UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base Corso di Laurea Magistrale in Informatica

Progetto di Intelligent Web

Studio ed analisi di un semantic-based recommended system

Studenti

Gabriele Merola N97000368

Ivan Capasso N97000381

Anno Accademico

2022/2023

Indice

1. Problema e requisiti

E' stato richiesto di realizzare una recommender system (**RS**), che prende in in input una URM (User Rating Matrix) e mediante una tecnica di collaborative filtering produce delle stime di ranking da confrontare con un test set.

Il RS in oggetto non deve essere uno di quelli già studiati durante il corso (most popular,top ranked, discounted similarity function,popular item correction).
La progettazione deve essere delegata a studi scientifici, provenienti da pubblicazioni su rivista classificata di fascia Q1 o Q2 su Scimago (https://www.scimagojr.com/).
Il dataset da utilizzare per l'addestramento del RS in oggetto è Movielens [4]
L'analisi sperimentale prevede il calcolo dell'accuratezza basato sul Mean absolute error (**MAE**) rispetto ad un test set, deve riportare il tempo di esecuzione della generazione delle stime dei ranking e la scelta dei valori per gli iperparametri deve essere motivata sulla base dei lavori scientifici di riferimento o da un'analisi sperimentale.

2. Soluzione individuata

La tecnica User-Based Neighborhood è una tecnica di collaborative filtering (**CF**) basata sulla somiglianza degli utenti per effettuare raccomandazioni personalizzate. In questa tecnica, gli utenti vengono raggruppati in base alle loro preferenze, e si cerca di consigliare elementi che siano piaciuti a utenti con gusti simili.

Tra le varie soluzioni User-Based si è scelto di prendere in analisi un'interessante tecnica denominata Semantic based collaborative filtering (**SemCF**) [1]. Gli autori della suddetta tecnica utilizzano le informazioni semantiche del dominio per migliorare l'accuratezza della CF classica, gestendo il problema del cold start e della sparsità.

Per la fase di testing è stato utilizzato MovieLens[4], un web-based recommended system che consiglia film, basandosi sulle loro preferenze cinematografiche utilizzando tecniche di CF sulle valutazioni e recensioni sui film degli altri membri.
Per i nostri scopi, era sufficiente solo il dataset che mette a disposizione la piattaforma, in particolare è stata utilizzata la versione *small* più recente.

Utilizzando la versione *small* del dataset di Movielens, si è scelto di utilizzare come caratteristiche semantiche unicamente i generi dei film.

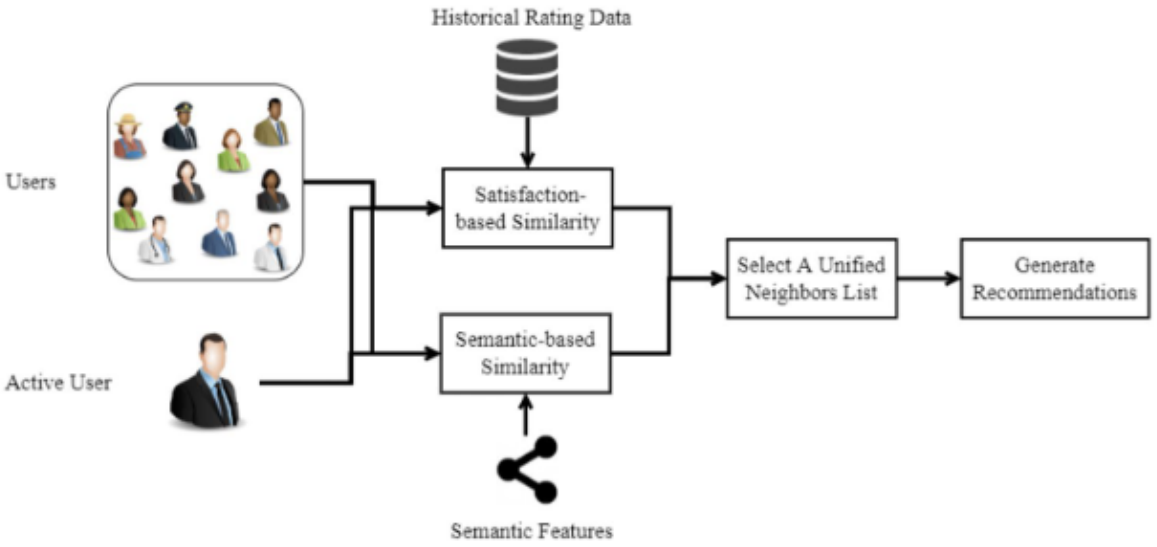
Il RS che utilizza la SemCF è stato, quindi, implementato, analizzato e testato in linguaggio Python (v3.9).
E' possibile consultare il codice su Github: https://github.com/Noctino52/SemCF

