Disseny i implementació d'una plataforma de recomanació automàtica

Andreu Blázquez Ávila

Resum– Els sistemes recomanadors són una de les eines més utilitzades en l'actualitat per les grans companyies degut al gran catàleg de productes del que disposen. En aquest projecte destacarem la importància d'aquests sistemes, realitzarem el estudi de l'estat del art dels algoritmes més interessants i analitzarem el comportament d'aquests algoritmes. A més a més, dissenyarem una interfície per utilitzar el sistema i es mostraran alguns dels resultats que podrà veure l'usuari final.

Paraules clau– Aprenentatge computacional, sistemes recomanadors, filtrat col·laboratiu, usuaris i productes semblants, valoracions de productes, metodologia de validació de sistemes recomanadors.

Abstract– Recommendation systems are one of the most used tools by the companies due to the large product catalog that they have. This project will show the importance of these systems, the algorithms used to build them and the results of one implementation. You can also see the interface designed to use system and some of the final results that will be shown to the user.

Keywords– Machine learing, recommender systems, collaborative filtering, product ratings, neighborhood nearest, product ratings, validation metodology for recommender systems.

1 INTRODUCCIÓ

Podem definir un sistema recomanador com un mecanisme que utilitza la informació dels usuaris per oferir productes del seu interès de forma automàtica. Permet realitzar un filtrat entre una gran quantitat de productes que pot ser molt útil per ajudar a l'usuari a prendre una decisió.

Alguns exemples de llocs a Internet on són imprescindibles aquests tipus de sistemes són Amazon, Netflix o Youtube que disposen de catàlegs amb milions de productes.

En el cas de Netflix, la companyia ha assegurat que el 75% dels continguts que es consumeixen al seu servei són fruit de les recomanacions automàtiques que ofereixen als seus usuaris.

Per a un usuari és impossible contemplar totes les opcions quan ha d'escollir un producte entre milers de similars, per aquest motiu, les companyies mencionades anteriorment són el clar reflex de la necessitat d'oferir productes adaptats als gustos de l'usuari. S'han de trobar els productes adequats per a l'usuari i oferir-li aquests productes.

• E-mail de contacte: andreu.blazqueza@e-campus.uab.cat
• Menció realitzada: Computació
• Treball tutoritzat per: Marçal Rossinyol
• Curs 2016/2017

Tant les empreses com els usuaris treuen benefici d'aquests sistemes. Les empreses tenen la possibilitat d'incrementar les seves vendes, vendre productes més variats i entendre millor què volen els usuaris. Per una altra banda, els usuaris troben millors productes degut a que se'ls facilita la navegació i també poden ajudar a altres usuaris deixant les seves opinions.

Existeixen algoritmes d'aprenentatge computacional que ens permeten oferir els productes adequats als usuaris adequats. En aquest projecte ens centrarem en l'estudi i l'anàlisi d'aquests algoritmes per ser capaços de proporcionar bones recomanacions.

Quan es treballa amb aquests tipus de sistemes, la informació principal de la que es disposa és la següent:

- Un conjunt d'elements que són els productes oferts a l'usuari.
- Metadades que descriuen el contingut dels productes.
- Un conjunt d'usuaris consumidors dels productes.
- Un conjunt de puntuacions que els usuaris han deixat als diferents productes per valorar-los segons els seus gustos.

Mitjançant el processament de totes aquestes dades es poden extreure similituds entre els usuaris i els productes que ens permeten realitzar les recomanacions.

2 OBJECTIUS

Els objectius a assolir en aquest projecte són els següents:

- Estudi de l'estat de l'art en algoritmes de recomanació. Decidir quins algoritmes utilitzarem en el nostre projecte.
- Realitzar una implementació dels algoritmes triats en el punt anterior.
- Validar i analitzar els resultats de la implementació:
  - Anàlisi quantitatiu: validar els resultats de forma numèrica. Per això necessitarem una mesura d'avaluació del rendiment.
  - Anàlisi dels efectes dels diferents paràmetres: analitzar com canvien els resultats en funció de variar els paràmetres i comparar les diferències.
  - Anàlisi qualitatiu: anàlisi dels elements recomanats a un usuari segons els seus gustos o preferències.
- Dissenyar una interfície per poder utilitzar fàcilment el sistema.

3 METODOLOGIA

Per assolir els objectius del projecte s'ha dividit el treball en dos apartats clarament diferenciats:

- Desenvolupament de la implementació: implementació i validació dels algoritmes que aprenen y recomanen a través de les dades reals de les que es disposen.
- Disseny de la interfície per a l'usuari: com l'usuari visualitza el sistema i pot introduir les seves dades per rebre recomanacions.

3.1 Desenvolupament de la implementació

Per realitzar aquesta part del projecte utilitzarem el llenguatge de programació Python 2.7.13 amb l'ús d'algunes llibreries utilitzades comunament en problemes d'aprenentatge computacional com scipy, numpy o sklearn i una interfície de desenvolupament anomenada PyCharm Community edition 2017.1.2 adaptada al llenguatge que ens permet executar i depurar fàcilment el codi.

A través de diferents scripts s'implementaran i es validaran les diferents versions dels algoritmes de recomanació.

3.2 Disseny de la interfície

Per a que qualsevol usuari pugui utilitzar el sistema es desenvoluparà una interfície web utilitzant diferents llenguatges i eines. Aquestes eines principalment seran:

- Notepad++ 6.9.2 per desenvolupar el codi de la interfície.
- HTML 5 i CSS 3 amb la llibreria bootstrap 3.3.7 de cara a la part més visual.
- Javascript amb la llibreria jQuery 3.2.1 per fer dinàmic el cercador d'elements (productes).
- PHP 5.6 i AJAX per implementar la funcionalitat de la interfície i fer la crida als scripts Python que ens proporcionen els elements recomanats.
- MySQL 5.7 per les bases de dades y MySQL Workbench 6.3 per la gestió d'aquestes.

4 ESTAT DEL ART

Després de realitzar l'estudi de l'estat de l'art en sistemes recomanadors, hem observat que els algoritmes que millor funcionen i els més utilitzats per implementar-los s'anomenen algoritmes de filtrat col·laboratiu basats en memòria (en anglès: Memory-Based Collaborative Filtering). En aquest projecte ens hem decantat per utilitzar aquest tipus d'algoritmes.

La idea bàsica en els mètodes col·laboratius és trobar la similitud entre usuaris o elements a partir d'una matriu de puntuacions i utilitzar aquesta similitud per predir si ens agradarà o no un element. Això implica que necessitem determinar ja sigui usuaris similars (model basat en usuaris) o elements similars (model basat en elements) per tal de poder fer les prediccions.

