

PLATAFORMA EMOVIE: XARXA SOCIAL DE PEL·LÍCULES

Memòria del projecte d'Enginyeria en
Informàtica realitzat per Roger Llopart
Pla i dirigit per Jordi Gonzàlez i Sabaté.

Bellaterra, 16 de setembre de 2013

El firmant, Jordi González i Sabaté, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha estat realitzada sota la seva direcció per Roger Llopart Pla

Bellaterra, 16 de setembre de 2013

Firmat: Jordi González i Sabaté

Agraïments

Voldria agrair a tothom que m'ha ajudat amb aquest projecte. Especial menció al meu tutor del projecte, Jordi Gonzàlez i Sabaté, per tota l'ajuda que m'ha brindat. També mencionar a les comunitats de Mahout i Movielens, el treball de les quals ha estat inestimable per a la realització d'aquest projecte.

Finalment també agrair l'ajuda a Iván García i la Estefania Gomez, per l'ajuda amb la redacció.

Índex

1	Introducció	1
1.1	Descripció del problema	1
1.2	Motivació	1
1.3	Objectiu	1
1.4	Planificació temporal	2
1.4.1	Anàlisi de l'estat de l'art	2
1.4.2	Desenvolupament de les modificacions als recomanadors	3
1.4.3	Obtenció de resultats dels recomanadors	3
1.5	Estructura de la memòria	3
2	Estat de l'art	5
2.1	Algorismes de recomanació	5
2.1.1	Recomanació de pel·lícules	6
2.2	Importància de la recomanació a la web	6
2.3	Conjunt de proves	7
3	Implementació de la Web	9
3.1	Frontal Web	9
3.2	Zona d'administració	10
3.3	Integració amb el recomanador	11
3.3.1	Servidor	11
3.3.2	Treballador	12
3.3.3	Client	12
3.4	Configuració del servidor	12

4	Recomanadors analitzats	15
4.1	Avaluació del sistema	15
4.2	Implementació dels recomanadors	16
4.3	Recomanador basat en veïnatge	16
4.3.1	Funcionament	16
4.3.2	Adicions al sistema	17
4.4	Slope-One	18
4.4.1	Funcionament	18
4.5	Recomanador basat en la descomposició en vectors singulars . . .	18
4.5.1	Funcionament	20
4.6	Resultats	20
5	Conclusions	23
5.1	Desenvolupament web	23
5.2	Recomanador	23
5.3	Possibles ampliacions	23
5.3.1	Avaluació d'altres paradigmes de recomanació	24
5.3.2	Avaluació d'altres recomanadors col·laboratius	24
	Bibliografia	25

Capítol 1

Introducció

1.1 Descripció del problema

Actualment, a qualsevol pàgina web, hi ha massa informació per a que l'usuari pugui consumir-la tota. En cas de buscar alguna cosa concreta, l'usuari utilitzarà algun sistema de cerca per tal de trobar el que necessita. Però hi ha un gran nombre d'usuaris que no busquen res específic, sinó que busquen informació que els hi resulti rellevant. Per a aquest segon grup d'usuaris, s'utilitzen sistemes recomanadors, que es el que s'ha analitzat en aquest projecte.

1.2 Motivació

La motivació darrera d'aquest projecte es assolir uns majors coneixements sobre els sistemes recomanadors. A més a més, interessa veure quins sistemes són aplicables al món web, el qual requereix d'un grau de dinamisme important.

1.3 Objectiu

L'objectiu d'aquest projecte seria la creació d'una xarxa social amb temàtica orientada a les pel·lícules. El primer objectiu d'aquesta aplicació web seria, doncs, implementar aquest recomanador.

Aleshores es podria separar l'objectiu d'aquest projecte en dues parts:

- Desenvolupament de l'interfície web: S'haurà de desenvolupar tota una interfície web amb la que l'usuari pugui interactuar. Aquesta requerirà d'unes parts que seràn explicades més en detall en un capítol de la memòria final, que serien un sistema per a l'inserció i edició de pel·lícules, un altre sistema per a mostrar la informació sobre la pel·lícula i finalment algun sistema per a avaluar-lo.
- Sistema recomanador: El sistema recomanador, investigant diferents alternatives a utilitzar.