3. Semantic-based Collaborative Filtering

La tecnica SemCF in oggetto tenta di predire le valutazioni di un utente a dei film ancora non valutati da quest'ultimo tramite due tipi di similarità con altri utenti:

- Similarità con altri utenti basata sulle valutazioni
- Similarità con altri utenti basata su caratteristiche semantiche (generi dei film)

Successivamente SemCF prevede la creazione di una lista unica di utenti che corrisponde all'intersezione tra il gruppo di utenti più simili per valutazioni e il gruppo di utenti più simili per caratteristiche semantiche.
Sulla base della similarità con gli utenti contenuti in questa lista, Il SemCF RS predice i voti ai film che l'utente non ha ancora valutato.



In sostanza la SemCF è suddivisa in quattro fasi:

1. **Semantic-based Similarity**: generazione di un valore di similarità basato sui dati semantici delle features degli item. (**SemSim**)

2. **Satisfaction-based Similarity**: generazione di un valore di similarità basato sulle valutazioni che danno gli utenti agli item. (**PreSim**)

3. **Selezione dei vicini**: Dai due passi precedenti, ottengo per ciascun utente due liste di utenti simili (a seconda del tipo di similarità). Queste due liste vengono poi incrociate per ottenere un'unica lista. (**UNL**)

4. **Generazione delle raccomandazioni**: Conoscendo i valori di similarità tra due utenti secondo SemSim e PreSim, viene calcolato un valore di similarità unico, e questo viene a sua volta utilizzato per calcolare un valore di predizione, basato sui top **NumNibors** utenti presenti nella UNL.

3.1. Semantic-based Similarity

Partendo dalla URM, per ciascun utente (Au) vengono generate due partizioni dell'insieme dei film:
-LikeSet: Contiene tutti gli item che Au ha recensito con un voto superiore ad α
-DislikeSet: Contiene tutti gli item che Au ha recensito con un voto inferiore ad α

α è uno dei due iper-parametri che caratterizzano la tecnica SemCF e rappresenta la soglia di gradimento di un utente nei confronti di un film.

Successivamente per ogni coppia di utente (au,u) viene calcolato un valore di similarità in base ai rispettivi LikeSet e Dislikese. Questi valori sono ottenibili calcolando una similarità tra gli item, usando la metrica di similarità binaria di Jaccard:

$$SemSimI(V_p, V_q) = \frac{F_{11}}{F_{10} + F_{01} + F_{11}}$$

Dove F11 è il numero totale di caratteristiche semantiche in comune tra i due item. F10 è il numero totale di caratteristiche in cui p rientra, ma non q. F01 è il numero totale di caratteristiche in cui q rientra, ma non p. Questa metrica può essere utilizzata per conoscere se due item, di due utenti diversi, sono simili tra di loro.

$$SemSim^{+}(AU, u) = \frac{\sum_{i^{AU} \in LikeSet^{AU}} \sum_{i^{u} \in LikeSet^{u}} SemSimI(i^{AU}, i^{u})}{|LikeSet^{AU}| * |LikeSet^{u}|}$$

$$SemSim^{-}(AU, u) = \frac{\sum_{i^{AU} \in DisLikeSet^{AU}} \sum_{i^{u} \in DisLikeSet^{u}} SemSimI(i^{AU}, i^{u})}{|DisLikeSet^{AU}| * |DisLikeSet^{u}|}$$

Per dare lo stesso peso alle similarità dei Likeset e dei Dislikeset si calcola un valore unico, risultato della media tra i due valori precedenti.

$$SemSim(AU, u) = \frac{SemSim^{+}(AU, u) + SemSim^{-}(AU, u)}{2}$$

Dopo questo passaggio si è ottenuto il primo valore di similarità tra due utenti, che corrisponde alla similarità basata sulle caratteristiche semantiche (**SemSim**).