La principal diferència entre aquests dos models és la següent:

- Models basats en usuaris (user-based): usuaris semblants puntuen de forma similar el mateixos elements. Si l'usuari A i l'usuari B van puntuar elements d'una forma similar en el passat, es poden utilitzar les puntuacions de l'usuari A per predir puntuacions de l'usuari B respecte elements que aquest no ha puntuat.
- Models basats en elements (item-based): un usuari puntua elements semblants de forma similar, per tant, les puntuacions que l'usuari A ha donat a elements d'una certa categoria poden ser utilitzades per predir puntuacions d'altres elements d'una categoria similar.

A continuació s'explica amb detall com funcionen aquests algoritmes i es mostra un exemple d'aplicació.

4.1 Models basats en usuaris

Aquest model consisteix en identificar els usuaris similars a l'usuari objectiu per al qual es calculen les prediccions de puntuacions. Per tal de trobar els usuaris més semblants a l'usuari objectiu i, es calcula la seva similitud amb la resta d'usuaris de la base de dades. S'ha de definir una funció de similitud adequada pels valors especificats per part dels usuaris.

Aquest càlcul de similitud és complicat pel fet que diferents usuaris poden puntuar elements en escales diferents. Un usuari podria tenir tendència a puntuar alt la majoria dels ítems, mentre que un altre podria fer el contrari.

4.1.1 Formalització de l'algoritme

Disposem d'una matriu de puntuacions m x n que anomenarem R amb m usuaris i n elements, on Iu fa referència al conjunt dels índexs dels elements per als quals l'usuari u ha especificat una puntuació (una fila de la matriu R). Per exemple, si l'usuari (fila) u ha qualificat el primer, tercer i cinquè element (columnes) i no ha donat puntuació a cap

altre element, llavors direm que $lu = \{1, 3, 5\}$. Per tant, el conjunt d'elements qualificat pels usuaris u i v és la intersecció entre $Iu \cap Iv$. Per exemple, si l'usuari v ha qualificat els primers quatre elements $lv = \{1, 2, 3, 4\}$, llavors la intersecció entre $Iu \cap Iv = \{1, 3, 5\} \cap \{1, 2, 3, 4\} = \{1, 3\}$. És possible (i bastant comú) que aquesta intersecció sigui un conjunt buit perquè generalment aquestes matrius són molt disposen de poques puntuacions (matrius escasses). El conjunt $Iu \cap Iv$ defineix les puntuacions observades mútuament, que s'utilitzen per calcular la similitud entre els usuaris u i v.

Un exemple de matriu R podria estar representat per la següent taula:

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$
$u1$	7	6	7	4	5	4
$u2$	6	7	?	4	3	4
$u3$	?	3	3	1	1	?
$u4$	1	2	2	3	3	4
$u5$	1	?	1	2	3	3

Les entrades amb interrogant signifiquen que l'usuari no ha puntuat el element corresponent, per tant, són objectius interessants per realitzar prediccions.

Les funcions que utilitzarem per determinar la similitud $Sim(u, v)$ entre dos vectors de puntuacions seran el coeficient de correlació de Pearson i la similitud del cosinus. Com que $Iu \cap Iv$ representa el conjunt d'índexs dels elements per als que tant l'usuari u com l'usuari v han especificat una puntuació, el coeficient es calcula només entre aquesta sèrie d'elements ignorant la resta.

Com s'ha comentat anteriorment el principal problema de calcular la similitud entre usuaris amb aquest coeficient és que diferents usuaris poden puntuar elements en escales diferents, per tant, les puntuacions han d'estar normalitzades mitjançant la puntuació mitja que l'usuari a donat als elements que ha puntuat. El primer pas consisteix en calcular la puntuació mitja μ_u per a cada usuari u utilitzant les puntuacions especificades:

$$\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|} \ \forall u\{1...m\}$$

La puntuació normalitzada s_{uj} per a l'usuari u element j es defineix com la resta de la puntuació mitja μ_u a tots i cada un dels elements j puntuats per l'usuari u. Anomenarem r_{uj} a aquesta fila d'elements tal que:

$$s_{uj} = r_{uj} - \mu_u \ \forall u\{1...m\}$$

Llavors, el coeficient de correlació de Pearson entre els usuaris u i v es defineix de la següent forma:

$$Pearson(u, v) = \frac{\sum_{k \in I_u \cap I_v} s_{uk} \cdot s_{uv}}{\sqrt{\sum_{k \in I_u \cap I_v} {s_{uk}}^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} {s_{vk}}^2}}$$

Aquest coeficient es calcula entre l'usuari objectiu i tots els altres usuaris per trobar els top-k usuaris similars. En aquest punt es pot definir el grup d'usuaris similars al usuari objectiu com el conjunt d'usuaris k amb el coeficient de Pearson més alt, o es pot definir un llindar (threshold) com a mètrica de similitud on es descarten els usuaris que no arribin a aquest llindar. A causa del fet que quan s'utilitzen dades reals la informació de puntuacions és escassa, en aquest projecte establirem diferents llindars per veure com varien les prediccions en funció d'anar variant aquest valor. Podríem dir que és una millora heurística.

En aquest punt, igual que hem fet anteriorment, s'utilitzen les puntuacions normalitzades dels top-k usuaris similars a l'usuari objectiu per proporcionar una predicció també ponderada. La puntuació mitja de l'usuari objectiu després s'afegeix (es suma) a la predicció anterior per proporcionar una predicció real $\hat{r}_{uj}$ de l'usuari u element j. La notació "^" a la part superior de $\hat{r}_{uj}$ indica que és una puntuació predita a diferència de una que ja es va observar en la matriu original de puntuacions. Direm que $P_u(j)$ és la llista dels k usuaris més similars a l'usuari objectiu u, que han especificat qualificacions per al element j. Per tant, la funció de predicció és la següent:

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} Sim(u, v) \cdot s_{vj}}{\sum_{v \in P_u(j)} |Sim(u, v)|}$$

4.1.2 Exemple d'aplicació

Suposem que disposem de la matriu R representada per la taula que hem vist anteriorment on trobem cinc usuaris (1, 2, 3, 4, 5) i sis elements (1, 2, 3, 4, 5, 6). En aquest cas el rang del les puntuacions va del 1 al 7 on 1 és la més negativa. En aquest exemple el usuari objectiu és el usuari 3 i es vol predir la puntuació dels elements 1 i 6, és a dir, $\hat{r}_{31}$ i $\hat{r}_{36}$.

