Detecció de similitud de documents amb Hashing

Álvar Hernández, Víctor Alcázar, Albert Ribes

Algorísmia

Grup 10

FIB - UPC

Quatrimestre Tardor 2016-2017

Índex

Capítol 1

Descripció de la pràctica

En aquesta pràctica es demana la implementació i experimentació dels métodes descrits en el llibre "*Mining of Massive Datasets*" [1] sobre la detecció de similitud de documents amb tècniques de hashing. Concretament, es demana la implementació del càlcul de similitud de Jaccard de dos documents representats per conjunts de k-shingles, la implementació del grau de similitud de Jaccard a través d'una implementació via signatures de minhash basades en t funcions de hash, i la implementació d'un algorisme *Locality-Sensitive Hashing* (LSH) basat en signatures minhash. A més a més, també es demana experimentar amb les implementacions fetes per evaluar la correctesa i els temps emprats per fer els càlculs.

Aclariment En alguns idiomes, com poden ser el català o el castellà, l'alfabet eś diferent que en l'anglès, donat que existeixen les lletres amb accents i alguns símbols especials. A més a més, el llibre *Mining of Massive Datasets* [1] no fa no cap tipus de distinció entres les lletres majúscules i les minúscules, cosa que molts ordinadors actualment sí que fan. També s'ha de tenir en compte que en el càlcul de la similitud de documents els signes de puntuació són poc rellevants. Per això hem decidit fer una simplificació del problema. Per a aquesta pràctica es treballarà amb textos que únicament contenen lletres minúscules i el caràcter "espai". Tot i així, per poder treballar amb qualsevol tipus de text hem implementat un filtre que transforma tots els caràcters especials en els descrits. Som conscients que això pot afectar en el càlcul de la similitud, però hem considerat que el canvi és mínim, i que la simplificació fá més fàcil i simple l'execució. A l'hora de executar els programes, l'usuari final no ha de preocupar-se per això, doncs es tracta d'un procés transparent per a ell.

1.1 Representació de les dades

Com que tots els algorismes estan programats en *C++* es fan servir estructures de dades de l'STL. La informació concreta de cada una de elles es pot trobar a la página cplusplus.com. Aquesta és la pàgina que hem utilitzat nosaltres.

Conjunt de k-shingles

Un conjunt de k-shingles és el que resulta d'gafar totes les possibles sub-cadenes de text de longitud k d'un document, sense considerar les sub-cadenes repetides.

Per representar-ho fem servir un set d'strings. La classe set permet guardar elements diferents i consultarlos ràpidament. La classe set de l'STL manté les sub-cadenes ordenades lexico-gràficament.

Bossa de k-shingles

Una bossa de k-shingles es com un conjunt de k-shingles, però sí que té en comple els elements repetits.

Per representar-ho hem fet servir un map d'strings i enters, de manera que l'enter representa la quantitat d'aparicions del shingle dins del text.

Matriu de representació de conjunts de k-shingles

Una matriu de representació de conjunts de k-shingles indica tots els k-shingles que formen part d'un grup de conjunts de k-shingles.

El llibre proposa una matriu booleana respecte el conjunt universal, que conté tots els possibles shingles, i a on un True indica que el conjunt hi és present, i un False que no; però ja indica que aquesta representació no és viable per a la implementació, ja que seria una matriu massa gran i la majoria del valors serien False, és a dir, es tractaria d'una matriu dispersa.

La nostra implementació és un vector de conjunts de k-shingles. D'aquesta manera queden clars els elements del conjunt universal que hi son presents. La seva posició queda implícita en l'string, és a dir, si $k = 5$ l'string " " (5 espais) és a la posició 0 del conjunt universal, mentres que l'estring "zzzzz" és a la última posició. Tenim implementat un mètode que donat un string retorna la posició implícita que té en el conjunt universal.

Signatura de minhash

La tècnica anomenada minhashing, comprimeix sets grans de manera que encara podem deduir la similitud d'aquests, a partir de la seva versió comprimida.

Una signatura MinHash és, el conjunt de valors mínims que pren un set de singles determinat, aplicant-li diferents funcions de hash.

Matriu característica de signatures de minhash

La matriu característica és una taula on es representen els Sets. Les columnes corresponen als sets de shingles (representant els documents) i les files corresponen als elements del set universal, és a dir, tots els shingles que es troben als sets.

En aquesta taula, hi haurà un 1 a la fila f i columna c si l'element de la fila f és un membre del set de la columna c, i un 0 en cas contrari.