1.4 Planificació temporal

La planificació es pot veure a la taula 1.1. D'alguns punts es realitzarà una petita explicació.

Taula 1.1: Planificació temporal

<i>Tasca</i>	<i>Durada</i>
Anàlisi de l'estat de l'art	3 setmanes
Desenvolupament de modificacions sobre recomanadors trobats	2 setmanes
Devenvolupament de la plataforma web	1 mes
Integració de la plataforma web amb el sistema recomanador	1 setmana
Obtenció de resultats dels recomanadors	2 setmanes
Redacció de la memòria	3 setmanes

1.4.1 Anàlisi de l'estat de l'art

S'ha de buscar informació sobre els diferents sistemes recomanadors que hi ha per tal de realitzar una mica d'exploració abans de començar a mirar implementacions concretes.

1.4.2 Desenvolupament de les modificacions als recomanadors

Tot i que hi ha moltes implementacions de recomanadors, hi ha algunes que no s'adapten exactament al que es vol, per tant s'han de realitzar modificacions.

1.4.3 Obtenció de resultats dels recomanadors

S'han d'obtenir resultats dels diferents recomanadors utilitzats per tal de poder comparar-los de forma objectiva.

1.5 Estructura de la memòria

A l'inici d'aquesta memòria es pot veure l'estat de l'art, que introdueix una mica els diferents conceptes tractats dins d'aquesta. Seguidament venen els dos principals components del projecte i, per tant, de la memòria. Primerament, hi ha l'apartat de l'implementació web, on s'explica el que s'ha dut a terme per tal de realitzar el frontal web. A continuació, s'explica tota la feina que s'ha realitzat respecte als recomanadors. Per a finalitzar, l'últim capítol parla de les conclusions, explicant també possibles camins per seguir amb el treball que s'ha realitzat en aquest projecte.

Capítol 2

Estat de l'art

2.1 Algorismes de recomanació

Els algorismes de recomanació són algorismes, que, com el seu nom indica, són emprats per a recomanar alguna cosa a algun individu. Aquesta definició, per simple que pugui semblar, introdueix dos elements molt importants. Introdueix el concepte de l'element a ser recomanat, d'ara en endavant el *producte*, i el concepte de qui rep la recomanació, d'ara en endavant, l'*usuari*. Des d'aquest punt, apareixeràn aleshores tres mètodes per atacar el problema d'aconseguir una recomanació de qualitat:

- **Recomanadors col·laboratius**

Aquests recomanadors es basen en tenir un conjunt d'*usuaris* que valorin els *productes* i, amb aquesta informació, determinen usuaris similars entre ells. Aleshores recomenarà a l'*usuari* aquells *productes* que hagin agradat als *usuaris semblants* amb ell. El principal problema que tenen aquests recomanadors és que pateixen molt a l'inici del projecte, ja que cal tenir informació de molts *usuaris* i molts *productes* per tal de proveir recomanacions encertades.

- **Recomanadors basats en contingut**

Aquest altre sistema el que fa és categoritzar els *productes*. Aleshores,

quan l'usuari indica quins *productes* li agraden, sap quines son les característiques més importants per a aquest *usuari*.

- **Recomanadors híbrids**

Aquest sistema consisteix en intentar unir els recomanadors col·laboratius amb els recomanadors basats en contingut. És el sistema més complex, donada la gran quantitat de variables que hi participen, però alhora també és el sistema que pot arribar a donar millors resultats, si es ben implementat i parametritzat.

2.1.1 Recomanació de pel·lícules

Existeixen moltes webs que es dediquen a la recomanació de pel·lícules. Per a acotar l'anàlisi de les tècniques que es fan servir reduïrem el conjunt a 10.[12]

Com havíem comentat a la secció anterior, es pot distingir fàcilment entre els *recomanadors col·laboratius* i els *recomanadors basats en contingut*. Exemples del primer conjunt podrien ser, per exemple, Netflix, Movielens, Flixter i Criticker. Per altra banda, formarien part del segon conjunt webs tals com Rotten Tomatoes i Jinni. Finalment, hi ha un tercer conjunt que són els que es basen en rebre com a input una pel·lícula i retornar-ne d'altres. Aquestes webs, com serien IMDb, Clerkdogs o Nanocrowd, molt probablement es basen en un *recomanador basat en contingut* el qual extreu les característiques de la pel·lícula mencionada i en busquen d'altres que tinguin paràmetres similars.