El primer pas és calcular la puntuació mitja de cada usuari:

$$\mu_1 = \frac{7+6+7+4+5}{6} = 5.5$$
$$\mu_2 = \frac{6+7+4+3+4}{5} = 4.8$$
$$\mu_3 = \frac{3+3+1+1}{4} = 2.5$$
$$\mu_4 = \frac{1+2+2+3+3+4}{6} = 2.5$$
$$\mu_5 = \frac{1+1+2+3+3}{5} = 2$$

El segon pas és normalitzar la matriu amb les puntuacions mitges de cada usuari tal i com hem explicat anteriorment, llavors, obtindríem una matriu R' amb les puntuacions normalitzades de la següent forma:

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$
$u1$	1.5	0.5	1.5	-1.5	-0.5	-1.5
$u2$	1.2	2.2	?	-0.8	-1.8	-0.8
$u3$	?	1	1	-1	-1	?
$u4$	-1.5	-0.5	-0.5	0.5	0.5	1.5
$u5$	-1	?	-1	0	1	1

En tercer lloc hem de calcular la similitud entre l'usuari objectiu (el usuari 3 en aquest cas) i la resta d'usuaris. Podem utilitzar la similitud del cosinus o el coeficient de correlació de Pearson. En aquest cas utilitzarem Pearson. Un exemple entre el usuari 1 i 3 seria el següent:

$$Pearson(1, 3) = \frac{1.5*1+1.5*1+(-1.5)*(-1)+(-0.5)*(-1)}{\sqrt{1.5^2+1.5^2+(-1.5)^2+(-0.5)^2} \cdot \sqrt{1^2+1^2+(-1)^2+(-1)^2}} = 0.894$$

La matriu que representaria les similituds entre l'usuari 3 i tota la resta d'usuaris es pot representar de la següent forma:

	$u1$	$u2$	$u3$	$u4$	$u5$
$u3$	0.894	0.939	1.0	-1.0	-0.817

Si establim un llindar de similitud de 0, és a dir, ens quedem amb tots els usuaris amb coeficients 0 o positius, els usuaris més semblants a 3 serien 1 i 2 amb coeficients 0.894 i 0.939 respectivament. Per últim s'aplica la fórmula de predicció utilitzant les puntuacions dels usuaris més semblants al usuari 3, en aquest cas, els usuaris 1 i 2.

Llavors, per predir $\hat{r}_{31}$ utilitzem r_{11} i r_{21} tal que:

$$\hat{r}_{31} = 2 + \frac{1.5 * 0.894 + 1.2 * 0.939}{0.894 + 0.939} = 3.35$$

I per predir $\hat{r}_{36}$ utilitzem r_{16} i r_{26} tal que:

$$\hat{r}_{36} = 2 + \frac{-1.5 * 0.894 + (-0.8) * 0.939}{0.894 + 0.939} = 0.86$$

Com a conclusió d'aquest petit exemple, podríem dir que a l'usuari 3 li recomanaríem l'element 1 amb més prioritat que l'element 6.

4.1.3 Variant de la funció de predicció

Per tal de realitzar les prediccions, en comptes d'utilitzar les puntuacions directes r_{uj} o les puntuacions centralitzades amb les mitges s_{uj}, es pot emprar el z_{uj} o Z-score, que divideix s_{uj} entre la desviació estàndard σ_u de les puntuacions observades de l'usuari u. La desviació estàndard es defineix de la següent forma:

$$\sigma_u = \sqrt{\frac{\sum_{j \in I_u} s_{uj}}{|Iu| - 1}}$$

Llavors el z-score es calcula tal que:

$$z_{uj} = \frac{r_{uj} - \mu_u}{\sigma_u} = \frac{s_{uj}}{\sigma_u}$$

Si anomenem $P_u(j)$ al conjunt dels top-k usuaris similars al usuari objectiu u que també han puntuat l'element j, la puntuació $\hat{r}_{uj}$ de l'usuari objectiu u per l'element j és defineix de la següent forma:

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} Sim(u, v) \cdot z_{uj}}{\sum_{v \in P_u(j)} |Sim(u, v)|}$$

En aquest cas, per obtenir la predicció final s'ha de multiplicar el resultat de la divisió per la desviació estàndard σ_u de l'usuari objectiu. Això es degut a que s'aplica una funció de normalització durant el càlcul de de la puntuació i per aquest motiu s'ha d'aplicar la seva funció inversa durant el procés del càlcul de la predicció final per obtenir el resultat adequat.

Un dels problemes amb el Z-score és que les votacions predites podrien ser amb freqüència fora del rang permissible, no obstant això, podem realitzar una classificació per preferència per a un usuari en particular.

4.2 Models basats en elements

4.2.1 Formalització de l'algoritme

En els models basats en elements, la diferència principal és que es busca la similitud entre els elements en comptes d'entre els usuaris per realitzar les prediccions.

Per tant, les similituds han de ser calculades entre els elements (columnes de la matriu R de puntuacions).

Abans de calcular les similituds entre les columnes hem de normalitzar cada fila de la matriu R de qualificacions restant la mitja de puntuacions de cada usuari u a cada element j tal que $R' = r_{uj} - \mu_u$ per a tot r_{uj}. Aquest procés és idèntic al descrit anteriorment que dona com a resultat el càlcul de les puntuacions normalitzades s_{uj}. Anomenem U_i els índexs del conjunt d'usuaris U que han puntuat l'element i. Si el primer, tercer, i quart usuaris han puntuat l'element i, llavors tenim que $U_i = \{1, 3, 4\}$.

Llavors, la similitud del cosinus entre els elements (columnes) i i j es defineix de la següent forma:

$$Cosine(i, j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} {s_{ui}}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} {s_{uj}}^2}}$$

En aquest cas també podríem utilitzar el coeficient de correlació de Pearson per fer el càlcul de la similitud entre els elements i i j.

Considerem el cas en què hem de predir la puntuació de l'element t per a l'usuari u. El primer pas és determinar el k-top d'elements més similars a t mitjançant la similitud del cosinus. Anomenarem $Q_t(u)$ als elements més semblants a t que l'usuari u ha puntuat. El valor promig d'aquestes puntuacions sense normalitzar serà el valor predit per a l'element t. El pes de l'element j en aquesta predicció és igual a la similitud del cosinus entre aquest element j i l'element objectiu t. Per tant, la funció de predicció és la següent:

$$\hat{r}_{ut} = \frac{\sum_{j \in Q_t(u)} Sim(j, t) \cdot r_{uj}}{\sum_{j \in Q_t(u)} |Sim(j, t)|}$$

La idea bàsica és aprofitar les puntuacions pròpies de l'usuari en els elements similars a l'element objectu quan anem a fer la predicció. Per exemple, en un sistema de recomanació de pel·lícules, els grups d'elements similars haurien de ser de gèneres similars. El historial de puntuacions del mateix usuari en aquestes pel·lícules és un element molt fiable dels interessos d'aquest usuari.