Cal notar, que a la implementació representarem aquesta matriu de forma dispersa.

Signature Matrix

La signature matrix, o com l'hem anomenat a la implementació, minhash matrix, és la matriu que conté les signatures MinHash de diferents Sets. Cada signatura està calculada amb les mateixes funcions de hash. Aquesta matriu ens és útil per calcular la similitud de Jaccard entre diferents sets, fent servir les seves signatures de hash.

Capítol 2

Descripció de la implementació

2.1 Similitud de Jaccard

L'objectiu d'aquest apartat és obtenir la similitud de Jaccard de dos documents representats per conjunts de k-shingles. La idea bàsica serà dividir el número d'elements que es troben a la intersecció dels dos documents entre el número d'elements a la unió.

Per tal de fer això, en primer lloc, realitzem un filtrat dels documents dels quals volem obtenir la similitud de Jaccard. Això és degut a que, com hem explicat anteriorment, requerim que els textos únicament continguin lletres en minúscula sense accents, i espais. Un cop filtrats, els transformem en un sol string, i a continuació en un set de k-shingles agafant totes les possibles subcadenes de longitud k de l'string inicial. Ara que tenim els dos documents representats com a un conjunt de k-shingles, podem calcular la seva similitud de Jaccard.

Per fer aquest càlcul canviarem la representació dels shingles com a strings a enters sense signe. Això ens serà útil per a ocupar menys memòria i realitzar els càlculs amb més agilitat. Partint del fet que, dins els sets els shingles estaran ordenats en ordre creixent, l'algorisme empleat, al començament tindrà un punter al primer element de cadascun dels sets. Si el shingle del primer set és igual al del segon incrementarem en 1 la quantitat d'elements a la intersecció i avançarem els dos punters als següents shingles a analitzar, en canvi si el del primer set és més petit que el del segon, o el del segon més petit que el del primer, avançarem el punter del que tingui l'element més petit per tal de trobar un possible element igual al que apunta el punter del set que té l'element més gran. La unió es calcularà sumant la quantitat d'elements que tenen els dos sets i restant-li la intersecció. Finalment retornarem la divisió entre la intersecció i la unió obtingudes anteriorment.

Una versió alternativa que hem implementat és, en lloc de transformar l'string resultant de filtrar el document en un set de k-shingles, el transformem en un bag de k-shingles. En aquest cas l'algorisme canviarà lleugerament ja que ara els k-shingles es trobaran en una estructura diferent. Aquesta estructura, per cada shingle contindrà un enter indicant el número de vegades que aquest apareix en el text. Ara, l'algorisme quan es trobi dos elements iguals incrementarà la intersecció en el mínim del número de vegades que

aquest element apareix en cadascun dels texts. I de la mateixa forma d'abans, quan un dels elements del conjunt de k-shingles sigui més petit que l'altre, s'avançarà el punter del que tingui l'element més petit. La unió s'anirà calculant durant l'execució de l'algorisme, sumant el número de vegades que apareix cadascun dels shingles recorreguts. Finalment, igual que en el cas anterior, retornarem la divisió entre la intersecció i la unió obtingudes.

2.2 Aproximació de similitud de Jaccard amb signatures minhash

L'objectiu d'aquest apartat és obtenir una aproximació de la similitud de Jaccard de dos documents smb signatures de minhash. Per fer-ho, prèviament s'ha d'haber filtrat els documents a analitzar amb l'execuable filter.out.

La conversió a signatures de minhash comença fent una conversió a sets de k-shingles. Una vegada fets s'ha de fer una altra conversió per tenir dues signatures de minhash. Amb les dues signatures de minhash, i donat que les dues tenen el mateix tamany degut a les propietats del minhash, es pot procedir a fer el càlcul de la similitud. Aquest cálcul es defineix con la quantitat de ńumeros que són iguals mateixes posicions de les dues signatures de minhash.

2.3 Locality-Sensitive Hashing (LSH)

L'objectiu d'aquest apartat és obtenir tots els parells de documents que tenen una similitud igual o superior a una donada, i fer-ho de forma eficient. Per fer-ho s'ha de convertir tots els fitxers en una matriu de minhash. Llavors s'ha de decidir una b i una r, que són la quantitat de bandes i la quantitat de files per banda que es farán. Per fer-ho s'ha trobat la b que satisfà que:

$$t \approx \frac{1}{b}^{1/r} \tag{2.1}$$

Sapiguent que $b * r \approx n$. Això és aproximat ja que n no sempre és un nombre divisible.

