Apprendimento Automatico

Fabio Aiolli
Università di Padova

Sistemi di Raccomandazione

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 amici o following 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

...

Allora sotto sotto c'è un sistema di raccomandazione...

Netflix Prize

- Dal 2006 al 2009, Netflix sponsorizzò una famosa competizione offrendo 1.000.000$ al team che, basandosi su un dataset di oltre 100M di ratings, fosse capace di migliorare anche solo del 10% le prestazioni dell'algoritmo al tempo utilizzato da Netflix.

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

Tipologie di feedback in RS

Explicit Feedback se è disponibile:

- Rate degli item su una scala ordinata (stellette).
- Un ordine degli items dal più favorito al meno favorito.
- Preferenza su coppie di item.

Implicit Feedback se è disponibile:

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

Approcci 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

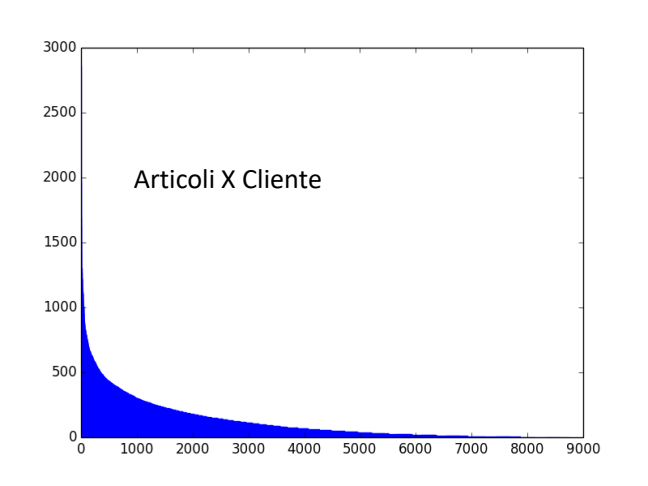
- Metodi Ibridi
  - CB > CF quando ho poco storico (cold-start problem)
  - CF > CB quando l'informazione implicita contenuta sulla interazione (rete sociale utenti-items) diventa prevalente rispetto a quella esplicita del contenuto
  - In casi intermedi possiamo utilizzare metodi ibridi che combinano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

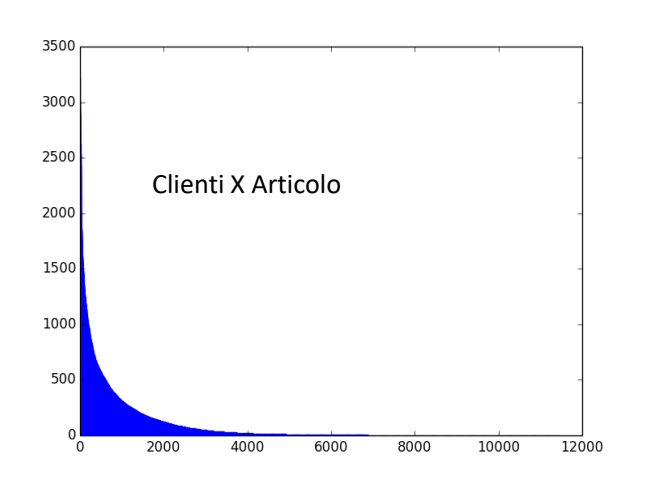
Rating Matrix

R	Item_1	Item_2	Item_3	…	Item_m
User_1	4		5		1
User_2		2			2
…					
User_n	2		3		1

- Matrice molto sparsa, tipicamente 0.1% di valori presenti

Effetto Long-Tail





Problemi in RS

- Rate prediction (explicit feedback) in cui vogliame 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

Million Song Dataset Challenge

Finished

Thursday, April 26, 2012 — Kudos • 150 teams — Thursday, August 9, 2012

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!

Metodi per Collaborative Filtering

- 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 è come trattare i dati mancanti!

Valutazione Rating

Rating Espicito

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

Root Mean Square Error

Top-N

AUC (Area Under ROC Curve),
prec@n,
ecc.

Collaborative Filtering

- Matrix Factorization e regressione

$$r_{ui} = \mathbf{x}_u^\top \mathbf{y}_i$$

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

- Neirest Neighbors based CF

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

Discesa Gradiente su Xu

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

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

$$\nabla Q(\mathbf{x}_{us}) = -2 \sum_{i \in R(u)} \epsilon_{ui} \mathbf{y}_{is} - \beta_u \mathbf{x}_{us}$$

$$\mathbf{x}_{us} \leftarrow \mathbf{x}_{us} - \eta \nabla Q(\mathbf{x}_{us})$$

Discesa Gradiente su Yi

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

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

$$\nabla Q(\mathbf{y}_{is}) = -2 \sum_{u \in R(i)} \epsilon_{ui} \mathbf{x}_{us} - \beta_i \mathbf{y}_{is}$$

$$\mathbf{y}_{is} \leftarrow \mathbf{y}_{is} - \eta \nabla Q(\mathbf{y}_{is})$$

Neirest Neighbors
Similarità tra utenti e tra items

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

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

$U(i)$ Insieme di utenti che hanno comperato l'oggetto i

$I(u)$ Insieme di oggetti comperati dall'utente u

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 Neighbours
- Jaccard o altre misure di correlazione
- Analisi dei path esistenti tra i due nodi (hitting time, pagerank, ecc.)
- Ecc