RECOMMENDER SYSTEM IMPLEMENTATO CON COLLABORATIVE FILTERING

Cecilia Peccolo matr. 1220253

May 2022

Obiettivi L'obiettivo del progetto è stimare la valutazione che ogni utente darebbe a dei film non che non ha visto in base a che voto hanno dato gli altri utenti. Grazie a queste stime, sarà poi possibile restituire in output a un utente specifico, una lista di 30 film consigliati tra quelli non visionati, ordinati in modo decrescente per valutazione stimata.

1 Il dataset

Il dataset scelto contiene 25 milioni di valutazioni date da 162000 utenti a 62000 film nel sito *"grouplens.org"*. Il dataframe è scaricabile direttamente dalla pagina web `https://grouplens.org/datasets/movielens/100k/`. Viste le considerevoli dimensioni del dataset, si è scelto di prenderne solo una parte per ridurre tempi di elaborazione impiegate dal calcolatore.
Per selezionare un dataset ridotto si è scelto di selezionare anzitutto i primi 5000 utenti e tra questi, escludere coloro che hanno dato meno di 50 valutazioni. Inoltre, si sono esclusi anche i film che avessero meno di 500 voti. In questo modo si è semplificato il processo di elaborazione dei dati dal punto di vista di tempi di calcolo, ma l'algoritmo sviluppato è applicabile anche a dataset di dimensioni maggiori. Inoltre, mantenere utenti e film che abbiano un numero di valutazioni superiore alla soglia determinata permette di ottenere delle stime più accurate e diversificate.
Inizialmente troviamo i dati in formato lungo, quindi una riga per ogni valutazione, ma per sviluppare l'algoritmo si è creata una matrice: questa contiene nelle righe gli utenti e nelle colonne i film; i valori variano su scala continua tra 1 e 5, e sono le valutazioni date da un utente a un film: dove non è stato espresso un giudizio è presente un *Null*, sostituito in secondo luogo da uno 0. La matrice dei dati quindi è una matrice con 1584 righe e 263 colonne. Il tasso di sparsità è circa del 63% quindi bisognerà stimare **152273** valori, che sono le valutazioni che ciascun utente non ha dato a un determinato film.

1.1 Dataset di test

Per valutare l'efficacia dell'algoritmo nel prevedere delle stime, si è definito un dataset di *test* in modo da evitare problemi di *overfitting*.
Anzitutto, si è creata una matrice di zeri delle stesse dimensioni di quella di *train*. Successivamente sono stati selezionati in modo casuale 38068 valori della matrice iniziale, pari al 25% delle valutazioni presenti inizialmente; questi sono stati spostati nella matrice di test con le stesse coordinate che avevano nella matrice iniziale, e successivamente sono stati sostituiti in quest'ultima da uno 0, ovvero verranno considerati come valutazioni mancanti.
Dato che l'obiettivo è stimare i valori della matrice di train dove è presente uno 0, i dati selezionati verranno stimati come quelle valutazioni che in partenza non erano presenti nel dataset. Dunque sarà più facile effettuare un confronto tra le stime e i veri valori per valutare l'efficienza dell'algoritmo.
Tale calcolo verrà effettuato tramite la misurazione dell' **Errore quadratico medio**:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{1}$$

dove $\hat{y}_i$ è la valutazione stimata.

1.2 Strutture dati

Data l'alta percentuale di valori mancanti nelle matrici dei dati, per le celle in cui era presente una valutazione, si è utilizzata la struttura dati del **dizionario**: la chiave è una tupla composta dall'indice dell'utente e quella del film, mentre il valore è la valutazione annessa alla relativa chiave: in questo modo risulta più veloce accedere direttamente alle celle in cui non sono presenti degli 0. Questa struttura è risultata molto conveniente soprattutto per sorteggiare i valori casuali da spostare nella matrice di test o, più in generale, per lavorare con i ratings avendo però le coordinate a cui è riferito. In aggiunta al dizionario, è stata creata una classe per i ratings: ognuno di questi è definito da valore, ID dello user che lo ha espresso e titolo del film a cui è riferito. Inoltre, all'interno della classe sono stati definiti due metodi: il primo, dati i valori delle valutazioni, crea un'array, usato all'inizio del programma per passare dal dataset iniziale alla matrice sparsa di train. Il secondo invece ordina il dizionario dei valori stimati e, per uno specifico utente dato in input da tastiera, restituisce la lista di titoli di film raccomandati, ovvero quei film corrispondenti ai 30 punteggi stimati più alti.