2.2 Importància de la recomanació a la web

La recomanació a la web, que fins fa un temps no era massa comuna, cada cop està guanyant més importància tot i que no en siguem conscients. Un dels grans exemples de recomanació a internet és la publicitat. Un altre gran exemple de recomanacions el trobem als e-commerce, on fins i tot ja hi ha empreses de tercers que es dediquen a implementar els sistemes de recomanació, evitant així al desenvolupador del comerç on-line aquesta feina [2].

Un exemple que és interessant de remarcar és *Netflix*. Aquesta empresa va veure clar que els sistemes de recomanació eren importants, fins al punt que va organitzar un concurs on, els desenvolupadors que aconseguissin trobar el millor algorisme de recomanació, s'emportarien un premi d'un milió de dòlars [7].

2.3 Conjunt de proves

Per a poder provar els diferents recomanadors que s'utilitzaran cal un conjunt de proves. Donat que, encara que es publiqués la web, aconseguir un conjunt de proves suficientment gran és un procés lent, s'ha utilitzat un conjunt de proves bastant comú en el camp de la recomanació, el de Movielens [5].

Movielens ha posat a disposició pública tres conjunts de dades diferents, un de cent mil puntuacions, un de un milió de puntuacions, i un de deu milions. El de deu milions té la limitació de que no han donat informació sobre els usuaris mentre que els altres dos conjunts sí que en contenen.

Capítol 3

Implementació de la Web

S'ha desenvolupat una web molt senzilla per a tal d'utilitzar aquest recomanador. El recomanador, apart de precís, s'ha de veure si és prou ràpid per a poder donar resultats a una velocitat acceptable pel món web.

La web s'ha desenvolupat mitjançant PHP¹ [8], utilitzant MySQL [6] per a la base de dades. El motiu per a triar aquest llenguatge ha estat, simplement, que és el llenguatge amb el que més he treballat.

3.1 Frontal Web

El frontal web té les següents parts:

- Pàgina principal/cercador (Figura 3.1).
- Resultats de cerca, que és un llistat paginat de resultats (Figura 3.2).
- Detalls d'una pel·lícula i la possibilitat de donar-li una puntuació (Figura 3.3).
- Login/Registre d'usuari.

Adicionalment, un cop t'has registrat i connectat amb el teu compte, tens l'opció d'entrar al recomanador. Aquest mostra un llistat, molt semblant al de resultats

¹PHP: Hypertext Preprocessor

de cerca, amb recomanacions per a l'usuari, on les primeres son les que segurament més li interesin, com es pot veure a la figura 3.4.

Figura 3.1: Pàgina principal.

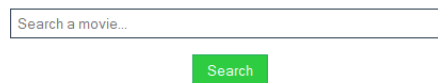
A search bar with the placeholder text "Search a movie..." and a green "Search" button below it.

Figura 3.2: Pàgina dels resultats de cerca.

Results for **Minds**

Dangerous Minds (1995)

No crytics consensus available

Hearts and Minds (1996)

No crytics consensus available

3.2 Zona d'administració

S'ha realitzat un panell d'administració molt senzill per tal de poder:

- Administrar usuaris.
- Administrar pel·lícules.
- Importar pel·lícules desde *Rotten Tomatoes*.

Figura 3.3: Fitxa d'una pel·lícula.

Toy Story 2

Year
1999
Runtime (minutes)
92

Critics Consensus

Toy Story 2 employs inventive storytelling, gorgeous animation, and a top notch voice cast to deliver another rich moviegoing experience for all ages, one that's arguably even better than its predecessor.

Synopsis

No synopsis available

Figura 3.4: Pàgina de recomanació de pel·lícules.

Recommendations for you

Dersu Uzala (1974)

No critics consensus available

Microcosmos (Microcosmos: Le peuple de l'herbe) (1996)

No critics consensus available

Paths of Glory (1957)

No critics consensus available

It Happened One Night (1934)

No critics consensus available

Inherit the Wind (1960)

No critics consensus available

3.3 Integració amb el recomanador

Donat que la pàgina està feta amb PHP i el recomanador en canvi en Java, s'ha hagut de buscar un sistema per a comunicar aquests dos components. El que s'ha acabat utilitzant ha estat Gearman [3], el qual és un sistema compost per tres parts que seràn explicades a continuació.

