

# Laboratorio Sperimentale di Matematica Computazionale (Parte I) Lezione 7

Gianna Del Corso <gianna.delcorso@unipi.it>

22 Marzo 2017

Si consideri un insieme di  $N$  utenti  $U = \{u_1 u_2 \dots, u_N\}$  e un insieme di  $M$  items  $I = \{i_1, \dots, i_M\}$ . Sia  $V = \{1, 2, \dots, v\}$  l'insieme dei possibili voti che un utente può assegnare ad un item. Definiamo  $V_0 = V \cup \{?\}$  l'insieme dei possibili voti più il valore  $?$  che corrisponde alla mancata valutazione di un item da parte di un utente. Sia  $A \in V_0^{n \times m}$  la *matrice utilità*, e assumiamo di osservare solo un sottoinsieme di valori  $\Omega$ , dove ogni valore  $a_{ui} \in V$  con  $(u, i) \in \Omega$  rappresenta il voto (o rating) che l'utente  $u$  ha assegnato all'item  $i$ .

Lo scopo di un sistema di raccomandazione è quello di predire i valori sconosciuti nella matrice di utilità per fare delle raccomandazioni personalizzate, cioè stimare i valori di  $A$  per  $(u, i) \notin \Omega$ .

Denotiamo con  $\mathcal{P}_\Omega(\cdot)$  il proiettore tale che

$$\mathcal{P}_\Omega(M)_{ij} = \begin{cases} m_{ij} & \text{se } (i, j) \in \Omega \\ 0 & \text{altrimenti.} \end{cases}$$

e sia  $A_0 = \mathcal{P}_\Omega(A)$ .

## 1 Dati

I dati sono contenuti nei files `u1training` e `u1test` che contengono rispettivamente 80.000 e 20.000 ratings su una scala 1-5, da parte di 943 users su 1682 items (film). I file appartengono al dataset `Movielens100K`.

Dopo aver costruito la matrice del training set con il comando `A=spconvert(u1training(:, 1:3))` e la matrice del test set con il comando `B=spconvert(u1test(:, 1:3))`, si porti B alla stessa dimensione di A.

## 2 Un semplice sistema di Raccomandazione

*Esercizio 1.* Si consideri una matrice  $A$  utenti-items. Si scriva una funzione

```
function [S_user, S_items]=similarity(A)
% A: matrice dei rating utenti-items
% S_user: matrice di similarita' (misura del coseno) tra ogni coppia
% di utenti di A
% S_items: matrice di similarita' (misura del coseno) tra ogni coppia
% di items di A
```

che presa da una matrice  $A$  contenente i ratings di utenti restituisce: la matrice  $S\_user$  delle similarità tra utenti  $S\_user(i,j)=\cos(A(i, :), A(j, :))$  e la matrice  $S\_item$  delle similarità tra item  $S\_item(i,j)=\cos(A(:, i), A(:, j))$

*Esercizio 2.* Si scriva una funzione

```
function [items_list]=recommend(A, user)
% A: matrice dei rating utenti-items
% user: indice di un utente
% items_list: insieme di indici di item da raccomandare a user
```

La funzione deve implementare il seguente schema:

- Costruisca le matrici  $S\_user$ ,  $S\_item$  chiamando la funzione `similarity`.
- Analizzando  $S\_user$  restituisca l'indice  $i_{\max}$  dell'utente (diverso da `user`) con cosine similarity più alta per l'utente `user`.
- scriva in `items_list` gli items che  $i_{\max}$  ha valutato con un punteggio 4 o 5 **non ancora** valutati di `user`.

*Esercizio 3.* Si scriva una funzione

```
function [p, r]=accuracy(B, item_list, user)
% B: matrice user-items (test-set)
% item_list: insieme di indici
% user: identificativo di un utente
% p: valore di precision
% r: valore di recall
```

che implementi il seguente schema

- calcoli il numero dei “true positive”  $tp=\#\{i|i \in \text{item\_list} \cap B(\text{user}, i) \geq 4\}$
- calcoli il numero dei “false positive”  $fp=\#\{i|i \in \text{item\_list} \cup B(\text{user}, i) < 4\}$
- calcoli il numero dei “false negative”  $fn=\#\{i|i \notin \text{item\_list} : B(\text{user}, i) \geq 4\}$
- restituisca  $p=tp/(tp+fp)$  come valore di precision e  $r=tp/(tp+fn)$  come valore di recall

Per ogni utente nel Test-set (cioè che nella matrice  $B$  ha almeno una valutazione) si calcoli l'insieme di item da raccomandare con la funzione `recommend(A, user)` e i valori di precision e recall. Si calcolino i valori medi su tutti gli utenti.

## 2.1 Il sistema Pure-SVD

Esercizio 4. Si scriva una funzione

```
function [A_mean]=transform(A){  
% A: matrice user-items  
% A_mean: matrice user_items
```

La funzione deve costruire una matrice **A\_mean** che sostituisce a tutti i valori zero in **A** la media dei punteggi che ha attribuito agli item votati.

Si scriva una funzione

```
function [C]=pure_svd(A, k){  
% A: matrice user-items  
% C: matrice user_items
```

La funzione deve

- costruire **A\_mean** utilizzando la funzione del punto precedente
- costruire con la SVD l'approssimazione  $M$  di rango  $k$  di **A\_mean**
- costruire la matrice  $B$  tale che

$$B_{ij} = \begin{cases} a_{ij} & \text{se } a_{ij} \neq 0 \\ m_{ij} & \text{se } a_{ij} = 0. \end{cases}$$

- restituisce la matrice  $C$  ottenuta da  $B$  arrotondando i valori  $d$  all'intero più vicino e tagliando i valori all'intervallo  $[0, 5]$  secondo la regola  $c_{ij} = 0$  se  $\text{round}(b_{ij}) < 0$  e  $c_{ij} = 5$  se  $\text{round}(b_{ij}) \geq 6$ .

Si scriva una funzione

```
function [U]=recommend_svd(A,C)  
% A: matrice user-item  
% C: matrice user-item ottenuta con la funzione pure_svd  
% U: matrice user-item binaria
```

La funzione deve restituire una matrice user-item  $U$  tale che  $U_{ij} = 1$  se  $a_{ij} = 0$  e  $c_{ij} \geq 4$ .

Utilizzando la funzione `accuracy` si calcolino i valori medi di precision e recall su tutti gli utenti del Test-set.