UIR Seminární práce

Jan František Sedláček

21. května 2020

Obsah

1 Problematika

1.1 Zadání

Ve zvoleném programovacím jazyce navrhněte a implementujte program, který umožní klasifikovat textové dokumenty do tříd podle jejich obsahu, např. počasí, sport, politika, apod.

1.2 Analýza

Řešení komplexní úlohy jako je strojové učení pro klasifikaci lze rozdělit na dílčí kroky

- Výběr programovacího jazyka
- Výběr vhodných algorithmů
- Samotná implementace řešení
- Testovaní a cross-validace výsledků

1.2.1 Zvolený programovací jazyk

Jako zvolený jazyk pro implementaci programu byla vybrána java ve verzi 11, o proti jiným jazykům například C nebo C++, sice výkonostně zaostává ale rychlé čtění a zpracování textových souborů jsou při úloze ve které je očekávána práce s velkým množstvím těchto dat velkou výhodou.

Úloha nevyžaduje složitejší UI proto bylo zvoleno swing, jedná se grafickou nadstavbu javy a hlavní výhodou je urychlení rychlosti práce.

Nebyly použity žádné jiné externí nebo nestandartní knihovny

1.2.2 Algoritmy

Popis jednotlivých vybraných algoritmů

Najivní Bayes

Naivní bayesovská klasifikace je druh jednoduchého pravděpodobnosti bayesovské klasifikace vychází z Bayesova teorému se silnou (tzv naivní) nezávislost hypotéz. Naivní Bayesův klasifikátor nebo Bayesův naivní klasifikátor, patří do rodiny lineárních klasifikátorů.

Vhodnějším termínem pro základní pravděpodobnostní model by mohl být „model se statisticky nezávislými charakteristikami“.

Jednoduše řečeno, naivní Bayesovský klasifikátor předpokládá, že existence charakteristiky třídy je nezávislá na existenci dalších charakteristik. Například: Ovoce lze považovat za jablko, je-li červené, zaoblené a asi deset centimetrů veliké. I když jsou tyto vlastnosti ve skutečnosti spojeny, naivní Bayesovský klasifikátor určí, že ovoce je jablko, a to nezávisle na těchto vlastnostech barvy, tvaru a velikosti.

K-NN

k-NN je neparametrickou metodou používanou pro klasifikaci a regresi. V obou případech jde o klasifikaci záznamu do kategorie, do které patří nejbližší sousedé k, do prostoru charakteristik identifikovaných učením. Výsledek závisí na tom, zda se algoritmus používá pro účely klasifikace nebo regrese:

Ve třídě k-NN je výsledkem třída členství. Vstupní objekt je klasifikován podle většinového výsledku statistiky třídy členství svých nejbližších sousedů (k je obecně malé kladné celé číslo). Pokud k = 1, pak je objekt přiřazen ke třídě členství svého blízkého souseda. v regresi k-NN je výsledkem hodnota tohoto objektu. Tato hodnota je průměrem hodnot k nejbližších sousedů. Metoda k-NN je založena na předchozím učení nebo slabém učení, kde je funkce hodnocena lokálně, přičemž konečný výpočet se provádí na konci klasifikace. Algoritmus k -NN patří mezi nejjednodušší z algoritmů strojového učení.

1.2.3 Implementace

Datová reprezentace

Bag of words Zjednodušená reprezentace používaná při zpracování přirozeného jazyka a při získávání informací . V tomto modelu je text (fráze nebo dokument) reprezentován jako množina jeho slov („bag“), bez ohledu na gramatickou strukturu a dokonce i na jejich seřazení, při zachování jejich multiplicity.

```
// bag of words Naive bayes vyhodnoceni
 for(HashMap.Entry<String, Integer> entry :
     bag.getBagHashMap().entrySet())
        {
            String key = entry.getKey();

            for (BagOfWords bag1: trainedClasses)
            {
                if (bag1.countOccurrences(key)>0)
                {
                    double pX=bag.countOccurrences(key);

                    double pY=(bag.getBagHashMap().size());

                    double z=(pX/pY);

                    double pX2=bag1.countOccurrences(key);

                    double pY2=(bag1.getBagHashMap().size());

                    double z2=(pX2/pY2);

                    double c=z*z2;

                    bag1.setWeight( bag1.getWeight() + c);

                }

            }

        }
```

