Recommender System for Movies

Big Data Project

Author: Filippo Guerranti

https://github.com/filippoguerranti/recommender_system

Outline

1. Dataset
2. Collaborative filtering
3. Content based

Dataset

Dataset

MovieLens

5-star rating and **free-text tagging** activity about movies

9742	**610**	**3683**	**100863**
movies	users	tags	ratings
`movies.csv`	`only IDs`	`tags.csv`	`ratings.csv`

Files

`movies.csv`

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
...	...	...

`ratings.csv *`

userId	movieId	rating
1	1	4.0
1	3	4.0
1	6	4.0
...	...	...

`tags.csv *`

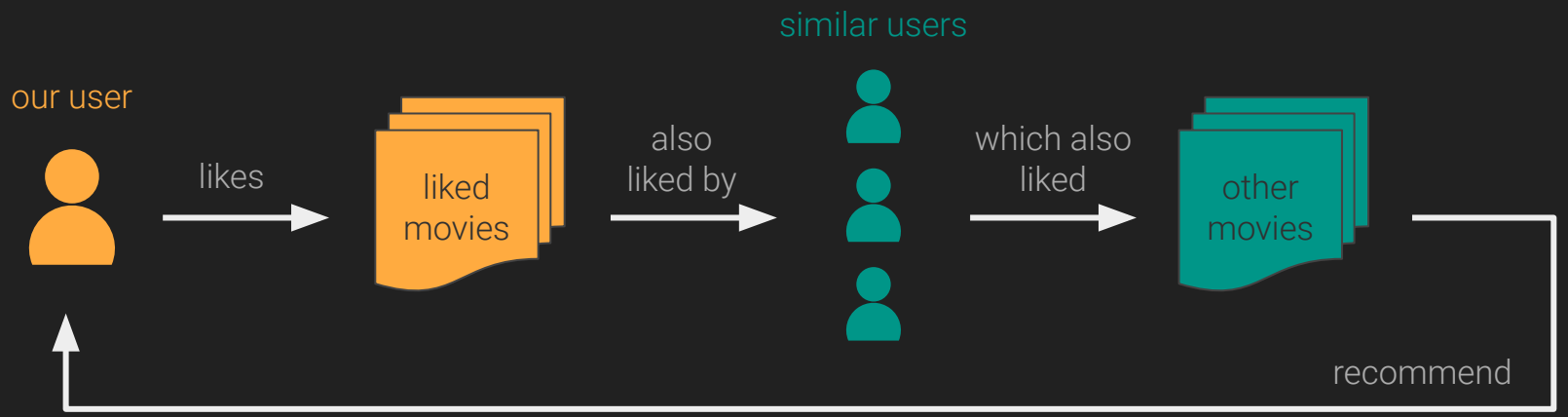
userId	movieId	tag
2	60756	funny
2	60756	Highly quotable
2	60756	will ferrell
...	...	...

* I have removed the timestamp column

Collaborative Filtering

Collaborative Filtering

Idea: find movies recommendations for a user identified by userId based on the ratings of users that have shown similar behaviour (similar ratings).



Collaborative Filtering

Libraries: pandas, numpy, **surprise**

- Reader
- Dataset
- accuracy
- KNNBaseline
- SVD
- model_selection.KFold

Outline:

- Implementation
  - different algorithms
  - 5-fold cross-validation
  - accuracy measures
- Results
- Predictions

Implementation

Algorithms:

- **KNN**
  - Cosine similarity and Baseline (KNN-cosine-baseline)
  - Pearson's Coefficient and Baseline (KNN-pearson-baseline)

- **SVD**
  - N. epochs: 20 | Learning Rate: 0.005 (SVD-20ep-.005lr)
  - N. epochs: 50 | Learning Rate: 0.003 (SVD-50ep-.003lr)

KNN

Goal: estimate the rating that user **x** will give to item **i**

Idea: find the **k** most similar items to item **i** that are rated by **x**

$$\hat{r}_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} sim(i,j) \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} sim(i,j)}$$

$sim(i,j)$ similarity measure between item **i** and item **j**

b_{xi} baseline estimate for user **x** and item **i** $\quad b_{xi} = \mu + b_x + b_i$

DEGRESSION

SVD

DEGRESSION

Goal: estimate the rating that user **x** will give to item **i**

Idea: use dimensionality reduction to find the **latent factors**

$$R \simeq QP^T \qquad \hat{r}_{xi} = \sum_s q_{is} \cdot p_{fs}$$

Find P and Q: by the **Stochastic Gradient Descent** method

$$\min_{P,Q} \sum_{training} (r_{xi} - q_i p_x)^2 + \lambda \left[\sum_x ||p_x||^2 + \sum_i ||q_i||^2 \right]$$

Implementation

5-fold cross-validation:

For each algorithm:

For each fold:

* **split** the dataset into training set and test set
* **train** the algorithm on the training set
* **test** the algorithm on the test set
* measure the **accuracy** of the current fold

Implementation

Accuracy measures:

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\sum_{test} \frac{(\hat{r}_{xi} - r_{xi})^2}{N}}$$

Results

Algorithm	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
KNN-cosine-baseline	0.869872	0.883382	0.879242	0.874197	0.879286	0.877196
KNN-pearson-baseline	0.879939	0.875910	0.875314	0.878922	0.877261	0.877469
SVD-20ep-.005lr	0.870731	0.884309	0.871147	0.866316	0.877215	**0.873944**
SVD-50ep-.003lr	0.868532	0.875322	0.879058	0.879815	0.868313	0.874208