3.3.1 Servidor

El servidor de Gearman és un procés que s'encarrega de rebre les connexions tant dels *treballadors* com dels *clients* i dirigeix la comunicació entre aquests altres dos elements.

3.3.2 Treballador

El treballador és un procés encarregat de dur a terme una o més tasques anomenades *funcions*. Les funcions reben una cadena de bytes d'entrada i el seu retorn es una altra cadena de bytes. A més a més, tenen la possibilitat d'anar enviant els resultats a mesura que els van obtenint, enlloc d'haver de fer esperar al client a que estigui tot calculat per tal d'enviar la resposta.

Al sistema definit el treballador és el codi en Java que s'encarrega de realitzar les recomanacions. Com a entrada rep únicament un nombre, que es l'identificador de l'usuari, i com a sortida retorna una cadena, codificada en JSON² de les recomanacions sol·licitades.

3.3.3 Client

El client és l'element que crida a funcions del treballador. Pot fer-ho de dues formes: o bé esperant a que el treballador de Gearman retorni el resultat o sinó directament fent la petició i seguint amb el procés. Això permet o bé comunicació amb un procés, o l'execució de tasques en segon plà.

A l'aplicació el client és utilitzat per el codi PHP per a obtenir resultats del recomanador.

3.4 Configuració del servidor

Un cop explicades les eines necessàries per a solucionar els diferents problemes que hi havia a l'hora d'integrar el sistema, cal configurar el servidor, que donat que era per a fer proves, era una màquina virtual per a funcionar amb totes les eines necessàries. Aleshores, les eines utilitzades finalment han estat.

- PHP 5.4 per a l'execució del codi.
- MySQL com a motor de base de dades.
- Apache2 com a servidor HTTP, que deriva les peticions cap a PHP.

²JavaScript Object Notation

- Gearman Server per a poder comunicar-nos amb el recomanador de forma senzilla.
- Java per a executar el recomanador.

A més, per a PHP calien diversos paquets, per tal de realitzar les diferents tasques.

- APC: un sistema d'optimització de PHP. Evita que s'hagi de fer el procés de compilació dels fitxers de codi amb cada petició guardant una còpia del codi compilat a memòria.
- Gearman per tal de poder comunicar-se amb el servidor de Gearman per a fer peticions al recomanador.
- cURL per tal de poder fer peticions a altres webs, en el nostre cas, a la API³ de *Rotten Tomatoes*.

³Application Programming Interface

Capítol 4

Recomanadors analitzats

En aquest apartat s'explicaran les diferents proves que s'han realitzat fins a arribar al recomanador final d'aquest projecte.

4.1 Avaluació del sistema

Per tal de poder apreciar la qualitat del sistema i la millora al realitzar diferents adicions, s'ha fet servir un sistema bastant estàndard en l'avaluació de recomanadors, que és el càlcul del RMSE¹. Aquest és un sistema que calcula de quant s'ha equivocat el sistema en les prediccions de les puntuacions predites a les reals. Això es fa mitjançant l'aplicació de l'equació 4.1.

Fòrmula del RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\text{predicció}_i - \text{valor}_i)^2}{n}} \quad (4.1)$$

Paral·lelament a això, també s'ha analitzat la velocitat del sistema realitzant recomanacions, ja que a part de quan bons són els resultats que es poden obtenir d'un recomanador, també ens interessa saber si pot donar els resultats de forma ràpida.

¹Root Median Square Error

4.2 Implementació dels recomanadors

Els recomanadors explicats a continuació han estat realitzats i avaluats mitjançant el modul Taste, de Mahout [1]. Aquest mòdul conté implementacions de força sistemes de recomanació, per tant el que s'ha hagut de fer principalment és analitzar els diferents possibles paràmetres.

4.3 Recomanador basat en veïnatge

El primer recomanador que s'ha realitzat ha estat un recomanador basat en veïnatge. Aquest recomanador es basa en buscar quan semblants són dos usuaris, per a veure quina importància poden tindre les recomanacions d'un en les de l'altre.