2 Algoritmo

2.1 Collaborative filtering user-based

Questa tecnica consiste nello stimare il rating che un utente potrebbe dare a un film facendo delle operazioni sulle valutazioni date da un gruppo di utenti "simili".

film	Abyss	Firm	Fugitive	Indiana Jones	One Flew Over the Cuckoo's Nest
utente 23	4	4	5	5	5
utente 1283	4	5	5	5	4

Table 1: Confronto tra utente 23 e 1283

film	Austin Powers 1	Austin Powers 2	Casablanca	E.T.	Forrest Gump
utente 23	5	3	4	5	5
utente 159	4	4	3	3.5	3

Table 2: Confronto tra utente 23 e 159

2.1.1 Somiglianza tra utenti

Per calcolare quanto due utenti sono simili, si è usato il **metodo del coseno**: questa tecnica calcola il coseno dell'angolo fra due vettori tramite la seguente formula:

$$d(x, y) = \cos(\theta_{x,y}) = \frac{x'y}{\sqrt{x'x}\sqrt{y'y}} \tag{2}$$

Questo calcolo viene effettutato per ogni coppia di righe della matrice di train iniziale, dove ogni riga corrisponde alle valutazioni disponibili per ogni utente. In questo modo si hanno valori più prossimi a 1 per utenti che hanno dato valutazioni simili agli stessi film, mentre più prossime a 0 per utenti poco somiglianti. Per esempio se consideriamo l'utente con ID 23, vediamo che ha somiglianza pari a 0.49 con l'utente 1283: infatti se prendiamo, ad esempio, i primi 5 film in cui entrambi hanno dato una valutazione vediamo che i valori risultano simili, come mostrato nella tabella 1

Invece, se effettuiamo il confronto fra utente 23 e quello con ID pari a 159, vediamo che la distanza del coseno è solo pari a 0.14. Infatti, come vediamo nella tabella 2 i due utenti hanno espresso delle valutazioni che si differenziano maggiormente rispetto al caso precedente.

Le distanze fra utenti vengono collocate in una matrice u x u (dove u è il numero totale di utenti), in cui ogni cella c_{ij} corrisponde la misura di somiglianza tra l'utente u_i e l'utente u_j. Nella diagonale (dove la misura di somiglianza dovrebbe essere 1 poichè sarebbe tra un utente e sè stesso) sono stati posizionati degli 0.

2.1.2 Stima dei valori mancanti

Una volta ottenuti i gradi di somiglianza fra gli utenti, si passa alla stima dei valori mancanti nella matrice di train: dunque, per ogni cella c_{ij}, in cui i è l'indice dello user e j è l'indice del film, si seleziona la i-esima riga della matrice delle similarità e la j-esima colonna nella matrice iniziale e si calcola una media ponderata. In questo modo i ratings dati dagli utenti che hanno visto il j-esimo film vengono pesati in base a quanto somigliano al i-esimo utente.

Il calcolo dell'errore quadratico medio viene effettuato come in espressione 1 tra i valori selezionati nella matrice di test e i corrispondenti stimati nella matrice di train. L'errore medio è pari a, circa, 0.803 .

2.2 Collaborative filtering item-based

Il modo in cui si è scelto di implementare il sistema di raccomandazione è solo uno degli innumerevoli possibili: un'alternativa facilmente paragonabile al procedimento utilizzato, è quello che prevede l'uso del collaborative filtering item-based. In questo caso le somiglianze non vengono calcolate fra utenti ma fra film: di conseguenza il metodo del coseno viene applicato fra i vettori film, quindi le colonne della matrice di partenza; saranno somiglianti film che hanno ottenuto valutazioni simili dallo stesso utente.
Applicando ai dati a disposizione questa tecnica, si è ottenuta una misura di errore pari a circa 0.77, quindi inferiore a quella calcolata sul procedimento user-based. Tuttavia, si è scelto di mantenere quest'ultimo perchè si ottenevano stime più differenziate fra loro: calcolando la somiglianza fra items invece che fra users, le stime ottenute si aggiravano tutte intorno al 4, in un range di pochi decimi di unità. Questo consente di abbassare l'errore quadratico medio, dato che le valutazioni dal 3 al 4 sono le più popolari, ma porta a stime finali meno precise per il singolo utente. Dunque si è scelto di aumentare leggermente la varianza delle stime a discapito di un leggero aumento della distorsione.