Predictions

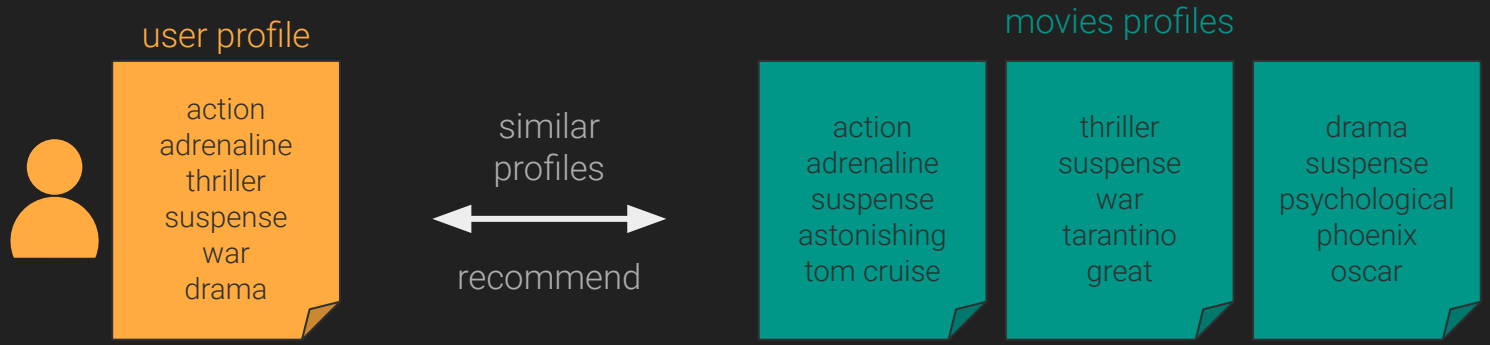
Top 5 recommendations for user 238

movieId	est_rating	title	genres
3275	4.55	Boondock Saints, The (2000)	Action|Crime|Drama|Thriller
5690	4.52	Grave of the Fireflies (Hotaru no haka) (1988)	Animation|Drama|War
3972	4.48	Legend of Drunken Master, The (Jui kuen II) (1...	Action|Comedy
7371	4.44	Dogville (2003)	Drama|Mystery|Thriller
7008	4.44	Last Tango in Paris (Ultimo tango a Parigi) (1...	Drama|Romance

Content based

Content Based

Idea: find movies recommendations for a user identified by userId based on the similarity between the user profile and the movie profile (content).



Content Based

Libraries: pandas, numpy, **sklearn**

- feature_extraction.text.TfidfVectorizer
- feature_extraction.text.CountVectorizer
- metric.pairwise.cosine_similarity
- decomposition.TruncatedSVD

Outline:

- Implementation
  - Movies class
  - Users class
  - ContentBased class
- Predictions

Implementation

Movies class: takes in input the movies and the tags DataFrames and creates the movies profiles.

```
def __init__(self, movies_df, tags_df, tfidf=True, lsa=True, n_components=40)
def __create_movies_profiles(self, movies_df, tags_df)
def __create_movies_terms(self, tfidf=True)
def __latent_semantic_analysis(self, n_components=40)
def movie_vector(self, movieId)
```

Implementation

Users class: takes in input the ratings DataFrame and an instance of Movies class and creates the users profiles.

```
def __init__(self, ratings_df, movies_instance)
def __create_users_movies_dict(self, ratings_df)
def __create_users_vectors(self, mv)
def user_vector(self, userId)
```

Implementation

ContentBased class: takes in input an instance of Users class and an instance of Movies class and creates the recommendations.

```
def __init__(self, users_instance, movies_instance)
def recommend(self, userId, n_recommendations=10)
```

Recommendations are created by computing the **cosine similarity** between the user profile and all the movies profiles.

Predictions

Top 5 recommendations for user 238

movieId	similarity	title	genres
5628	0.894	Wasabi (2001)	Action|Comedy|Crime|Drama|Thriller
5027	0.894	Another 48 Hrs. (1990)	Action|Comedy|Crime|Drama|Thriller
1432	0.894	Metro (1997)	Action|Comedy|Crime|Drama|Thriller
145	0.894	Bad Boys (1995)	Action|Comedy|Crime|Drama|Thriller
20	0.894	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller

Comparison

movieId	est_rating	title	genres
3275	4.55	Boondock Saints, The (2000)	**Action|Crime|Drama|Thriller**
5690	4.52	Grave of the Fireflies (Hotaru no haka) (1988)	Animation|**Drama**|War
3972	4.48	Legend of Drunken Master, The (Jui kuen II) (1...	**Action|Comedy**
7371	4.44	Dogville (2003)	**Drama**|Mystery|**Thriller**
7008	4.44	Last Tango in Paris (Ultimo tango a Parigi) (1...	**Drama**|Romance

Collaborative filtering

movieId	similarity	title	genres
5628	0.894	Wasabi (2001)	**Action|Comedy|Crime|Drama|Thriller**
5027	0.894	Another 48 Hrs. (1990)	**Action|Comedy|Crime|Drama|Thriller**
1432	0.894	Metro (1997)	**Action|Comedy|Crime|Drama|Thriller**
145	0.894	Bad Boys (1995)	**Action|Comedy|Crime|Drama|Thriller**
20	0.894	Money Train (1995)	**Action|Comedy|Crime|Drama|Thriller**

Content based

Code can be found at

https://github.com/filippoguerranti/recommender_system