Primjena Blumovog filtera u algoritmima detekcije noviteta

Damir Delijic 1/21 C

Contents

[Definicija problema 3](#_Toc137463985)

[Detekcija noviteta 3](#_Toc137463986)

[Blumov filter 3](#_Toc137463987)

[Tekstualni dokumenti 3](#_Toc137463988)

[Koncept algoritma 4](#_Toc137463989)

[Algoritam 5](#_Toc137463990)

[Poredjenje sa drugim metodama 5](#_Toc137463991)

[Privremeno 5](#_Toc137463992)

[Literatura 6](#_Toc137463993)

# Definicija problema

Zadatak se sastoji u dizajniranju i implementaciji algoritma za detekciju noviteta u skupu tekstualnih dokumenata. Algoritam treba da bude zasnovan na odgovarajucoj modifikaciji ideje na kojoj je zasnovan Blumovog filtera.

# Detekcija noviteta

Modeli istrenirani za detekciju noviteta posjeduju sposobnost prepoznavanja novih obzervacija, odnosno onih koje se u odredjenoj mjeri ne uklapaju u poznatu distribuciju. Detekcija noviteta se smatra polu-nadgledanom metodom detekcije anomalija. Trening podaci se sastoje iz iskljucivo cistih obzervacija koje ne sadrze anomalije. Sistem bi nakon treninga trebao da ima mogucnost davanja odgovora na pitanje “Da li je nova obzervacija drugacija od poznatih?” sa odredjenom sigurnoscu.

# Blumov filter

Blumov filter je prostorno efikasna probabilisticka struktura podataka koja se koristi u svrhe testiranja pripadnosti elementa nekom skupu. Lazno pozitivni rezultati su moguci, dok lazno negativni nisu. Ideja je da se elementi mapiraju u koristeci neku heš funkciju , i onda testirati pripadnost provjeravanjem da li vazi

Prazni Blumov filter je niz od bita postavljenih na . Definise se razlicith heš funkcija, od kojih svaka hešira element skupa u jednu od pozicija niza bitova, generisuci unformnu nasumicnu distribuciju. Tipicno je mala konstanta koja zavisi od zeljene preciznosti. Velicina niza bita, je proporcionalna broju i broju elemenata poznate distribucije.

Dodavanje elementa se vrsi tako sto se on predaje svakoj od heš funkcija da bi se dobile pozicija u nizu. Zatim se sve te pozicije setuju na .

Provjera da li neki element pripada se vrsi tako sto se preda svakoj od heš funkcija i provjerava se da li su sve od dobijenih pozicija setovane na . Ako jesu onda element pripada skupu, a ako barem jedna nije onda ne pripada.

# Tekstualni dokumenti

Reprezentaciji tekstualnih dokumenata se moze pristupiti na vise nacina. Mogu se predstaviti kao skupovi rijeci koje se pojavljuju, kao skupovi kljucnih rijeci ili kako skupovi šinglova. Jedino pristup sa šinglovima uzima u obzir redoslijed rijeci.

Definicija -šingla je sekvenca od tokena koji se pojavljuju u dokumentu. Tokeni mogu biti karakteri, rijeci ili nesto trece. Izbor tokena zavisi od primjene do primjene.

Dakle dokument je skup šinglova, predstavljen kao nula-jedan vektor. Svaki šingl je dimenzija i vektori su rijetki. Nad dokumentima se definise Jaccard-ova mjera slicnosti . su vektorske reprezentacije dokumenata uz pomoc šinglova . Odnosno . Gdje su:

* - broj koordinata koje su jedinice kod oba vektora
* - broj koordinata koje su jedinice kod a nule kod
* - broj koordinata koje su nule kod a jedinice kod

# Koncept algoritma

Obzirom da detekcija noviteta ukljucuje statisticku analizu kojom se odredjuje pripadnost nekoj distribuciji podataka, nije moguce koristiti Blumov filter sa bilo kakvom heš funkcijom jer bi se na takav nacin moglo provjeriti samo da li je dati element dio trenutnog skupa podataka. Da bi se moglo provjeriti da li je element stvarni novitet (odnosno da nije jedan od vec postojecih i nije dovoljno slican ostalima da bi pripadao poznatoj distribuciji), neophodno je koristiti heš funkciju koja cuva odredjene osobine originalnih podataka. Takvom funkcijom bi vrlo slicne dokumente bilo moguce heširati u istu vrijednost sa velikom vjerovatnoćom.