Els algoritmes basats en usuaris i en elements són molt semblants però en aquest cas no podem aplicar el z-score com a una variant de la funció de predicció perquè cada element es valora amb la puntuació d'usuaris diferents que puntuen en escales diferents. Per tant, en aquest cas la desviació estàndard d'un element no aporta informació, al contrari, estaríem afegint soroll a les prediccions.

4.2.2 Exemple d'aplicació

Considerem la mateixa matriu R que hem vist a l'exemple de l'algoritme basat en usuaris. En aquest cas també anem a predir $\hat{r}_{31}$ i $\hat{r}_{36}$ com hem fet a l'exemple anterior. Com que aquestes puntuacions són les que li falten a l'usuari 3, hem de calcular la similitud entre la columna corresponent a l'element 1 i la resta d'elements i entre la columna corresponent a l'element 6 i la resta d'elements per trobar els elements més semblants a l'1 i al 6 respectivament.

Aquesta similitud s'ha de calcular amb les puntuacions normalitzades havent restat la puntuació mitja de cada usuari a cada un dels ítems que ha puntuat aquest usuari. La matriu resultant R' amb la que es calcularan les similituds

entre els elements és exactament la mateixa que la del exemple basat en usuaris.

Per exemple la similitud entre els elements 1 i 3 basades en el cosinus és la següent:

$$Cosine(1,3) = \frac{1.5*1.5+(-1.5)*(-0.5)+(-1)*(-1)}{\sqrt{1.5^2+(-1.5)^2+1^2}\cdot\sqrt{1.5^2+(-0.5)^2+(-1)^2}} = 0.912$$

La següent taula mostra la similitud entre els elements 1, 6 i la resta d'elements de la matriu:

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$
$i1$	1	0.735	0.912	-0.848	-0.813	-0.990
$i6$	-0.990	-0.622	-0.912	0.829	0.730	1

Si establim un llindar de similitud 0, els elements més semblants a l'1 són el 2 i el 3, i els elements més semblants al 6 són el 4 i el 5, per tant, les puntuacions que l'usuari 3 ha donat als elements 2 i 3 seran utilitzades per predir la puntuació $\hat{r}_{31}$ i les puntuacions que l'usuari 3 ha donat als elements 4 i 5 seran utilitzades per predir la puntuació $\hat{r}_{36}$ tal i com es mostra a continuació:

$$\hat{r}_{31} = \frac{3*0.735+3*0.912}{0.735+0.912} = 3$$

$$\hat{r}_{36} = \frac{1*0.829+1*0.730}{0.829+0.730} = 1$$

Com podem veure en aquest cas també recomanaríem l'element 1 abans que el 6. També es pot observar que les prediccions semblen més consistents ja que no se surten del rang com ha passat a l'exemple basat en usuaris.

4.3 Afegint context a les recomanacions

En els algoritmes vists anteriorment, utilitzem principalment coeficients de similitud basats en les puntuacions donades pels usuaris per trobar relacions entre els usuaris o els elements dels que disposem. En cap moment estem filtrant els elements pel seu contingut.

En les bases de dades reals els usuaris consumeixen productes de continguts molt variats. Això provoca que si nosaltres hem vist i ens han agradat cinc pel·lícules d'animació per a infants, no necessàriament el sistema ens recomanarà pel·lícules d'animació per a infants. Molt probablement usuaris similars a nosaltres que han valorat de forma positiva les mateixes pel·lícules que nosaltres, també han valorat de forma positiva pel·lícules d'altres gèneres com podrien ser, per exemple, les més populars. Aquestes pel·lícules també són candidates a aparèixer en les recomanacions finals i potser no tenen res a veure amb les que ens han agradat a nosaltres.

Per solucionar aquest problema i donar més context a les recomanacions hem utilitzat les metadades dels elements per filtrar els resultats tenint en compte el contingut dels elements que li han agradat als usuaris.

Seguint l'exemple de les pel·lícules les metadades són claus ja que descriuen el contingut (gènere) d'aquestes de forma bastant precisa.

En aquest cas es disposa d'un domini concret de gèneres com podria ser per exemple $D = \{animation, terror, drama, comedy, adventure\}$ que utilitzarem per realitzar el filtrat de la sortida d'algun dels algoritmes vists anteriorment.

Simplement el hem fet és tenir en compte el número de cops que apareixen aquests gèneres en les pel·lícules que li han agradat al usuari objectiu u. Un cop tenim aquesta informació, re ordenem i filtrem la llista dels top-k elements recomanats per al usuari u segons les aparicions d'aquests gèneres en els elements d'aquesta llista.

Per exemple, si l'usuari ha indicat que li agraden cinc pel·lícules i en totes aquestes apareix la etiqueta $animation$ es dona prioritat a les pel·lícules recomanades que també contenen aquesta etiqueta per sobre d'altres.

D'aquesta forma millorem la qualitat de les recomanacions ja que a més a més de tenir en compte la similitud dels usuaris també tenim en compte el contingut dels elements.

5 AVALUACIÓ DEL RENDIMENT

Hi ha diverses formes d'avaluar la qualitat de les recomanacions d'un sistema recomanador però moltes d'elles no s'adapten bé al món acadèmic degut a que no ens permeten avaluar de forma numèrica els resultats de la implementació.

Després d'estudiar com avaluar de forma numèrica la qualitat de les prediccions realitzades pels sistemes recomanadors, hem decidit utilitzar la mètrica anomenada root mean square error (RMSE).

Hem escollit aquesta mètrica perquè és la més utilitzada i ens aporta una informació clara i senzilla de l'error obtingut en les prediccions realitzades pel sistema.

Sigui x un vector de puntuacions reals d'usuaris i $\hat{x}$ un vector de puntuacions predites, aplicarem la següent fórmula:

$$RMSE = \sqrt{\frac{1}{N}\sum (x_i - \hat{x}_i)^2}$$

Per exemple: si un usuari ha puntuat un element amb un 7 i l'algoritme prediu que aquest usuari hauria puntuat aquest element amb un 5 tenim un error de 2 punts en aquesta predicció.

El RMSE s'obté calculant la mitja dels errors obtinguts d'un conjunt de test, és a dir, estem aplicant el càlcul de l'exemple anterior per a tot un conjunt de prediccions i calculant la seva mitja.

Aquesta valor ens indica com de bo o dolent és el nostre algoritme. Quant més baix és, més precís és el nostre recomanador.

6 BASES DE DADES UTILITZADES

Hem utilitzat les següents bases de dades per provar la implementació del sistema:

MovieLens dataset. Aquesta base de dades disposa de:

- 700 usuaris.
- 9000 pel·lícules
- 100.000 puntuacions implícites de 0 a 5 totes múltiples de 0.5.
- Cada usuari ha puntuat al menys 20 pel·lícules.

