

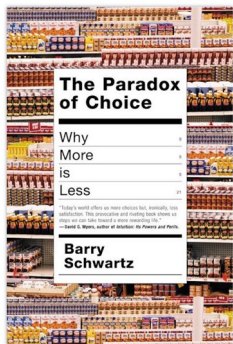
Sistemi di Raccomandazione

Corso di AA, anno 2018/19, Padova



Fabio Aioli

19 Dicembre 2018



” Troppa scelta rende le persone meno propense a scegliere e meno felici della scelta effettuata”



- 24 tipi di marmellata:
Il 60% dei clienti è stato attratto dal banco. In media 2 assaggi. Solo il 3% ha acquistato.
- 6 tipi di marmellata:
Il 40% dei clienti è stato attratto dal banco. In media 2 assaggi. Il 30% ha acquistato.



- Se guardando i prodotti di **Amazon** ti accorgi che il sistema ti suggerisce altri prodotti "correlati", oppure noti messaggi del tipo "chi ha acquistato questo, ha acquistato anche.."
- Se **Facebook** e **Twitter** ti raccomandano nuovi amici o persone da seguire in base a amici o followers attuali.
- Se **Netflix** ti suggerisce film che potresti essere interessato a guardare basandosi su cosa hanno guardato altri utenti simili a te o cercando di prevedere in base al genere, al regista, ecc. i film che potrebbero essere di tuo gradimento
- Se se se ... allora ...
- Sotto sotto c'è un **sistema di raccomandazione!**



Dal 2006 al 2009, Netflix sponsorizzò una famosa competizione offrendo 1.000.000\$ al team che, basandosi su un dataset di

- circa 480K utenti (users)
- circa 18K film (items)
- oltre 100M ratings,

fosse stato capace di migliorare anche solo del 10% le prestazioni dell'algoritmo al tempo utilizzato da Netflix per la predizione dei rating mancanti.

- R.M. Bell, Y. Koren, C. Volinsky (2007).
"The BellKor solution to the Netflix Prize"
- R.M. Bell, J. Bennett, Y. Koren, and C. Volinsky (2009).
"The Million Dollar Programming Prize"



Explicit Feedback se disponibile:

- Rate degli item su una scala ordinata (stellette);
- Un ordine degli items in base alla preferenza;
- Preferenze su coppie di item.

Implicit Feedback se disponibile:

- L'elenco degli items che l'utente ha visionato/comperato in precedenza;
- Analisi dei tempi di permanenza di un utente in un sito;
- La rete sociale di un utente (contextual RS).



- **Content Based (CB)**
 - Raccomando gli item più simili a quelli per cui l'utente ha già mostrato interesse. P.e. stesso genere (film), stesso autore (song) ecc.
- **Collaborative Filtering (CF)**
 - Raccomando a un utente gli item più simili a quelli che piacciono ai suoi simili o, viceversa, gli item più simili a quelli che piacciono a lui.
 - **Similarità ITEM-ITEM**: due oggetti sono simili se tendono ad ottenere lo stesso rate dagli utenti
 - **Similarità USER-USER**: due utenti sono simili se tendono a dare lo stesso rate agli oggetti
 - Nota che in questo tipo di approcci non si usa conoscenza specifica su oggetti e utenti ma solo caratteristiche del comportamento sociale determinato dall'interazione utenti-items (ratings)



- Metodi Ibridi

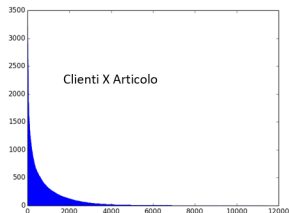
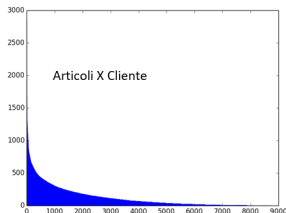
- $CB > CF$ quando ho poco storico (cold-start problem)
- $CF > CB$ quando linformazione implicita contenuta sulla interazione (rete sociale utenti-items) diventa prevalente rispetto a quella esplicita del contenuto
- In casi intermedi possiamo utilizzare metodi ibridi che combinano entrambi gli approcci

Matrice di Rating per feedback esplicito

La matrice di Rating rappresenta per ogni coppia utente/item il gradimento (la rilevanza) dell'item per l'utente. È una matrice molto sparsa, tipicamente solo 0.1% di valori presenti!