4.3.1 Funcionament

La base del funcionament d'aquest recomanador resulta bastant senzilla. Consisteix en, donades les puntuacions a diferents pel·lícules per part de dos usuaris, calcular quan semblants són. Per a això, es té en compte:

- Nombre de pel·lícules que comparteixen.
- Similitut entre les votacions en les pel·lícules compartides

Aquesta distància es pot calcular amb diferents algorismes. En el nostre cas, hem utilitzat la *distància logarítmica*.

Un cop realitzat aquest procés per a totes les parelles d'usuaris, ja es poden realitzar recomanacions. Per a realitzar una recomanació, el que es fa es una estimació de la nota que posaria l'usuari a les pel·lícules que no ha vist encara. Aquesta estimació es realitzada mirant la puntuació que han donat altres usuaris a aquestes pel·lícules, com es pot veure a l'algorisme 1.

S'ha de realitzar aquest càlcul per a cada pel·lícula i un cop fet, ordenar-les en funció de la puntuació predita per tal de mostrar a l'usuari les pel·lícules que és més probable que l'interesi veure.

Algorithm 1 Algorisme de predicció de la puntuació d'una pel·lícula

```

puntuacio_acumulada ← 0
semblanca_acumulada ← 0
for puntuacio ← pelicula_no_vista.puntuacions do
    un_altre_usuari ← puntuacio.usuari
    semblanca ← calcular_semlanca(usuari, un_altre_usuari)
    puntuacio_acumulada ← puntuacio_acumulada + puntuacio.valor ×
    semblanca
    semblanca_acumulada ← semblanca_acumulada + semblanca
end for
return puntuacio_acumulada/semblanca_acumulada

```

A la mínima que el nombre d'usuaris i pel·lícules creix, aquest càlcul resulta cada cop més pesat per al sistema, fins que resulta impossible realitzar-lo. Les formes de reduir aquest càlcul serien reduir el nombre de pel·lícules a contemplar o reduir el nombre d'usuaris a contemplar. Donat que de les pel·lícules no sabem encara quines li poden agradar i quines no, el més senzill es reduir el nombre d'usuaris a tenir en compte.

Per tant, el que es fa per a reduir el nombre de càlculs a realitzar és reduir el nombre d'usuaris que es faràn servir per a predir les pel·lícules que li poden interessar a un usuari. Això es pot fer tant agafant els N usuaris més semblants com agafant tots els usuaris fins a una determinada semblança. S'ha optat per la primera opció, i s'ha acabat donant un valor prou gran com per a tenir en consideració múltiples tipus d'usuari, 100.

4.3.2 Adicions al sistema

Fins ara s'ha parlat d'utilitzar les pel·lícules que tenen en comú dos usuaris per a determinar quan semblants són. Aquesta mètrica es bona, però es pot arribar a millorar. Per a això, el que es pot fer es tindre en compte dades que es tinguin de l'usuari.

S'ha realitzat una implementació molt simple tenint en compte dades addicionals dels usuaris. Concretament, les dades que s'han tingut en compte han estat les següents:

- Sexe
- Grup d'edat
- Localització

Apart de l'avantatge d'afegir noves variables al sistema, que fa que un cop entrenat correctament pugui ser més precís, afegeix l'avantatge de tindre variables abans de que l'usuari hagi puntuat gaires pel·lícules, facilitant el funcionament del sistema per a usuaris que tenen poques pel·lícules puntuades al sistema.

4.4 Slope-One

Aquest recomanador s'ha introduït a la comparació ja que es un recomanador que, tot i ser semblant a l'anterior, dona molts bons resultats.

4.4.1 Funcionament

Aquest recomanador el que fa és, donada una pel·lícula de la qual volem saber la possible puntuació per a un usuari, mirar la puntuació de pel·lícules que l'usuari comparteixi amb gent que ha vist la pel·lícula en qüestió. Per a això, el que es fa es calcular les diferències entre cadascuna de les pel·lícules. El que es fa a l'entrenament és calcular les diferències entre cada parella de pel·lícules, com es mostra a l'algorisme 2. Un cop calculades les diferències, l'estimació de les puntuacions es realitza amb l'algorisme 3 [11].