3.2. Satisfaction-based Similarity

La funzione di similarità basata sulle valutazioni viene calcolata a partire da due funzioni: La correlazione di Pearson[3] e la similarità binaria di Jaccard[2] basata sugli utenti.

Fissati due utenti, au e u, la correlazione di Pearson è:

$$Pearson(AU, u) = \frac{\sum_{i \in I_{AU,u}} (r_i^{AU} - \bar{r}_{AU})(r_i^{u} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_i^{AU} - \bar{r}_{AU})^2} \sqrt{\sum_{i \in I} (r_i^{u} - \bar{r}_u)^2}}$$

Dove:

- $\mathbf{I_{au,u}}$ sono gli item in comune tra Au e u;
- $\mathbf{r_i^{Au}}$ è la valutazione dell'utente Au sull'item i;
- $\mathbf{r_i^{u}}$ è la valutazione dell'utente u sull'item i;
- $\bar{r}_{\mathbf{Au}}$ è la valutazione media dell'utente Au;
- $\bar{r}_{\mathbf{u}}$ è la valutazione media dell'utente u.

La correlazione di Pearson è una misura di similarità utilizzata nelle tecniche CF che considera solo gli elementi comuni tra gli utenti [3].
Però il valore di similarità può essere inaccurato quando gli utenti hanno pochi elementi in comune. Se, ad esempio, due utenti A e B hanno votato rispettivamente 20 e 150 film e B ha votato gli stessi 20 film di A con le stesse valutazioni, allora la correlazione di Pearson fornirà un valore di similarità del 100%, ignorando i 130 film valutati da B, ma non da A.

Proprio per ovviare a questo problema, viene calcolata la similarità binaria di Jaccard basata sugli utenti.

$$Jaccard(AU, u) = \frac{|V_{AU} \cap V_u|}{|V_{AU} \cup V_u|}$$

Dove:

- $\mathbf{V_{Au}}$ è l'insieme degli item recensiti dall'utente Au
- $\mathbf{V_u}$ è l'insieme degli item recensiti dall'utente u

SemCF calcola,quindi, la similarità basata sulle valutazioni (**PreSim**), moltiplicando la correlazione di Pearson con la similarità di Jaccard. Quest'ultima svolge un ruolo di normalizzatore, per ottenere un valore di similarità più affidabile.

$$PreSim(AU, u) = Pearson(AU, u) * Jaccard(AU, u)$$

3.3. Selezione dei vicini

In questa fase SemCF crea per ogni utente **Au** due liste composte dai primi NumNibors utenti **u** con i valori di somiglianza più elevati:

- Semantic Neighborhood List (**SNL**), lista che contiene i primi NumNibors con il valore di SemSim(Au,u) più elevato; SNL include utenti con interessi simili a quelli dell'AU, ma con un metodo di valutazione non necessariamente simile;
- PreSim Neighborhood List (**PNL**), lista che contiene i primi NumNibors utenti con il valore di PreSim(Au,u) più elevato; PNL include utenti con un metodo di valutazione quanto più simile ad Au, ma con interessi non necessariamente simili.

A questo punto SemCF per ogni utente incrocia SNL e PNL, ottenendo una Unified Neighborhood List (**UNL**).
La UNL di un utente Au contiene gli utenti che condividono interessi simili ad Au e che hanno un metro di valutazione simile.

$$UNL = |PNL \cap SNL|$$

Se per un utente Au l'intersezione di questi due insiemi è vuota, vengono selezionati i primi NumNibors/2 utenti della PNL e della SNL, per poi unirli.

NumNibors è il secondo e ultimo iper-parametro e rappresenta il numero di utenti da cui deve dipendere la predizione della valutazione di un utente Au per un dato film.

3.4. Generazione della raccomandazione

A questo punto della SemCF per ciascun utente Au si ha una UNL e per ogni coppia (Au, u) si hanno due valori di similarità:

- PreSim(Au,u)
- SemSim(Au,u)

Banalmente, per ottenere una stima di similitudine unica per ciascuna coppia (Au,u) e per dare lo stesso peso ad entrambi i valori di similitudine, si effettua una media.

$$Sim(AU, u) = \frac{PreSim(AU, u) + SemSim(AU, u)}{2}$$

SemCF, infine, utilizza la UNL per predire la valutazione dell'utente Au per un film i :

$$p^{i}_{AU} = \bar{r}_{AU} + \frac{\sum_{u \in U}[Sim(AU,u) * (r^{i}_{u} - \bar{r}_{u})]}{\sum_{u \in U} Sim(AU,u)}$$

Dove:
- **U** è l'insieme degli utenti della UNL dell'utente Au;
- $\bar{r}_{u}$ è la media dei voti dell'utente u;
- $\bar{r}_{Au}$ è la media dei voti dell'utente Au.