IDF Nezpracovaná“ frekvence termínu je jednoduše počet výskytů tohoto termínu v uvažovaném dokumentu (mluvíme o „frekvenci“ jazyka).

```
// Naive Bayes IDF vyhodnoceni
    for(HashMap.Entry<String, Integer> entry :
        bag.getBagHashMap().entrySet())
        {
            String key = entry.getKey();

            for (IDF bag1: trainedClasses)
            {
                for (Word w:bag1.getWordsSets().getWordList())
                {
                    if (w.getValue().equals(key))
                    {
                        bag1.setWeight(bag1.getWeight()+w.getFreq());
                        bag1.setFileCount(bag1.getFileCount()+1);
                    }
                }

            }

        }
```

TFIDF Numerická míra, která vyjadřuje relevanci slova pro dokument ve sbírce. Tato reprezentace se často používá jako váhový faktor při získávání informací a těžbě textu. Hodnota tf-idf se úměrně zvyšuje, kolikrát se slovo v dokumentu objeví, ale zároveň je kompenzováno frekvencí slova ve sbírce dokumentů.

```
//Naive Bayes TFIDF vyhodnoceni
   for(HashMap.Entry<String, Integer> entry :
       bag.getBagHashMap().entrySet())
        {
            String key = entry.getKey();

            for (TFIDF bag1: trainedClasses)
            {
                for (Word w:bag1.getWordsSets().getWordList())
                {
                    if (w.getValue().equals(key))
                    {
                        bag1.setWeight(bag1.getWeight()+w.getFreq());
                        bag1.setFileCount(bag1.getFileCount()+1);
                    }
                }

            }

        }
```

2 Testování

2.1 Testovací data

Program byl testován na několika korpuses s několika různých oborů

- Základní zadaný korpus (historické noviny) dále je korpus
  - Zadaný předanatovaný soubor starých novinových článků
    * pol
    * kul
    * err
    * ...
- Jazykové rozpoznání (články z `wikipedia.org` v různých jazycích)
  - několik odborných článků z různých oborů v různých jazycích sloužící k rozpoznání jazyků
    * cs (Čeština)
    * en (Angličtina)
    * ge (Němčina)
- Reuters korpus (články z portálu reuters rok 1987)
  - Staré reuters články ve formátu sgm s několika možnostmi klasifikace (topics, places, writers)
    * earn
    * oil
    * oil-palm
    * ...
- Rozpoznání moderních novin (Články z portálu `novinky.cz`)
  - novodobé novinové články ze tří katogorií
    * krimi
    * sport
    * pol (politika)

2.2 Postup

1. Data pro testování jsou nahrána do vyrovnávací paměti pro testovací data
2. Program je naučen jednotlivě 6 algoritmů
3. Ručně je testováno několik souborů pro kontrolu nahrání dat
4. Automaticky je vyhodnocený zbytek dat, výsledkem je přesnost

2.2.1 Přesnost

```
Cn = celkový počet tříd
Ci = na jaké pozici byla určena první správná třída
```

$ACC = \frac{\sum(100/Ci^2)}{Cn}$

2.3 Výsledky

2.3.1 Korpus

- Bag of world
  - Naive Bayes = 66[%]
  - IDF = 13[%]
  - TF-IDF = 14[%]
- K-NN
  - Naive Bayes = 66[%]
  - IDF = 15[%]
  - TF-IDF = 14[%]

Trace

```
Classes File loaded: 21
Train File loaded: 412
Bag of Word Created
IDF Created
TF-IDF Created
Test File loaded: 99
Test File loaded: 99
```

2.3.2 Wiki jazyky

- Bag of world
  - Naive Bayes = 100[%]
  - IDF = 66[%]
  - TF-IDF = 54[%]
- K-NN
  - Naive Bayes = 100[%]
  - IDF = 66[%]
  - TF-IDF = 54[%]

Trace

```
Classes File loaded: 3
Train File loaded: 19
Bag of Word Created
IDF Created
TF-IDF Created
Test File loaded: 19
```