Ratings	Item_1	Item_2	Item_3	...	Item_m
User_1	4		5		1
User_2		2			2
...					
User_n	2		3		1

Effetto
Long-tail





- **Rate prediction** (explicit feedback) in cui vogliamo predire il rate nelle entries mancanti della matrice di rating
- **Item (o User) TOP-N recommendation** in cui vogliamo predire gli N items di maggior gradimento per un dato user



Million Song Dataset Challenge

Thursday, April 26, 2012

Kudos • 150 teams

Thursday, August 9, 2012

Finished

Dashboard

- Home
- Data
- Information
 - Description
 - Evaluation
 - Rules
 - SubmissionInstructions
 - Prizes
 - F.A.Q.
 - Resources
- Forum
- Leaderboard
 - Public
 - Private
- My Team

Leaderboard

1. aio
2. learner
3. nohair
4. Team Ubuntu
5. TheMiner

Predict which songs a user will listen to.

The Million Song Dataset Challenge aims at being the best possible offline evaluation of a music recommendation system. Any type of algorithm can be used: collaborative filtering, content-based methods, web crawling, even human oracles!* By relying on the [Million Song Dataset](#), the data for the competition is completely open: almost everything is known and possibly available.

What is the task in a few words? You have: 1) the full listening history for 1M users, 2) half of the listening history for 110K users (10K validation set, 100K test set), and you must predict the missing half. How much easier can it get?

The most straightforward approach to this task is pure collaborative filtering, but remember that there is a wealth of information available to you through the [Million Song Dataset](#). Go ahead, explore! If you have questions, we recommend that you consult the [MSD Mailing List](#).

Ready to start recommending? Read through our [Getting Started](#) tutorial. You can also look at this [open-source solution](#) offered by a [contestant](#).

For a more technical introduction to the MSD Challenge, see our [AdMIRE paper](#). (Please use this following [citation](#) when referring to the contest in an academic setting.)

* This contest is for computer models, but if you manage to get recommendations from humans for 110K listeners, we'd like to know how!



- **Rate Prediction**: (problema di regressione): Matrix Factorization, ovvero si apprende una rappresentazione per gli utenti e per gli items in modo che il loro prodotto scalare approssimi i rates presenti
- **Top-N recommendation** (problema di ranking): Matrix Factorization su preferenze (ranking tra coppie di items/users). Il problema qui come trattare i dati mancanti!



Valutazione del rating

Rates Prediction (Root Mean Square Error, RMSE): Sia \hat{r}_{ui} la predizione del rate data dal mio predittore e r_{ui} la reale valutazione data all'item i dall'utente u ,

$$RMSE = \sqrt{\frac{1}{|Te|} \sum_{(u,i) \in Te} (r_{ui} - \hat{r}_{ui})^2}$$

Top-N Recommendation: si usano metriche adatte per il ranking, infatti bisogna valutare quanto gli item effettivamente rilevanti per un utente vengano messi in alto nel ranking prodotto dal predittore

- AUC (Area Under ROC Curve): dove in pratica si calcola il numero di coppie di item (positivo,negativo) correttamente ordinate
- Alternative più "precision-oriented": prec@n, MAP, NDCG, ecc.



Matrix Factorization e Regressione: Cerchiamo vettori \mathbf{x}_u e \mathbf{y}_i per ogni utente e per ogni item, tale che $\hat{r}_{ui} = \mathbf{x}_u^\top \mathbf{y}_i$, e

$$\min_{\mathbf{x}_u} \sum_{i \in R(u)} |r_{ui} - \mathbf{x}_u^\top \mathbf{y}_i|^2 + \beta_u \|\mathbf{x}_u\|^2$$

$$\min_{\mathbf{y}_i} \sum_{u \in R(i)} |r_{ui} - \mathbf{x}_u^\top \mathbf{y}_i|^2 + \beta_i \|\mathbf{y}_i\|^2$$