Amb això ja es pot fer l'algorisme descrit en el llibre. Per fer-ho hem decidit implementar-ho amb un map, per a tenir més comoditat a la hora de sapiger si un set ja ha sigut afegit. En acabar, el LSH retorna tots els documents que tenen una similitud igual o superior a la indicada, i se suposa que ho fa més eficientment en temps que mirar tots els posibles parells de documents.

2.4 Com executar els programes

Tots els experiments que es descriuen es poden executar altre cop amb el material adjunt. A continuació explicarem què fa cadascún del executables y com utilitzar-los. Cal tenir en compte que tot està pensat per ser executat en un ordinador amb sistema operatiu Linux, i fer-ho amb un altre pot portar algun problemes.

2.4.1 Què contè cada directori

2.4.2 Què fa cada programa

En el material adjunt està el codi per generar els executables, però els executables. Per generar-los s'han de seguir els següents passos:

1. Dirigir-se amb el terminal al directori **release/src**
2. Executar la comanda **make**
3. Dirigir-se al directori **bin**
4. Aquest directori hauria de contenir tots els executables, del quals s'explicarà el funcionament a continuació.

Els executables que hi ha són:

- filter.out
- permgen.out
- jaccard.out
- lsh.out
- minhash.out

filter.out

Aquest programa serveix per assegurar-se que un document no generarà problemes deguts a caràcters especials. Filtrarà els documents indicats i generarà uns de nous que no donaran problemes. Es crida de la forma:
./filter.out DirPathInput DirPathOutput
DirPathInput indica el directori on són tots els documents que es volen tractar. El programa filtrarà tot el que hi hagi a dins.

DirPathOutput indica el directori on es desitja que es guardin els documents generats.

permgen.out

Aquest programa permet generar permutacions aleatòries d'algun documents. Es crida de la forma: **./permgen.out DirPathIntput DirPathOutput [-n amnt]**
Funciona de manera similar a filter.out. En el directori DirPathOutput es guardaran les permutacions sobre els documents a DirPathInput. Per defecte es generarà una permutació aleatòria per a cada document. Si es desitgen més, és pot fer amb -n seguir de la quantitat de permutacions de cada document desitjaes.

jaccard_pairs.out

Aquest programa llegeix un conjunt de documents i una similitud mínima i indica tots els parells de documents que tenen una similitud igual o superior a la indicada. Aquest càlcul es fa mitjançant la similitud de Jaccard amb k-shingles. Es crida de la forma: **./jaccard_pairs DirPathInput thold k**
DirPathInput conté tots els documents sobre els que es vol fer el càlcul, thold és la similitud mínima i k indica com fer els shingles amb els documents.

jaccard.out

Aquest programa llegeix dos documents i indica la seva similitud de Jaccard. Ho pot fer fer de dues maneres, tant amb conjunts de k-shingles com amb bosses de k-shingles. Es crida de la forma: **jaccard.out**
Després de fer la crida només s'han de seguir les instruccions.

lsh.out

Aquest programa llegeix un conjunt de documents i una similitud mínima i indica tots els parells de documents que tenen una similitud igual o superior a la indicada. Aquest càlcul es fa mitjançant aproximació de similitud de Jaccard amb signatures de minhash. Es crida de la forma

lsh.out FolderPath thold nperm k seed

Els paràmetres volen dir:

- **FolderPath**: és el directori on estan situats tots els documents dels quals es vol saber la similitud. El càlcul es farà amb tot el que hi hagi a dins.
- **thold**: és la similitud mínima que han de tenir els documents retornats.
- **nperm**: és la quantitat de permutacions que es desitja que faci l'algorisme. Quant més gran sigui més precís serà el càlcul, però també serà més costós.
- **k**: indica con serán els shingles que es faran amb els documents. Per a qué tot funcioni correctament, ha de tenir un valor com a mínim tan gran con la quantitat de caracters del document més petit.
- **seed**: llavor que s'utilitza per fer les signatures aleatòries de minhash.

minhash.out

Aquest programa llegeix un conjunt de documents i indica la similitud de tots els parells de documents. Aquesta similitud es calcula amb una aproximació de signatures de minhash. Es crida de la forma. **./minhash.out** Després de fer la crida només s'han de seguir les instruccions.

Capítol 3

Experimentació

3.1 Resultats de la similitud

3.2 Temps de càlcul

Capítol 4

Conclusió

Bibliografia

[1] Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman *Mining of Massive Datasets*