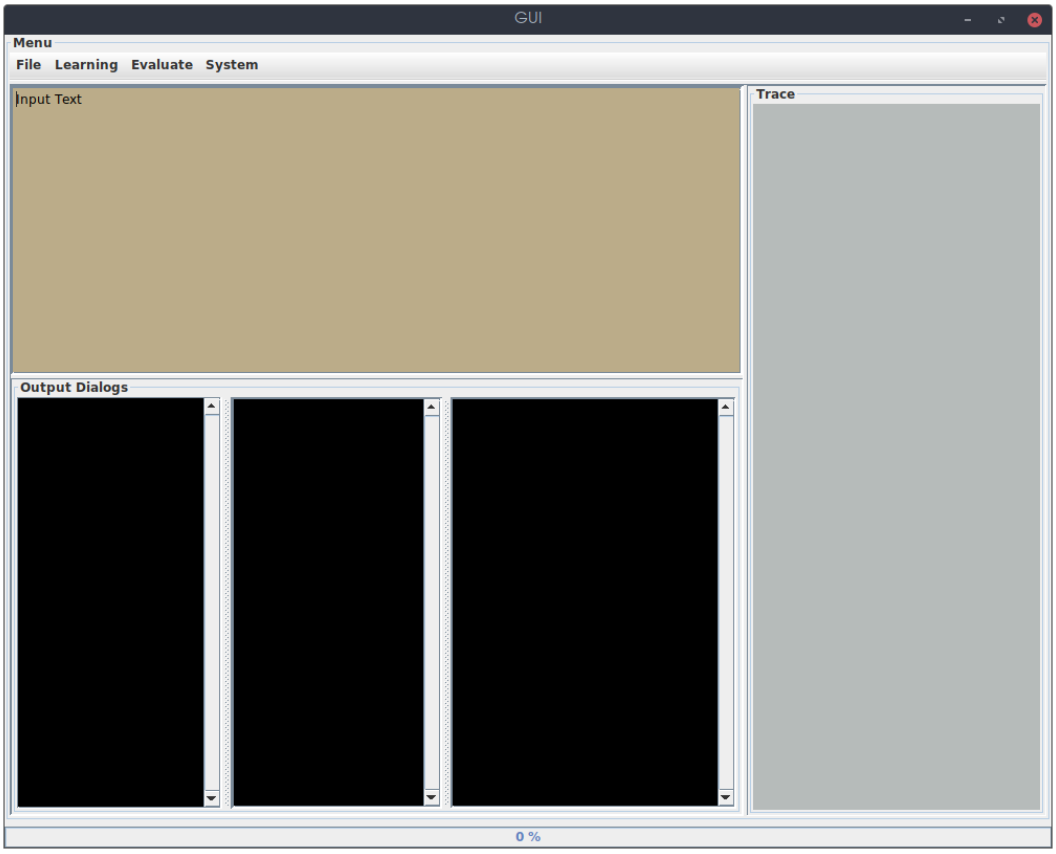
2.3.3 NovinkyCZ

Testováno ručně. Články brány z portálu `ct24.cz`. Převážně lépe vyhodnocovali IDF a TFIDF.

2.3.4 Reuters

Kvůli velkému množství dat a podobě souborů je učení a testování extrémně zdlouhavý proces

3 Uživatelská příručka



Položky menu

- File
  - Save ... (uložení modulů)
  - Read ... (načtení modulů)
- Learning
  - Learn ... (učení podle načtených dat !!nejdříve je třeba načíst data!!)
  - Read Train Data (spustí dialog pro načtení dat pro učení)
- Evaluate
  - ... [Textbox input] (vyhodnotí ručně zadaný text)
  - ... [Accuracy Test] (spustí dialog pro automatické testovaní)

- System
  - Clear Trace (vymaže logy)
  - Realtime testing "..."(spustí nebo ukončí automatické vyhodnocování ručně zadaného textu)
  - Exit (ukončí program)

Output Dialogs

- dialogová okna zobrazující výsledky

Trace

- logovací okno

4 Závěr

Výsledné procenta nebyla scela podle očekávání především u IDF a TFIDF. Při změně testovacích dat se ukázalo že IDF a TFIDF jsou více závislé na velikosti trénovací množiny a u TFIDF také počtů jednotlivých dokumenků. Naivní Bayes se ukázal jako o něco méně přesnější než K-NN ale minimálně v jazyce Java byl výkonostně lepší ovšem nedá se z toho vyvodit v tomto ohledu žádný závěr, protože by muselo dojít k testům na více jazycích a to není účelem této práce. Program nicméně splnuje základní požadavky. Vylepšení by mohlo přijít v podobě přesnějších algoritmů, vylepšení UI nebo přidání české lokalizace.