4. Analisi del Recommended System

Per effettuare l'analisi delle prestazioni è stato necessario dividere il dataset dei film in due parti:

- Training set, insieme di 7000 film del dataset Movielens con annesse valutazioni su cui addestrare il modello;
- Test set, insieme di 2741 film del dataset Movielens con annesse valutazioni su cui il modello addestrato viene testato.

Per scegliere il modello migliore, invece, si è scelto di dividere il training set in due ulteriori parti:

- Training set, insieme di film su cui addestrare i vari modelli in base agli iperparametri
- Validation set, insieme di film su cui valutare i modelli costruiti.

L'analisi delle prestazioni del sistema si è basata su due principali fasi:

- Fase di Model Building in cui sono stati costruiti i vari modelli del sistema in base ai diversi valori degli iperparametri;
- Fase di testing con cui vengono testate le prestazioni del miglior modello selezionato.

Le prestazioni vengono calcolate in base al MAE (Mean Absolute Error).

4.1. Model building e Validation

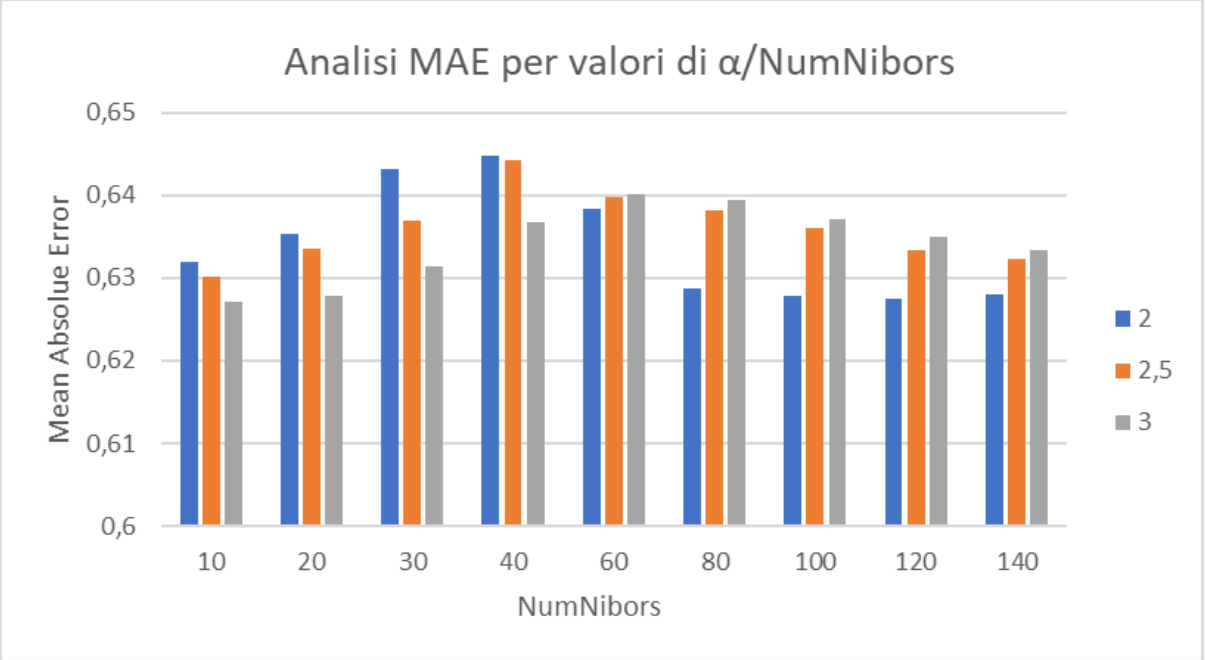
In questa fase sono stati valutati vari modelli del sistema in base ai diversi valori degli iperparametri, che, come citato precedentemente, sono:

- α, valore che rappresenta la soglia di gradimento di un utente nei confronti di un film e che può assumere tre valori: 2, 2.5, 3;
- **NumNibors**, valore che rappresenta il numero di utenti da cui deve dipendere la predizione della valutazione di un utente Au per un dato film.