3 Conclusioni

In generale l'algoritmo stima abbastanza bene i valori: infatti se consideriamo la tabella 3, in cui viene mostrato il confronto fra 10 dei valori presenti nella matrice di test e le loro corrispondenti stime, si può notare che i valori si discostano di poco. Tuttavia, si può notare che i valori stimati rimangono in un intervallo più ristretto rispetto ai valori iniziali: questo comporta una cattiva previsione sui film che hanno rating 1, 2 o 5, mentre i valori centrali (che sono anche la maggioranza dei dati) sono stimati quasi perfettamente. Il bilanciamento tra queste differenza più o meno ampie, porta ad avere la misura d'errore medio molto bassa.

Nel caso delle valutazioni basse, come quella con chiave (38,129), è difficile avere stime accurate se il film in generale ha ricevuto buone valutazioni dal resto degli utenti: infatti, la media non ponderata delle valutazioni al 128esimo film ("Indiana Jones and the Temple of Doom") è circa 3.73, quindi molto vicina alla stima ponderata ottenuta. Questo comporta che alcuni film vengano raccomandati a degli utenti che non l'hanno guardato, in quanto molto popolari e con valutazioni in generale molto alte. Possiamo notare questo aspetto dai due es-

chiave	valori stimati	veri valori
(0,95)	4.01	4.5
(4,11)	4.07	3
(17,91)	4.2	5
(20,107)	3.41	3.5
(38,129)	3.7	1
(97,145)	3.21	2.5
(309,60)	3.59	3
(627,183)	4.03	4
(1572,187)	3.65	3.5

Table 3: Confronto tra veri valori e valori stimati

empi di output per l'utente con ID 2 e quello con ID 1083 nelle tabelle 4 e 5. I due utenti hanno grado di similarità pari a 0.08, quindi i film raccomandati dovrebbero essere diversi per i due utenti. Tuttavia, troviamo alcuni titoli in comune come "Green Mile" o "Godfather", due film molto popolari con medie di valutazioni pari, rispettivamente a 4.01 e 4.27, il che supporta la tesi esposta precedentemente.

Altre possibili implementazini vedono un modo diverso di calcolare la somiglianza fra users o items: infatti, il metodo del coseno risente dell'elevato tasso di sparsità della matrice, con la conseguenza di stime di somiglianza sempre inferiori a una certa soglia (in questo caso vediamo che non superano mai, circa, 0.6). Ad esempio, si può far uso del metodo di discesa del gradiente o altre implementazioni più complesse.

Utente 2
'Sleepless in Seattle (1993)'
'Sin City (2005)'
'Green Mile, The (1999)'
'Fight Club (1999)'
'Matrix Revolutions, The (2003)'
'Léon: The Professional (a.k.a. The Professional) (Léon) (1994)'
'Big Lebowski, The (1998)'
'Godfather: Part II, The (1974)'
'Fish Called Wanda, A (1988)'
'Fargo (1996)'
'Blair Witch Project, The (1999)'
'Goodfellas (1990)'
'Babe (1995)'
'Seven (a.k.a. Se7en) (1995)'
'Batman Begins (2005)'
'GoldenEye (1995)'
'Star Trek: Generations (1994)'
'Psycho (1960)'
'Batman (1989)'
'Eternal Sunshine of the Spotless Mind (2004)'
'Toy Story 2 (1999)'
'Happy Gilmore (1996)'
'Interview with the Vampire: The Vampire Chronicles (1994)'
'E.T. the Extra-Terrestrial (1982)'
'L.A. Confidential (1997)', 'Jumanji (1995)'
'Memento (2000)'
'Casablanca (1942)'
'Santa Clause, The (1994)'

Table 4: Lista dei primi 30 film raccomandati per l'utente con ID 2

Utente 1083
'Green Mile, The (1999)'
'Avatar (2009)'
'Gattaca (1997)'
'Shrek 2 (2004)'
'E.T. the Extra-Terrestrial (1982)'
'Matrix Revolutions, The (2003)'
'Batman Begins (2005)'
'Mask, The (1994)'
'Babe (1995)'
'Heat (1995)'
'Life Is Beautiful (La Vita è bella) (1997)'
'Godfather: Part II, The (1974)'
'Beautiful Mind, A (2001)'
'Batman (1989)'
'Inglourious Basterds (2009)'
'Nutty Professor, The (1996)'
'Sin City (2005)'
'Alien (1979)'
'Lord of the Rings: The Return of the King, The (2003)'
'I, Robot (2004)'
'Jerry Maguire (1996)'
'L.A. Confidential (1997)'
'Broken Arrow (1996)'
'Erin Brockovich (2000)'
'North by Northwest (1959)'
'Happy Gilmore (1996)'
'Fish Called Wanda, A (1988)'
'Django Unchained (2012)'
'V for Vendetta (2006)'

Table 5: Lista dei primi 30 film raccomandati per l'utente con ID 1083