Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Дискретный анализ»
на тему: «Автоматическая классификация документов».

Студент: К. А. Калугин
Преподаватель: С. А. Сорокин
Группа: М8О-307Б

Дата:
Оценка:
Подпись:

Москва, 2021

Курсовая работа

Задача: Требуется разработать программу, получающую на вход сначала обучающие, а потом тестовые данные. При помощи наивного алгоритма Байеса программа должна обучится на первой части данных и классифицировать вторую часть.

Формат входных данных: В начале на вход подаются 2 числа - n и m - количество обучающих и количество тестовых данных. После это следует n строк формата <класс> (0 - спам, 1 - не спам) <текст письма> (оканчивающийся символом конца строки). После этого следует m строк вида <текст письма>, каждую из которых надо классифицировать.

Формат результата: Число 0, если тестовое сообщение является спамом или 1, если нет.

1 Описание

Как указано в [1] наивный байесовский классификатор — простой вероятностный классификатор, основанный на применении теоремы Байеса: $P(A|B) = \frac{P(B|A)P(A)}{P(d)}$.
По сути, он основан на выведенной в [**?**] формуле: $C = log(\frac{D_c}{D}) + \sum_{i=Q} log(\frac{W_{ic}+1}{|V|+L_c})$,
где:

D_c — количество документов в обучающей выборке принадлежащих классу c;

D — общее количество документов в обучающей выборке;

$|V|$ — количество уникальных слов во всех документах обучающей выборки;

L_c — суммарное количество слов в документах класса c в обучающей выборке;

W_{ic} — сколько раз i-ое слово встречалось в документах класса c в обучающей выборке;

Q – множество слов классифицируемого документа (включая повторы).

Вычислив C для всех классов (Spam и Ham), необходимо сравнить их. Документ относится к тому классу, C которого больше.

2 Исходный код

Программа считывает входящие строчки, раскладывает их на слова и обрабатывает те одним из двух способов: если строка относится к обучающим, слово вносится в std::map и сохраняется. Если же, обучающие строки закончились и строка относится к тестовым, то для нее вычисляется $C_{spam}C_{Ham}$, которые сравниваются. После этого, в зависимости от результата сравнения, выносится вердикт - является ли строка спамом или нет.

```
1  #include <iostream>
2  #include <map>
3  #include <string>
4  #include <math.h>
5
6  using namespace std;
7
8  int isSpam (map <string, pair <int, int>> & dictionary, string & letter, int training,
       pair <int, int> & trainingSpam, pair <int, int> & trainingHam) {
9      map <string, pair <int, int>> :: iterator it;
10     double spamChance = log (double (trainingSpam.first) / training);
11     double hamChance = log (double (trainingHam.first) / training);
12     //cout << "MS:" << dictionary.size () << endl << "SC:" << spamChance << endl << "HC
           :" << hamChance << endl << "TSpam:" << trainingSpam.first << " " <<
           trainingSpam.second << endl << "THam:" << trainingHam.first << " " <<
           trainingHam.second << endl;
13     string word;
14
15     for (int i = 0; i < letter.size (); i ++) {
16         if ((letter [i] >= 'A') && (letter [i] <= 'Z')) {
17             letter [i] -= 'A';
18             letter [i] += 'a';
19         }
20         if (letter [i] != ' '){
21             word.push_back (letter [i]);
22             if (i != letter.size () - 1) {
23                 continue;
24             }
25         }
26         it = dictionary.find (word);
27         if (it != dictionary.end ()) {
28             spamChance += log (double((it->second.first + 1)) / (dictionary.size () +
                   trainingSpam.second));
29             hamChance += log (double((it->second.second + 1)) / (dictionary.size () +
                   trainingHam.second));
30             //cout << "Word: " << word << endl << "WSpam:" << it->second.first << endl
                   << "WHam:" << it->second.second << endl;
31         }
32         else {
```

```
33            spamChance += log (1.0 / (dictionary.size () + trainingSpam.second));
34            hamChance += log (1.0 / (dictionary.size () + trainingHam.second));
35        }
36        word.clear ();
37    }
38    //cout << "SPAM:" << spamChance << endl << "HAM:" << hamChance << endl;
39    if (spamChance > hamChance) {
40        return 1;
41    }
42    else {
43        return 2;
44    }
45    return 0;
46 }
47
48 int main () {
49    map <string, pair <int, int>> dictionary;
50    map <string, pair <int, int>> :: iterator it;
51    int training, testing;
52    pair <int, int> trainingSpam, trainingHam;
53    cin >> training >> testing;
54    for (int i = 0; i < training; i ++) {
55        int classBuffer;
56        string letter;
57        cin >> classBuffer;
58        getchar ();
59        if (classBuffer == 0) {
60            trainingSpam.first ++;
61            getline (cin, letter);
62            string word;
63            for (int i = 0; i < letter.size (); i ++) {
64                if ((letter [i] >= 'A') && (letter [i] <= 'Z')) {
65                    letter [i] -= 'A';
66                    letter [i] += 'a';
67                }
68                if (letter [i] != ' '){
69                    word.push_back (letter [i]);
70                    if (i != letter.size () - 1) {
71                       continue;
72                    }
73                }
74                pair <int, int> bufferpair (1, 0);
75                it = dictionary.find (word);
76                trainingSpam.second ++;
77                if (it != dictionary.end ()) {
78                    it->second.first ++;
79                }
80                else {
81                    dictionary.insert(make_pair(word, bufferpair));
```

```
82                  }
83                  word.clear ();
84              }
85          }
86          else if (classBuffer == 1) {
87              trainingHam.first ++;
88              getline (cin, letter);
89              string word;
90              for (int i = 0; i < letter.size (); i ++) {
91                  if ((letter [i] >= 'A') && (letter [i] <= 'Z')) {
92                      letter [i] -= 'A';
93                      letter [i] += 'a';
94                  }
95                  if (letter [i] != ' '){
96                      word.push_back (letter [i]);
97                      if (i != letter.size () - 1) {
98                         continue;
99                      }
100                 }
101                 pair <int, int> bufferpair (0, 1);
102                 it = dictionary.find (word);
103                 trainingHam.second ++;
104                 if (it != dictionary.end ()) {
105                     it->second.second ++;
106                 }
107                 else {
108                     dictionary.insert(make_pair(word, bufferpair));
109                 }
110                 word.clear ();
111             }
112         }
113     }
114     for (int i = 0; i < testing; i ++) {
115         string letter;
116 
117         getline (cin, letter);
118         int sOrH = 0;
119         sOrH = isSpam (dictionary, letter, training, trainingSpam, trainingHam);
120         if (sOrH == 1) {
121             cout << 0 << endl;
122         }
123         else if (sOrH == 2) {
124             cout << 1 << endl;
125         }
126     }
127     return 0;
128 }
```