Book-Crossing dataset. Aquesta base de dades disposa de:

- 4975 usuaris.
- 137110 llibres.
- 268142 puntuacions implícites de 1 a 10 totes múltiples de 1 (números sencers).
- Cada usuari al menys ha puntuat 15 llibres.

7 Resultats obtinguts

En aquest apartat es mostra com varia el RMSE en funció de cada base de dades utilitzada, del algoritme utilitzat i dels paràmetres que s'han anat variant en les diferents les execucions. Els detalls es resumeixen a continuació:

- Tipus d'algoritme: Basat en usuaris o basat en elements.
- Mètrica de similitud: coeficient de correlació de Pearson o similitud del cosinus.
- Llindars de similitud: rang de -1 a 1 amb salts de 0.1.
- Diferents funcions de predicció: només amb la normalització de les puntuacions amb la mitja de cada usuari o incloent la desviació estàndard (z-score) de cada usuari. Aquest últim punt només s'ha realitzat als models basats en usuaris.

Per cada base de dades i experiment realitzat s'han predit 1000 puntuacions d'usuaris a elements.

7.1 MovieLens dataset

7.1.1 Model basat en usuaris

La figura 1 mostra els resultats obtinguts pel model basat en usuaris:

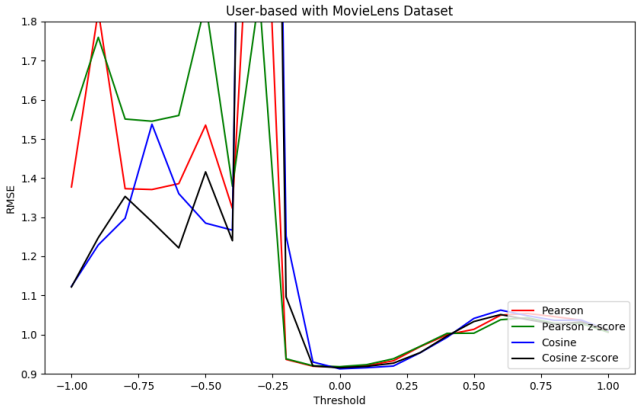

Fig. 1: User-based with MovieLens dataset

El RMSE mínim per aquesta base de dades (on recordem que les puntuacions anaven de 0 a 5) és de 0.912 utilitzant la mètrica de similitud del cosinus i un llindar de similitud de 0. Com ja havíem comentat anteriorment, utilitzem aquest llindar per descartar els usuaris amb un coeficient de similitud inferior al valor del llindar establert. Com es pot apreciar a la figura 1, el RMSE comença a estabilitzar-se quan el llindar és major de -0.1 i els millors resultats es donen amb un llindar de 0 en totes les variants.

També es pot apreciar que el RMSE és completament inestable amb llindars més petits de -0.2 i no té una clara tendència a estabilitzar-se en valors d'entre 0.9 i 1.1 fins arribar a un llindar superior o igual a -0.1. Això és provocat pel fet que els usuaris amb similitud inferior a -0.1 no ens aporten informació de l'usuari objectiu i si no els descartem estem afegint soroll. Aquest soroll es el causant de la inestabilitat als resultats.

Si establim un llindar 0 o superior els resultats continuen sent estables però el RMSE té tendència a pujar pel que podem dir que és el valor òptim per aquesta base de dades.

7.1.2 Model basat en ítems

La figura 2 mostra els resultats obtinguts per l'algoritme item-based:

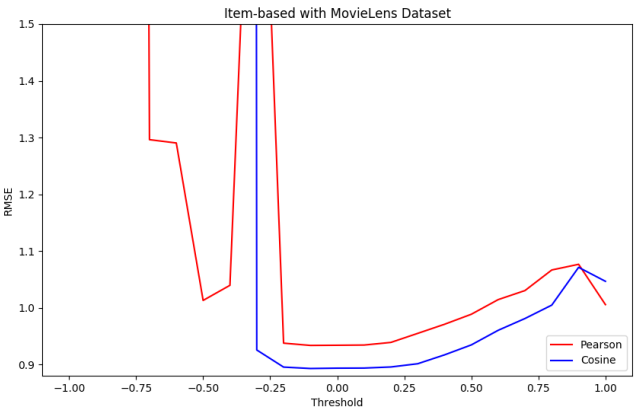

Fig. 2: Item-based with MovieLens dataset

El RMSE mínim és de 0.89 amb la mètrica de similitud del cosinus i amb un llindar de 0. Aquest valor és lleugerament inferior al 0.91 obtingut en el algoritme user-based, pel que podríem dir que funciona una mica millor però sense haver-hi grans diferencies entre tots dos.

També podem observar que en aquest cas la similitud del cosinus funciona clarament millor que el coeficient de correlació de Pearson com a mètrica de similitud. El RMSE sempre és més baixos pel que escolliríem aquesta funció de similitud pel nostre recomanador basat en elements.

Com en el cas anterior, el RMSE s'estabilitza a partir d'utilitzar un llindar de -0.2 aproximadament això vol dir que els elements amb una similitud inferior a -0.2 no ens aporten informació. Entre -0.2 i 0.2 trobem els millors llindars de similitud, fora d'aquest rang el RMSE comença a pujar constantment.

Un altre conclusió que podem extreure és que utilitzant llindars superiors a 0.2 sembla que comencem a ser massa restrictius a l'hora de descartar usuaris. Descartem usuaris que realment si que ens aporten informació útil dels usuaris objectius i això provoca que el RMSE pugi constantment.

7.2 Book-Crossing dataset

7.2.1 Model basat en usuaris

La figura 3 mostra els resultats obtinguts per l'algoritme user-based:

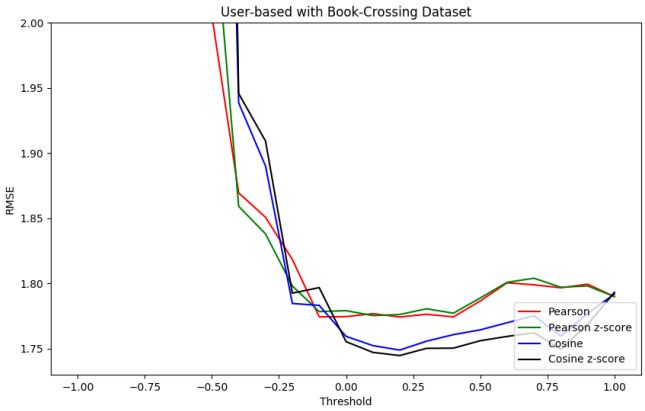

Fig. 3: User-based with Book-Crossing dataset

En aquesta base de dades on el rang de puntuacions va de 1 a 10 podem veure que el RMSE és més alt degut a que el domini de possibles puntuacions inclou valors més alts i això provoca que el RMSE oscil·li en valors superiors. El que no ha variat gaire és el comportament vist anteriorment amb el mateix model i la base de dades de MovieLens.