4.5 Recomanador basat en la descomposició en vectors singulars

El recomanador basat en la descomposició en vectors singulars es basa en calcular uns vectors de característiques dels usuaris i de les pel·lícules. El que fa llavors no es treballar directament amb els valors de les puntuacions a l'hora de

Algorithm 2 Entrenament del Slope-One

```

usuaris ← usuaris_amb_ambudes_pelicles(pelicula1, pelicula2)
diferencia_puntuacions ← 0
nombre_puntuacions ← comptar(usuaris)
for usuari ← usuaris do
    diferencia_puntuacions ← diferencia_puntuacions +
    obtindre_puntuacio(usuari, pelicula1) − obtindre_puntuacio(usuari, pelicula2)
end for
diferencies[pelicula1][pelicula2] ← diferencia_puntuacions
pes_diferencies[pelicula1][pelicula2] ← diferencia_puntuacions
diferencies[pelicula2][pelicula1] ← −diferencia_puntuacions
pes_diferencies[pelicula2][pelicula1] ← diferencia_puntuacions

```

Algorithm 3 Recomanació del Slope-One

```

puntuacions_acumulades ← 0
pes_puntuacions_acumulat ← 0
for puntuacio ← usuari.puntuacions do
    pelicula_vista ← puntuacio.pelicula
    puntuacions_acumulades ← puntuacions_acumulades +
    (puntuacio.valor + diferencies[pelicula_no_vista][pelicula_vista]) ×
    pes_diferencies[pelicula_no_vista][pelicula_vista]
    pes_puntuacions_acumulat ← pes_puntuacions_acumulat +
    pes_diferencies[pelicula_no_vista][pelicula_vista]
end for
return puntuacions_acumulades/pes_puntuacions_acumulat

```

fer les estimacions, sinó que treballa amb una estructura intermitja, bastant més compacta.

4.5.1 Funcionament

El primer que s'ha de fer es obtenir les matrius que s'han comentat de característiques d'usuaris i característiques de pel·lícules. La llibreria utilitzada ja dú varies implementacions, però únicament n'he estudiada una. El que fa aquesta implementació és, simplement, per a cada característica de cada usuari, calcular la predicció del resultat, i observant l'error, anar estimant un valor cada cop més bo per a la característica. Un cop s'han calculat aquestes matrius, per a fer la predicció d'una pel·lícula no vista per un usuari, el que s'ha de fer és simplement multiplicar el vector de característiques de l'usuari per el vector de característiques de la pel·lícula. Això fa que el nombre d'operacions per a l'estimació de la puntuació d'una pel·lícula sigui molt reduït, a més a més, constant. Sempre s'han de realitzar tantes multiplicacions i sumes com característiques hi hagi.

4.6 Resultats

A la figura 4.1 es pot apreciar que Slope-One i SVD funcionen molt millor que el recomanador basat en veïnatge. Aquests gràfics han estat realitzats executant l'algorisme del RMSE deu vegades per a cada procès. S'ha realitzat una segona execució, de 20 iteracions i més variacions d'aquests 2 algorismes. Els resultats es poden apreciar a la figura 4.2.

Un altre element que s'ha estudiat es la velocitat amb la que els diferents sistemes calculen les recomanacions, ja que és un punt que resulta bastant interessant per a la recomanació web. A la figura 4.3 es pot veure una comparativa dels diferents sistemes. A aquest gràfic es pot apreciar que els càlculs realitzats són molt més ràpids a Slope-One i a SVD. Aquest últim té uns temps de càlcul molt baixos ja que el nombre d'operacions a realitzar per a cada estimació és molt reduït.

Donats aquests resultats, el recomanador que anirà millor per a la web seria el SVD, ja que dona bons resultats i, a més a més, és el sistema més ràpid amb

bastanta diferència.

Figura 4.1: RMSE dels diferents algorismes.

