VYSOKÉ UČENIE TECHNICKÉ V BRNE
FAKULTA INFORMAČNÝCH TECHNOLOGIÍ

Bezpečnost informačních systémů (BIS)

2017/2018

Detekcia spamu

Ľubomír Gallovič (xgallo03)

December 2017

1 Úvod

Cieľom projektu bolo spracovať vstupný emailový súbor a určiť či ide o nevyžiadanú poštu (spam), alebo nie. Pre klasifikáciu bol zvolený prístup machine learning, konkrétne Bayesovo filtrovanie, pretože ide o relatívne jednoduchý ale presný spôsob detekcie spamu.

2 Priblíženie použitého prístupu

Bayesovo filtrovanie je štatistickou technikou pre detekciu spamu. Pred použitím je túto metódu potrebné natrénovať, teda získať štatistické data z veľkého množstva spamových a nespamových emailov. Následne sú tieto data použité k výpočtu skóre, určujúceho či email je, alebo nie je spam. Podľa toho, ktoré skóre je vyššie je daný email príslušne klasifikovaný. V trénovacej fázi je zistený počet výskytov slov v spamových a nespamových emailoch a matematickým vzorcom je následne vypočítaná podmienená pravdepodobnosť daného slová pre obidve kategórie. Pri samotnom skórovaní emailu sa skóre vypočíta na základe podmienených pravdepodobností jednotlivých slov nachádzajúcich sa v emaili, a podľa toho, ktoré zo skóre "spamovosti" a "nespamovosti" je vyššie, je správa príslušne klasifikovaná.

3 Vybraná metóda Bayesovho filtrovania

Existuje viacero variánt Bayesovho filtrovania, pričom vybraná bola technika "Multinomial Naive Bayes", pretože dosahovala konzistentne najlepšie výsledky[1]. Bola testovaná v obidvoch variantách: "term frequency" a "binary". Matematický vzorec pre výpočet skóre daného emailu je nasledovný:

$$P(c|d) = P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

Jedná sa teda o súčin podmienených pravdepodobností slov nachádzajúcich sa v emaili $P(t_k|c)$, a pravdepodobnosti patrenia emailu do danej kategórie $P(c)$ (napríklad do spamu patrí 60% všetkých poslaných emailov, teda daná správa má väčšiu pravdepodobnosť, že ide o spam). Násobením pravdepodobností však môže dôjsť k floating point underflow chybe, teda v praxi sa využíva výpočet skóre cez súčet logaritmov pravdepodobností.

$$log(P(c)) + \sum_{1 \leq k \leq n_d} log(P(t_k|c))$$

V trénovacej fázi je potrebné vypočítať podmienenú pravdepodobnosť jednotlivých slov, čo sa robí pomocou nasledujúceho vzorca:

$$P(t|c) = \frac{T_{ct} + 1}{(\sum_{t' \epsilon V} T_{ct'}) + B'}$$

Ide teda o podiel počtu výskytov daného slova vo všetkých emailoch danej kategórie a celkového počtu slov. Pripočítanie 1 v čitateli a B' v menovateli sa robí z dôvodu Laplacelovho vyhladzovania, pričom B' je počet unikátnych slov vo všetkých emailoch danej kategórie. Rozdiel medzi "term frequency" a "binary" je v tom, že v binary sa pre každý email pripočíta maximálne jeden výskyt slova, aj keby jeho výskytov bolo viac.

4 Zhomaždené emaily na trénovanie

Na machine learning bolo potrebné zhromaždiť veľké množstvo spamových aj nespamových emailov. Boli použité verejne prístupné emaily zo šiestich korpusov: Enron-spam, CSDMC2010 , SpamAssassin, TREC 2005, TREC 2006 a TREC 2007. Dokopy bolo zozbieraných 247 640 emailov, pričom 145 165 bolo spamových. Emaily z každého korpusu boli rozdelené na trénovacie a testovacie a to približne v pomere 2:1. Zdrojový kód na trénovanie sa nachádza v súbore `training.py` a vygeneruje dva súbory `hamData.pickle` a `spamData.pickle`, ktoré sú potebné pre správne fungovanie klasifikátora emailov.

5 Testovanie vytvoreného detektoru spamov

Zdrojový kód pre klasifikáciu emailov sa nachádza v súbore `test.py`. Na testovanie bola vyhradená približne tretina všetkých zozbieraných emailov, pričom tento pomer bol udržaný pre každý korpus. Pri skórovaní nebola zohľadnená pravdepodobnosť patrenia emailu do danej kategórie $P(c)$, pretože nie je možné predpokladať, že pomer spamových a nespamových emailov bude pri testovaní rovnaký ako v reálných situáciách. Pri výpočte skóre "nespamovosti" emailu je k výsledku pripočítaná konštantná hodnota, ktorá zapríčiňuje, že program má vyššiu tendenciu klasifikovať emaily ako vyžiadanú poštu. To je z dôvodu preferencie menšieho množstva falošne pozitívnych výsledkov, na úkor väčšieho množstva falošne negatívnych. Lepšie výsledky boli dosiahnuté variantou algoritmu "term frequency", preto bola táto varianta použitá vo finálnej verzii. Výsledky programu u jednotlivých korpusov sú nasledovné:

	Úspešnosť detekcie HAMu	Úspešnosť detekcie SPAMu
Enron-spam	99.595 %	63.060 %
CSDMC2010	99.660 %	62.318 %
Spamassassin	98.623 %	65.099 %
TREC 2005	99.402 %	85.297 %
TREC 2006	97.385 %	66.606 %
TREC 2007	99.638 %	64.963 %

Dosiahnuté hodnoty detekcie spamu by bolo možné zvýšiť zmenšením modifikátora skóre "nespamovosti" emailu, pričom počet falošne pozitívnych výsledkov by sa zásadne nezvýšil. Pre potreby projektu to však nie je nutné.

6 Záver

V tomto projekte bol vytvorený program pre detekciu spamov. Program dokáže detekovať nevyžiadanú poštu s presnosťou, ktorá je dostatočná pre potreby projektu. Celkovú úspešnosť by bolo možné zvýšit vhodnou zmenou modifikátora skóre. Program bol písaný v jazyku Python3 a testovaný na systéme Ubuntu 16.

References

[1] Metsis V.; Androutsopoulos I.; Paliouras G. *Spam Filtering with Naive Bayes - Which Naive Bayes?*. 2006, [Online]
https://classes.soe.ucsc.edu/cmps290c/Spring12/lect/14/CEAS2006_corrected-naiveBayesSpam.pdf