Il model-building set è stato composto da 5000 film del Training set con annesse valutazioni. Il Validation set, invece, è stato composto con i restanti 2000 film del training set.
I modelli che sono stati costruiti sono riassunti in questa tabella:

α	NumNibors	MAE	Tempo d'esecuzione
2	10	0,631856011	1 hr 7 min 30 sec
2	20	0,635319522	1 hr 6 min 31 sec
2	30	0,643233851	1 hr 7 min 35 sec
2	40	0,644843237	1 hr 6 min 49 sec
2	60	0,638284288	1 hr 12 min 24 sec
2	80	0,628657428	1 hr 13 min 33 sec
2	100	0,627885325	1 hr 7 min 46 sec
2	120	0,62751737	1 hr 12 min 33 sec
2	140	0,627945246	1 hr 17 min 27 sec
2.5	10	0,630147176	1 hr 3 min 5 sec
2.5	20	0,633590281	1 hr 2 min 43 sec
2.5	30	0,636870243	1 hr 2 min 46 sec
2.5	40	0,644244614	1 hr 4 min 19 sec
2.5	60	0,639719381	0 hrs 59 min 29 sec
2.5	80	0,638096729	0 hrs 59 min 36 sec
2.5	100	0,635984166	1 hr 0 min 6 sec
2.5	120	0,633447525	1 hr 7 min 30 sec
2.5	140	0,63236149	1 hr 4 min 9 sec
3	10	0,627048705	0 hrs 56 min 48 sec
3	20	0,627898923	0 hrs 55 min 27 sec
3	30	0,631459021	0 hrs 55 min 57 sec
3	40	0,636706607	1 hr 5 min 51 sec
3	60	0,640081988	0 hrs 56 min 38 sec
3	80	0,639395588	0 hrs 58 min 44 sec
3	100	0,637166214	1 hr 0 min 15 sec
3	120	0,635046864	1 hr 2 min 28 sec
3	140	0,633307736	1 hr 1 min 35 sec

Da questa tabella si evince come il modello migliore sia quello con α=3 e NumNibors=10.



Da questo grafico, invece, si può notare come il MAE tende ad aumentare per valori di NumNibors da 10 a 40, ma da un valore vicino a 40 il MAE tende a diminuire fino ad assestarsi intorno ad un certo valore.

4.2. Testing e risultati

Trovato il modello migliore nella fase precedente, lo si è applicato al test set, ottenendo il seguente risultato:

[**Time**= 1 hr 29 min 23 sec] [α= 3] [**NumNibors** = 10] **MAE** = 0.6491054008619211

Il risultato sembra essere in linea con quelli dello studio da cui è stata presa in considerazione SemCF.
I risultati dellla sperimentazione effettuata in questo progetto differiscono da quelli dello studio “fonte” per i valori del MAE (che nella ricerca si attesta intorno allo 0.8). Questo è spiegabile dall’utilizzo di una sola caratteristica semantica, mentre nella ricerca originaria ne vengono utilizzate ben 4.
Un ulteriore differenza è che in questo progetto si è trattato α come un iper-parametro e quindi i vari modelli sono stati addestrati anche in base a questo valore.

5. Riferimenti bibliografici

[1] Alhijawi, B., Obeid, N., Awajan, A. *et al.* New hybrid semantic-based collaborative filtering recommender systems. *Int. j. inf. tecnol.* **14**, 3449–3455 (2022) (Scimago Q2). https://doi.org/10.1007/s41870-022-01011-x
[2]Sujoy Bag, Sri Krishna Kumar, Manoj Kumar Tiwari An efficient recommendation generation using relevant Jaccard similarity,Information Sciences,Volume 483,2019, Pages 53-64, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2019.01.023. (Scimago Q1)
[3] Benesty, J., Chen, J., Huang, Y., Cohen, I. (2009). Pearson Correlation Coefficient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol 2. Springer, Berlin, Heidelberg. (Scimago Q2) https://doi.org/10.1007/978-3-642-00296-0_5
[4] https://doi.org/10.1145/2827872 (MovieLens)