Jedna takva funkcija, koja cuva udaljenost izmedju dokumenata je *minhashing*. Veza izmedju Jaccard-ove slicnosti i *minhasing*-a je ta da je vjerovatnoca da je heš vrijednost dva dokumenta nakon nasumicne permutacije redova ista jednaka njihovoj Jaccard-ovoj slicnosti. Dokaz se moze naci u knjizi [3] u poglavlju 3.3.3. Heš vrijednost dokumenta je pozicija prve jedinice u njegovom vektoru.

Dokumenti su predstavljeni pomocu matrice tako sto su redovi prestavljaju šinglove, a kolone predstavljaju dokumente. Za -tu kolonu ako je -ti red jedinica znaci da -ti šingl pripada -tom dokumentu.

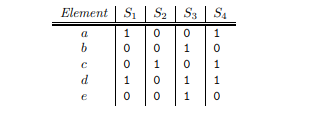


Figure 1: Reprezetacija dokumenata pomocu karakteristicne matrice

# Algoritam

Nije izvodljivo permutovanje velike karakteristicne matrice eksplicitno. Cak je i nasumicno biranje permutacije miliona redova dugotrajno, a sortiranje takvih redova jos dugotrajnije. Pa, tako permutovane matrice, iako su konceptualno primamljive, nije moguce implementirati.

Srecom, moguce je simulirati efekat nasumicne permutacije pomocu heš funkcije koja mapira redne brojeve redova matrice u onoliko bucket-a koliko ima redova. Takva funkcija ce tipicno mapirati neke parove brojeva u iste korpe a ostale ostaviti praznim. Razlika je nebitna dokle god je broj redova velik i nema puno kolizija. Heš funkcija “permutuje” red u poziciju u permutovanom redoslijedu. Pa umjesto biranja nasumicnih permutacija redova, biramo nasumicnih heš funkcija redova.

Implementacija dijela algoritma koji trenira se sastoji iz sledeceg:

* Izracuna se optimalno za dimenzije dataseta.
* Generise se seed-ova za heš funkciju koja ce se koristiti za permutacije redova.
* Za svaki od seed-ova se:
  + Inicijalizuje jedan Blumov filter i vektor potpisa.
  + Na osnovu seed-a se dobija heš funkcija kojom se vrsi implicitna permutacija.
  + Izracuna se vektor hešova (*minhash-ova*) za svaki od dokumenata za datu permutaciju.
  + Filter se popunjava tako sto se postavi jedinica na svakoj od pozicija odredjenih vektorom . za svaki heš iz .
  + Filter i seed koji je koristen se cuvaju za detekciju.

Algoritam detekcije:

* Za svaki od seed-ova se izvrsi permutacija redova ulaznog vektora .
* Za svaku od permutacija se provjerava da li je pozicija u odgovarajucem filteru.
* Ako jeste jedinica za svaku od permutacija onda element sa velikom vjerovatnocom pripada poznatoj distribuciji.
* Ako postoji filter kroz koji ne prolazi onda se racuna kao novitet.

# Poredjenje sa drugim metodama

# Privremeno

# Literatura

[1] <https://scikit-learn.org/stable/modules/outlier_detection.html>

[2] <https://en.wikipedia.org/wiki/Bloom_filter>

[3] http://infolab.stanford.edu/~ullman/mmds/book.pdf