3 Консоль

```
PS C:\VSC\DA>.\a.exe
10 5
1
You have 5 notifications .
0
-70% sale on BLACK FRIDAY
0
You have won 1000000$ !!!
1
Congratulations we won first round !
0
Notification that you won 1000000$
1
Will you go on shopping during black friday ?
0
You can buy a car right now !
0
You have -100% sale ! You can get air conditioner for free right now !
1
How much is 500$ in rubles ?
1
Hi ! We want to buy a car .
Is 500$ a lot ?
1
How much is 1000000$ in rubles ?
1
You got -70% sale on air conditioner !!!
0
Our new black car costs about 1000000$ ! We never had a car before . We will
buy it next friday .
1
Congratulations ! This is notification that you can still get from -70% to
-100% sale !!!
0
PS C:\VSC\DA>.\a.exe
10 6
1
You was invited to LP MAI educational program .
0
```

```
You was invited to free online courses !
0
You was invited to online quiz !
1
You was invited to MAI open door day .
0
You was invited to Codeforces Round 693 .
0
You was invited to flat earthers meeting .
0
You was invited to round earthers meeting .
1
You was invited to meeting with MAI professors .
1
You was invited to online MAI meeting .
0
You was invited to educational quiz !
Hi ! I want to invite you to MGU open door day ! Don‘t worry it is free .
1
My LP program is not working . I need your help .
1
You was invited to EDUCATIONAL flat earthers meeting !
0
You was invited to MAI online quiz with our proffesors !
0
You was invited to codeforce quiz ! Feel free to invite your friends .
0
You was invited to DA MAI educational program .
1
```

4 Выводы

Выполнив курсовую работу по предмету «Дискретный анализ», я узнал об алгоритме, позволяющем определять примерный смысл текста, основываясь на том, из каких слов он составлен и том, как часто эти слова встречались в уже классифицированных текстах. Подобные алгоритмы активно используются в наше время, например в почтовых клиентах. Кроме того, мне было интересно реализовать алгоритм, который может иметь прикладное применение.

Список литературы

[1] *Википедия-Наивный байесовский классификатор.*
URL: `https://ru.wikipedia.org/wiki/Наивный_байесовский_классификатор` (дата обращения: 16.12.2013).

[2] *Наивный байесовский классификатор.*
URL: `http://bazhenov.me/blog/2012/06/11/naive-bayes.html` (дата обращения: 16.12.2013).