El RMSE cau constantment fins que s'arriba a un llindar de aproximadament 0 on trobem els valors més baixos i, per tant, els millors. A partir d'aquí els valors del RMSE comencen a pujar lleugerament.

El que si que crida l'atenció en aquest cas i que crec que és un fet a destacar és que amb aquesta base de dades on el domini de possibles puntuacions és més gran, si s'inclou la desviació estàndard a l'hora de realitzar les prediccions els resultats milloren. Dit amb altres paraules, si tenim en compte la forma en la que cada usuari puntua els elements, els resultats són encara millors.

Això es degut al fet que usuaris diferents puntuen de forma diferent com hem explicat anteriorment i, uns poden tenir tendència a puntuar tot molt alt i d'altres a puntuar tot molt baix. En aquest casos on el domini de possibles puntuacions és més gran podem afirmar que incloure la desviació estàndard quan realitzem les prediccions és una bona idea.

Per últim podem observar que el RMSE mínim és de 1.74 utilitzant la similitud del cosinus amb un llindar de 0.2 i incloent el z-score en la fórmula de predicció.

Fins ara, en totes les proves que hem realitzat hem pogut observar que la similitud del cosinus funcionat millor. En aquest experiment la diferència ha estat en la forma de realitzar les prediccions.

7.2.2 Model basat en ítems

La figura 4 mostra els resultats obtingts per l'algoritme item-based:

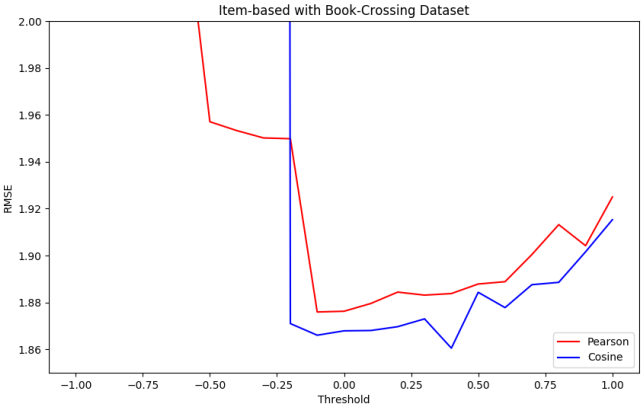

Fig. 4: Item-based with Book-Crossing dataset

En aquesta última prova el coeficient de similitud del cosinus segueix sent la millor mètrica de similitud pel que podem dir que definitivament funciona millor que Pearson. El RMSE més baix en aquest cas és de 1,86 amb un llindar de 0.4. És el llindar òptim més alt que hem vist fins ara.

El comportament del RMSE és gairebé idèntic al experiment basat en elements realitzat anteriorment. Llindars per sota de -0.2 no son gens bons, introduïm massa soroll, i llindars per sobre de 0.5 són massa restrictius encara que podrien ser vàlids ja que el RMSE no es dispara.

En aquesta base de dades el model basat en usuaris ha funcionat millor que el basat en elements. Això es justament el contrari que ha passat en la base de dades anterior, però les diferencies no són molt grans en cap cas.

El que si que podria variar segons si utilitzem un model o un altre són les recomanacions finals, pel que segons quin tipus d'elements recomanem ens convé triar un algoritme o un altre.

8 Interfície d'usuari

De cara a la interfície final hem utilitzat la base de dades de MovieLens perquè ens ha semblat que era més visual treballar amb pel·lícules a l'hora de avaluar qualitativament les recomanacions.

El funcionament de la interfície es resumeix en el diagrama de seqüencia de la figura 5.

A continuació mostrem un possible exemple d'interacció entre el usuari i la interfície per veure amb més detall el procès que l'usuari segueix fins a rebre les recomanacions.

En primer lloc, quan l'usuari accedeix a la interfície, es troba la vista de la figura 6 on pot interactuar amb el cercador de pel·lícules que li apareix.

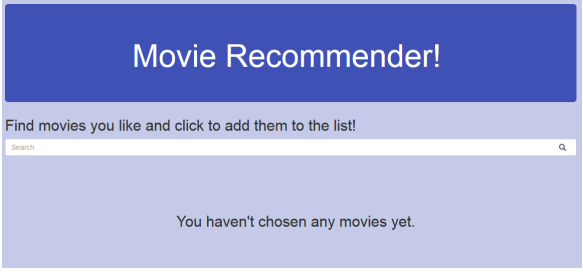

Fig. 6: Vista inicial

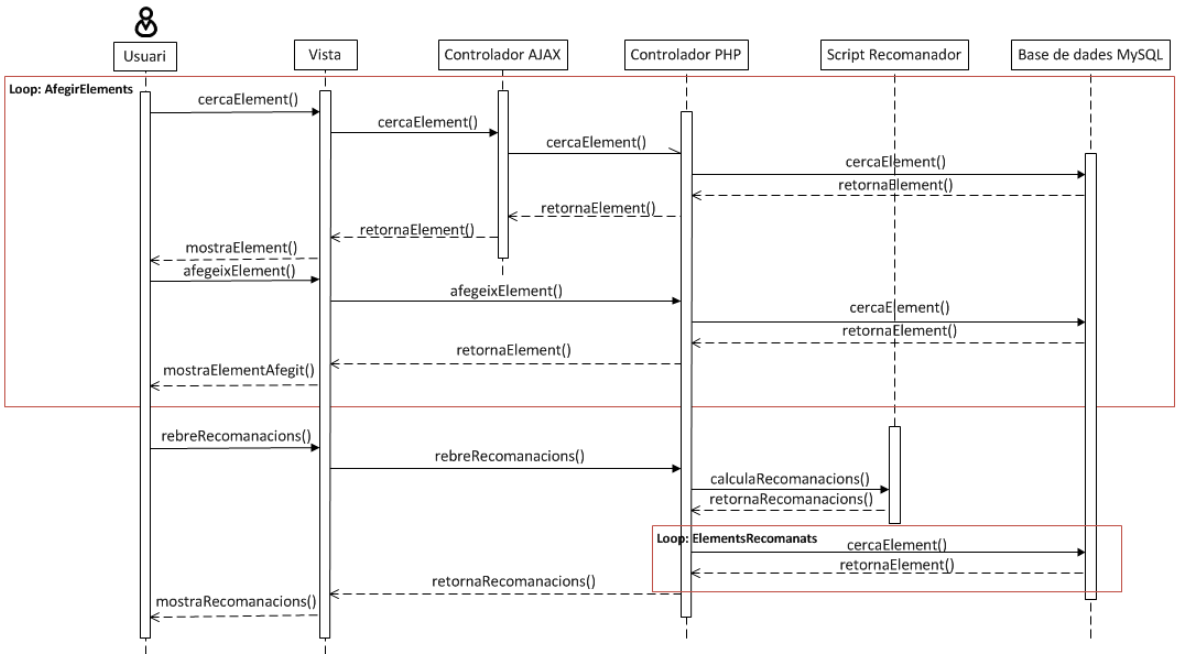

Fig. 5: Diagrama de seqüència de la interfície.