Nearest Neighbors Based:

$$\hat{r}_{ui} = \frac{\sum_{v \in R(i)} \mathbf{w}_{uv} r_{vj}}{\sum_{v \in R(i)} \mathbf{w}_{uv}} \quad \text{e} \quad \hat{r}_{ui} = \frac{\sum_{j \in R(u)} \mathbf{w}_{ij} r_{uj}}{\sum_{j \in R(u)} \mathbf{w}_{ij}}$$



$$\min_{\mathbf{x}_u} \sum_{i \in R(u)} |r_{ui} - \mathbf{x}_u^\top \mathbf{y}_i|^2 + \beta_u \|\mathbf{x}_u\|^2$$

$$\min_{\mathbf{x}_u} Q(\mathbf{x}_u) = \sum_{i \in R(u)} \underbrace{|r_{ui} - \sum_s x_{us} y_{is}|^2}_{\epsilon_{ui}} + \beta_u \|\mathbf{x}_u\|^2$$

$$\nabla Q(x_{us}) = -2 \sum_{i \in R(u)} \epsilon_{ui} y_{is} - \beta_u x_{us}$$

$$x_{us} \leftarrow x_{us} - \eta \nabla Q(x_{us})$$



$$\min_{\mathbf{y}_i} \sum_{u \in R(i)} |r_{ui} - \mathbf{x}_u^\top \mathbf{y}_i|^2 + \beta_i \|\mathbf{y}_i\|^2$$

$$\min_{\mathbf{y}_i} Q(\mathbf{y}_i) = \sum_{u \in R(i)} \underbrace{|r_{ui} - \sum_s x_{us} y_{is}|^2}_{\epsilon_{ui}} + \beta_i \|\mathbf{y}_i\|^2$$

$$\nabla Q(y_{is}) = -2 \sum_{u \in R(i)} \epsilon_{ui} x_{us} - \beta_i y_{is}$$

$$y_{is} \leftarrow y_{is} - \eta \nabla Q(y_{is})$$



$$\mathbf{w}_{uv} = \frac{|I(u) \cap I(v)|}{|I(u)|^{\frac{1}{2}} |I(v)|^{\frac{1}{2}}}$$

$$\mathbf{w}_{ij} = \frac{|U(i) \cap U(j)|}{|U(i)|^{\frac{1}{2}} |U(j)|^{\frac{1}{2}}}$$

dove:

- $I(u)$ è l'insieme di item che sono stati ratati dall'utente u
- $U(i)$ è l'insieme di users che hanno ratato l'item i

Link Prediction



- Predizione di link (futuri) a partire da una rete sociale di individui
 - L'approccio tipico prevede di mappare il problema in un problema di ranking/classificazione
 - In pratica ogni possibile arco è rappresentato da un insieme di features
- Common Neighbors
 - Jaccard o altre misure di correlazione
 - Analisi dei path tra i due nodi
 - Ecc.

The screenshot shows the Facebook Recruiting Competition page. At the top, there is a blue header with the Facebook logo and the text "Facebook Recruiting Competition". Below this, it says "Tuesday, June 5, 2012" and "Jobs • 418 teams". On the right side, it says "Finished" and "Tuesday, July 10, 2012".

The main content area is divided into several sections:

- Dashboard:** A sidebar menu with options: Home, Data, Information, Forum, Leaderboard, and My Team.
- Information:** A section with sub-headers: Description, Background, Evaluation, Rules, Prizes, and Submission Instructions.
- Forum:** A section with sub-headers: Public and Private.
- Leaderboard:** A section showing a list of participants: 1. Akiba Varadar, 2. Manojan Dinty, 3. QTAC, 4. Glen, 5. Anonymous 12278, 6. Miguel, 7. Cleomb Topphyle, 8. Brady Bennett.

The main content area contains the following text:

Show them your talent, not just your resume.

Looking for Round 2 ?

Want an interview at Facebook? Facebook will review the top entries in the competition and offer you an interview if they like what they see. This is your opportunity to demonstrate your skills on a real-world social network dataset, and show them your creativity, open-mindedness and tenacity in the face of an open-ended predictive modeling problem.

The challenge is to recommend missing links in a social network. Participants will be presented with an external anonymized, directed social graph (no, not Facebook, keep guessing) from which some edges have been deleted, and asked to make ranked predictions for each user in the test set of which other users they would want to follow.

Please note: You must compete as an individual in recruiting competitions. You may only use the data provided to make your predictions. Facebook will review the code of the top participants before deciding whether to offer an interview.