, *article* and other words which are commonly used and have no category preference, but it can not filter out all possible unrelated words. Manning *et al.* [2] discussed three methods of feature selection for text classification: mutual information, Chi^2 and frequency based.

This program utilizes the mutual information method. The mutual information (MI) evaluate how much information the presence/absence of the word will contribute the correct the classification, *i.e.,* , the correlation between word and category.

Define random variable $e_c = 0, 1, e_t = 0, 1$ for the presence of book in category and word in book.

$$MI(t, c) = \sum_{e_c} \sum_{e_t} P(e_t, e_c) \log \frac{P(e_t, e_c)}{P(e_t)P(e_c)} \tag{10}$$

where $P(e_t)$ is the probability of word t appear in any category books; $P(e_c)$ is the probability of a book belongs to category c; $P(e_t, e_c)$ is the join probability of the presence of word t and category c. For example, $P(e_t = 1, e_c = 1)$ is the probability word t appears in category c's books. Since both e_c, e_t follows multinomial distribution, the probability can be estimated as:

$$P(e_t = 1, e_c = 1) = \frac{\text{\#books in category c contain t}}{\text{\#books in category c}}$$

$$P(e_c = 1) = \frac{\text{\#books in category c}}{\text{\#books in all categories}}$$

other cases can be developed similarly.

Table **??** shows the top 10 MI words in each category.

Table 1: Individual Features Weight

AMEH	BIOL	CS	CRIM	ENG
reconstruction	biology	files	criminal	writing
civil	cell	programming	crime	reading
revolution	dna	file	crimes	essay
west	cells	arrays	police	words
empire	gene	variable	court	sentence
america	proteins	software	investigation	revising
war	chromosomes	user	sentencing	write
south	molecular	input	victims	plagiarism
north	genetics	operators	constitutional	punctuation
cold	endocrine	converting	justice	narrative
MANAG	MARKET	NURSE	SOCI	
management	marketing	nursing	social	
teams	pricing	clinical	sociology	
managerial	sales	practice	stratification	
performance	buying	diagnostic	gender	
resource	selling	nurses	inequality	
leading	advertising	care	poverty	
organizational	markets	health	race	
business	segmentation	therapeutic	family	
contingency	consumer	assessment	experience	
employees	product	diagnosis	sociological	

1.3 Experiment

1.3.1 input1

To scale to big data input, a database version program is developed to store relevant tables in MySQL (A non-database version program is also developed which stores all information in memory).

1.3.2 input1&2

1.4 How to run

References

[1] P. Murphy Kevin. Naive bayes classifiers. http://www.cs.ubc.ca/ murphyk/Teaching/CS340-Fall06/reading/NB.pdf/, 2006. [Online; accessed 12-Feb-2014].

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.