Quan l'usuari comença a introduir caràcters al cercador per buscar una pel·lícula en funció del nom del títol, es mostren de forma dinàmica les coincidències trobades a la base de dades tal i com es pot veure figura 7.

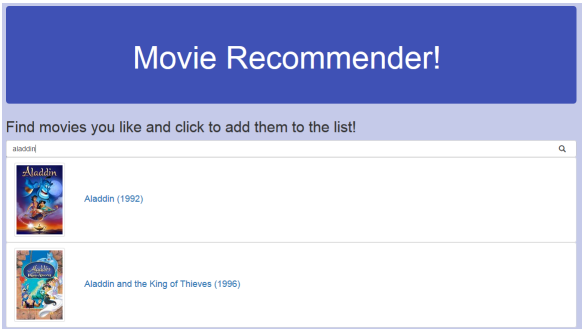

Fig. 7: Usuari cercant pel·lícules

Si l'usuari fa clic sobre alguna de les pel·lícules que apareixen, aquesta s'afegeix a la llista de pel·lícules que li agraden tal i com podem veure a la figura 8. En aquest cas hem afegit la pel·lícula "Aladdin" a la llista.

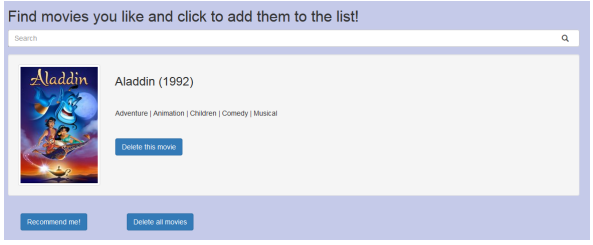

Fig. 8: Llista de pel·lícules seleccionades

Un cop que l'usuari ha seleccionat totes les pel·lícules que desitja, pot fer clic sobre el botó "Recommend me!" on es processa la entrada escollida i, per últim, es mostren els resultats de l'algoritme de recomanació. En la figura 9 podem veure que les dos primeres pel·lícules que es recomanen a l'usuari són "Shrek 2" i "Mulan".

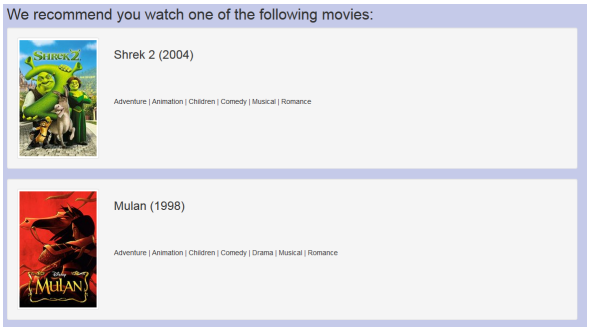

Fig. 9: Pel·lícules recomanades pel sistema

9 AVALUACIÓ QUALITATIVA

En aquest apartat visualitzarem alguns dels resultats que ens mostra el recomanador escollint diferents entrades i analitzarem si són adequats tenint en compte el contingut de la entrada triada.

En primer lloc, com entrada hem triat cinc pel·lícules d'animació (dibuixos animats), concretament:

movieid	title	generes
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
364	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical|IMAX
588	Aladdin (1992)	Adventure|Animation|Children|Comedy|Musical
595	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
2081	Little Mermaid, The (1989)	Animation|Children|Comedy|Musical|Romance

Fig. 10: Pel·lícules triades com a entrada

El top 10 de pel·lícules recomanades ha estat el següent:

movieid	title	generes
1566	Hercules (1997)	Adventure|Animation|Children|Comedy|Musical
1907	Mulan (1998)	Adventure|Animation|Children|Comedy|Drama|Musical|Romance
3034	Robin Hood (1973)	Adventure|Animation|Children|Comedy|Musical
4306	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Romance
8360	Shrek 2 (2004)	Adventure|Animation|Children|Comedy|Musical|Romance
26093	Wonderful World of the Brothers Grimm, The (1962)	Adventure|Animation|Children|Comedy|Drama|Fantasy|Musical|...
56152	Enchanted (2007)	Adventure|Animation|Children|Comedy|Fantasy|Musical|Romance
73854	Rudolph, the Red-Nosed Reindeer (1964)	Adventure|Animation|Children|Fantasy|Musical
78499	Toy Story 3 (2010)	Adventure|Animation|Children|Comedy|Fantasy|IMAX
106696	Frozen (2013)	Adventure|Animation|Comedy|Fantasy|Musical|Romance

Fig. 11: Sortida de l'algoritme

En aquesta primera prova els resultats semblen molt bons. Hem triat cinc pel·lícules molt semblants d'animació i els resultats han estat molt similars a la entrada triada. Als resultats podem veure pel·lícules com "Hercules", "Shrek" o "Mulan" que són pel·lícules d'un estil molt similar. Inclús podem trobar "Toy Story 3" què és una de les pel·lícules de la mateixa saga que "Toy Story" la qual estava inclosa a la entrada triada.

A continuació realitzarem una segona prova on només introduirem dues pel·lícules també d'un estil bastant similar (acció i drama) per veure com varien els resultats quan tenim menys informació a la entrada:

movieid	title	generes
110	Braveheart (1995)	Action|Drama|War
3578	Gladiator (2000)	Action|Adventure|Drama

Fig. 12: Pel·lícules triades com a entrada

El nou top 10 de pel·lícules recomanades és el següent:

movieid	title	generes
1262	Great Escape, The (1963)	Action|Adventure|Drama|War
2890	Three Kings (1999)	Action|Adventure|Comedy|Drama|War
3654	Guns of Navarone, The (1961)	Action|Adventure|Drama|War
4339	Von Ryan's Express (1965)	Action|Adventure|Drama|War
4826	Big Red One, The (1980)	Action|Adventure|Drama|War
7143	Last Samurai, The (2003)	Action|Adventure|Drama|War
7307	Flesh & Blood (1985)	Action|Adventure|Drama|War
49530	Blood Diamond (2006)	Action|Adventure|Crime|Drama|Thriller|War
52319	Inglorious Bastards (Quel maledetto treno blindato) (1978)	Action|Adventure|Drama|War
61026	Red Cliff (Chi bi) (2008)	Action|Adventure|Drama|War

Fig. 13: Sortida de l'algoritme

Com en el cas anterior també podem donar per molt bons els resultats. Podem veure pel·lícules com "The last samurai" o "Blood Diamond" que continuen sent pel·lícules d'estils molt similars i molt populars.