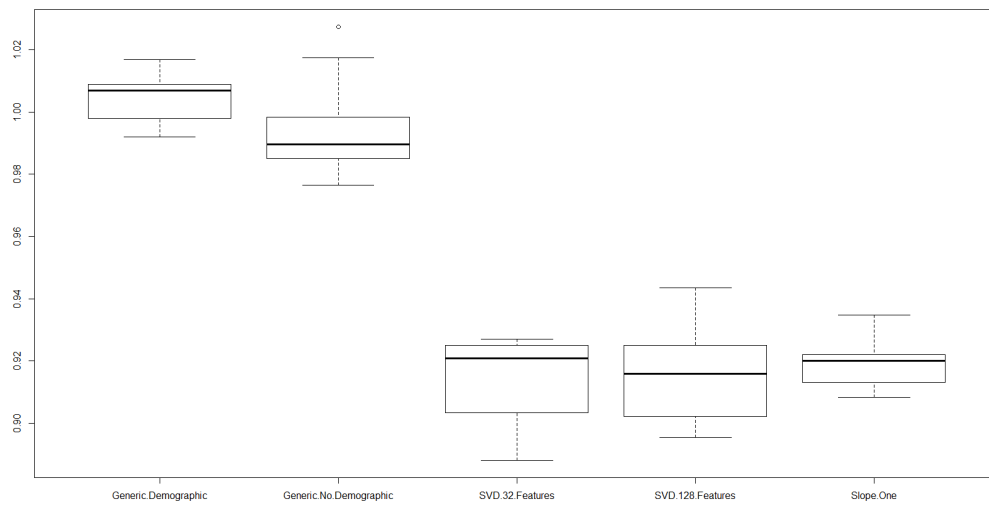


Figura 4.2: RMSE de diferents variants del SVD i Slope-One.

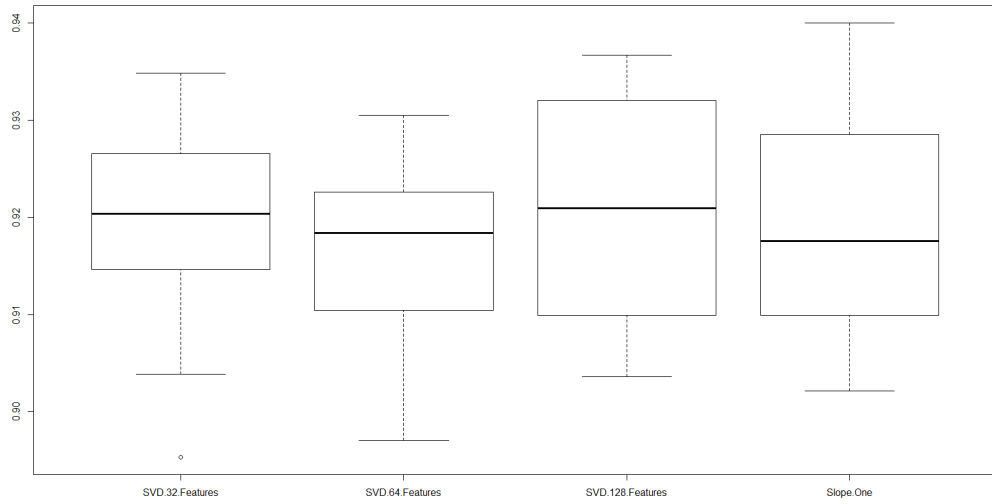
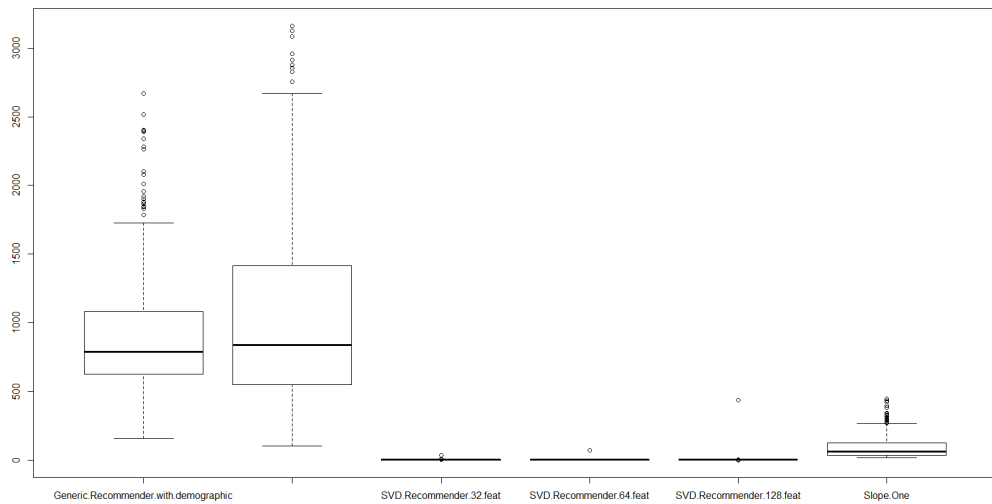


Figura 4.3: Temps tardat en realitzar recomanacions pels diferents sistemes.