La qüestió és: ¿Què passa quan escollim pel·lícules de continguts molt diferents? A continuació farem la prova. En aquest cas hem triat cinc pel·lícules de contingut completament diferent com a entrada:

movieid	title	generes
110	Braveheart (1995)	Action|Drama|War
2324	Life Is Beautiful (La Vita è bella) (1997)	Comedy|Drama|Romance|War
6942	Love Actually (2003)	Comedy|Drama|Romance
8360	Shrek 2 (2004)	Adventure|Animation|Children|Comedy|Crime|Drama
136020	Spectre (2015)	Action|Adventure|Crime

Fig. 14: Pel·lícules triades com a entrada

Les pel·lícules recomanades són les següents:

movieid	title	generes
356	Forrest Gump (1994)	Comedy|Drama|Romance|War
970	Beat the Devil (1953)	Adventure|Comedy|Crime|Drama|Romance
1907	Mulan (1998)	Adventure|Animation|Children|Comedy|Drama|Musical|Romance
2890	Three Kings (1999)	Action|Adventure|Comedy|Drama|War
4719	Osmosis Jones (2001)	Action|Animation|Comedy|Crime|Drama|Romance|Thriller
6990	The Great Train Robbery (1978)	Action|Adventure|Comedy|Crime|Drama
7835	Sons of the Thin Man (1947)	Comedy|Crime|Drama|Musical|Mystery|Romance
26093	Wonderful World of the Brothers Grimm, The (1962)	Adventure|Animation|Children|Comedy|Drama|Fantasy|Musical|...
48032	Tiger and the Snow, The (La tigre e la neve) (2005)	Comedy|Drama|Romance|War
69644	Ice Age: Dawn of the Dinosaurs (2009)	Action|Adventure|Animation|Children|Comedy|Romance

Fig. 15: Sortida de l'algoritme

El recomanador ens ofereix contingut molt variat depenent principalment dels gustos dels usuaris de la base de dades. Podem veure que hi ha pel·lícules d'animació com "Ice Age", pel·lícules d'acció com "Three kings" i una comèdia dramàtica com "Forest Gump". La varietat dels continguts dels elements recomanats depèn molt de la entrada.

Si la entrada conté gèneres molt variats els resultats seran pel·lícules de gèneres molt variats i si cerquem pel·lícules de continguts molt similars les recomanacions ofereixen continguts molt semblants com ha passat a les dues primeres proves. En els casos on el contingut és molt variat, els resultats depenen molt de la popularitat i les qualificacions proporcionades pels usuaris de la base de dades.

10 Conclusions

Després de tot el estudi, la implementació realitzada i l'anàlisi dels resultats podem dir que els algoritmes de filtrat col·laboratiu basats en memòria combinats amb heurísitques de filtrat per contingut proporcionen recomanacions automàtiques molt vàlides que poden ser de gran utilitat en la presa de decisions de l'usuari.

Si a un usuari se li fa difícil la navegació en un lloc que disposa de catàlegs de milers de productes, el més probable és que busqui alternatives on li facilitin la navegació. En aquest punt els resultats que hem vist en aquest projecte són claus. Probablement molts de nosaltres consumiríem pel·lícules que hem vist recomanades en aquestes petites proves que hem realitzat.

Com son algoritmes d'intel·ligència col·lectiva que aprenen directament de les dades de les que es disposa, la quantitat i qualitat de les dades influeix directament en els resultats que s'obtenen.

Si no disposem de suficients dades els resultats poden no ser els esperats o desitjats pel que s'haurien de buscar altres tècniques com filtratge directe basat en contingut.

Com hem comentat a la introducció, per llocs com Netflix que disposen de milions d'usuaris i la majoria del continguts que es consumeixen estan directament relacionats amb les recomanacions automàtiques que realitzen, és imprescindible comptar amb aquest tipus d'algoritmes de filtrat col·laboratiu. Aporten un valor afegit que té influencia directa en els beneficis que obtenen.

Agraïments

Vull agrair al meu tutor Marçal Rossinyol l'ajuda i el constant seguiment del projecte al llarg d'aquests mesos.

Als meus companys Marc Leiva i Aitor Acosta que han estat els meus companys de viatge durant aquests quatre bonics anys.

A la meva família, als meus amics, i a tota la gent que ha estat al meu costat recolzant-me en aquesta etapa de la meva vida tant en els bons com en els pitjors moments.

Gràcies.

Referències

[1] Charu C. Aggarwal. *Recommender Systems*. Number 978-3-319-29657-9. Springer International Publishing, 2016.

[2] William W. Cohen. Collaborative filtering: A tutorial. Carnegie Mellon University, Maig 2004. https://www.cs.cmu.edu/∼wcohen/collab-filtering-tutorial.ppt.

[3] SciPy developers. Numpy and scipy documentation, 2017. https://docs.scipy.org/doc/.

[4] Sklearn developers. Scikit-learn documentation, 2017. http://scikit-learn.org/stable/.

[5] Jordi Gonzàlez. Anàlisi d'errors. Universitat autònoma de barcelona, 2016.

[6] Jordi Gonzàlez. Sistemes recomanadors. Universitat autònoma de barcelona, 2016.

[7] Andrés González. Sistemas de recomendación de contenido con machine learning. Cleverdata, Septembre 2014. http://cleverdata.io/sistemas-recomendacion-machine-learning/.

[8] GroupLens. Movielens dataset. Universitat de Minnesota, Octubre 2017. https://grouplens.org/datasets/movielens/.

[9] Aarshay Jain. Quick guide to build a recommendation engine in python. Analytics Vidhya, Juny 2016. https://www.analyticsvidhya.com/blog/2016/06/quick-guide-build-recommendation-engine-python/.

[10] Agnes Johannsdottir. Implementing your own recommender systems in python. Cambridges Park. https://cambridgespark.com/content/tutorials/implementing-your-own-recommender-systems-in-Python/index.html.

[11] Vincy. Jquery ajax autocomplete example. PHP pot, November 2014. http://phppot.com/jquery/jquery-ajax-autocomplete-country-example/.

[12] Cai-Nicolas Ziegler. Book-crossing dataset. Institut für Informatik, Septembre 2004. http://www2.informatik.uni-freiburg.de/∼cziegler/BX/.

[13] Tomáš Řehořek. Evaluating recommender systems. Recombee blog, Desembre 2016. https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35.