Capítol 5

Conclusions

5.1 Desenvolupament web

Sobre el desenvolupament de la pàgina web, s'ha desenvolupat una web que és molt senzilla, que compleix amb l'objectiu, que era donar un frontal per a que l'usuari pugués puntuar pel·lícules, i obtindre recomanacions.

5.2 Recomanador

S'han provat 3 implementacions diferents de recomanadors. S'ha pogut veure que dues d'aquestes (SVD i Slope-One) s'han desmarcat fàcilment del recomanador basat en veïnatge, que es el que s'havia donat al curs, i per tant el que més coneixia i en el que més s'ha treballat.

5.3 Possibles ampliacions

Entre les moltes possibles ampliacions a aquest treball, hi ha dues que mereixen especial menció.

5.3.1 Avaluació d'altres paradigmes de recomanació

Únicament s'ha avaluat un dels paradigmes de la recomanació, que és el filtratge col·laboratiu, per tant encara queda la branca dels recomanadors basats en contingut i dels recomanadors híbrids, que poden donar bastant de joc.

5.3.2 Avaluació d'altres recomanadors col·laboratius

S'han analitzat 3 algorismes recomanadors, els més coneguts, que son el basat en veïnatge, el Slope-One i el SVD. Es podrien analitzar altres, com per exemple recomanadors basats en xarxes neurals [4, 9, 10].

Bibliografia

- [1] *Apache Mahout*. URL: <http://mahout.apache.org/>.
- [2] *Brainsins*. URL: <http://www.brainsins.com/>.
- [3] *gearman [Gearman Job Server]*. URL: <http://gearman.org>.
- [4] *Introduction to Restricted Boltzmann Machines*. URL: <http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/>.
- [5] *MovieLens Datasets*. URL: <http://www.grouplens.org/node/73>.
- [6] *MySQL*. URL: <http://www.mysql.com>.
- [7] *Netflix Prize*. URL: <http://www.netflixprize.com>.
- [8] *PHP: Hypertext Processor*. URL: <http://php.net>.
- [9] Tae Hyup Roh, Kyong Joo Oh i Ingoo Han. “The collaborative filtering recommendation based on SOM cluster-indexing CBR”. A: *Expert Systems with Applications* 25 (2003), 413–423.
- [10] Ruslan Salakhutdinov, Andriy Mnih i Geoffrey Hinton. *Restricted Boltzmann Machines for Collaborative Filtering*. Inf. tèc. 6 King’s College Rd., Toronto, Ontario M5S 3G4, Canada: University of Toronto, 2007.
- [11] *Slope One*. URL: http://en.wikipedia.org/wiki/Slope_One.
- [12] *Top 10 movie recommendation engines*. URL: http://news.cnet.com/8301-17939_109-10200031-2.html.

Firmat: Roger Llopart Pla
Bellaterra, 16 de setembre de 2013

Resum

En aquest treball s'han estudiat diferents sistemes de recomanació, i s'han utilitzat per a veure com podrien arribar a recomanar pel·lícules als usuaris.

Això s'ha realitzat dins del context de la recomanació web, un sistema mitjançant el qual es poden obtenir moltes dades dels usuaris, però que per altra banda requereix de temps de resposta ràpids.

Resumen

En este trabajo se han estudiado varios sistemas de recomendación, y se han utilizado para ver como recomendar películas a los usuarios.

Esto se ha llevado a cabo dentro del contexto de la recomendación web, un sistema mediante el cual se pueden obtener muchos datos de los usuarios, pero que por otro lado requiere de tiempos de respuesta rápidos.

Abstract

In this study we have evaluated multiple recommender systems, and they've been used to recommend films to users.

This has been done in the context of recommendation on the website. This context allows you to obtain a lot of data about the user, but requires quick response times.