Recommender Systems

Of Steam® Games with Python

Ye Yizhou, Gong Jin, Stefanidis Kostas

Project purpose

- Hybrid recommendation system for Steam.
- Usual approach: either user- or product centric recommendations.
- Goal of project: more accurate recommendations with more input data.

What did we do - main parts

High level

- Data mining for our source data.
- Algorithm for the recommendation system.

Expected end result of project:

- Recommendations on a rating from 1-5.

Data mining

- Data source: Two datasets from https://www.kaggle.com and one from https://store.steampowered.com/api/appdetails.
- Roughly 14k games and 200k user interactions.
- Combining these datasets was a challenge.
- An issue with missing data.
- Variables: Price, time spent playing, genres and release date.

Algorithm - Calculating ratings and similarities

- Rating from 1-5 for each of the variables. For instance, the game that any given user has played the most would get a rating of 5.
- For any given genre the rating would be the average of all the games belonging in that genre that the user owns.

- Pearson correlation to calculate the similarities between the users and the items.

$$\mathrm{simil}(x, y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

- Where Ixy is the set of items rated by both userX and userY.

User-Game table

	Game1	Game2	Game3	Game4	Game5	Game6	...	GameN
User1	1	3	5	5	2	1		2
User2	1	5		2	5	1		1
User3	5	2	4		1			1
User4		5		4		1		
UserN	1		5	3	4	2		3

Pearson similarity of users

	User1	User2	User3	User4	User5	User6	...
User1	1	0.2	0.3	0.4	-0.5	-1	
User2	0.2	1	-0.4	0.5	0.1		
User3	0.3	-0.4	1	0.1	-0.3	0.2	
User4	0.4	0.5	0.1	1	0.9		
...							

Algorithm - End formula

- The final rating that our hybrid recommender system assigns to a game is the average of a combination of 5 variables.

1. The average rating of the 5 most similar users
2. The average rating of the 5 most similar games
3. The rating towards games belonging in the same genre.
4. The rating towards games belonging in the given time period.
5. The rating towards games belonging in the given price zone.

Algorithm - End formula

Name	Dota 2
Genre	Action, Free to Play, Strategy
Release Date	2013.7.9
Price	0

Final rating = (3.8+3.4+4+4.5+5)/5 = 4.14

The average rating of the 5 most similar users

The average rating of the 5 most similar games

Rating from the user's attitude to games from [2010, 2015]

Rating from the user's genre preferences

Rating from the user's preference to Free to Play games

Example result

```
In [1]: runfile('D:/Courses/Recommend sys/untitled0.py', wdir='D:/Courses/Recommend sys')
For user 10524 we recommand:
PLAYERUNKNOWN'S BATTLEGROUNDS
Outlast 2
SMITE
Rocket League
Fallout Shelter
```

Method evaluation

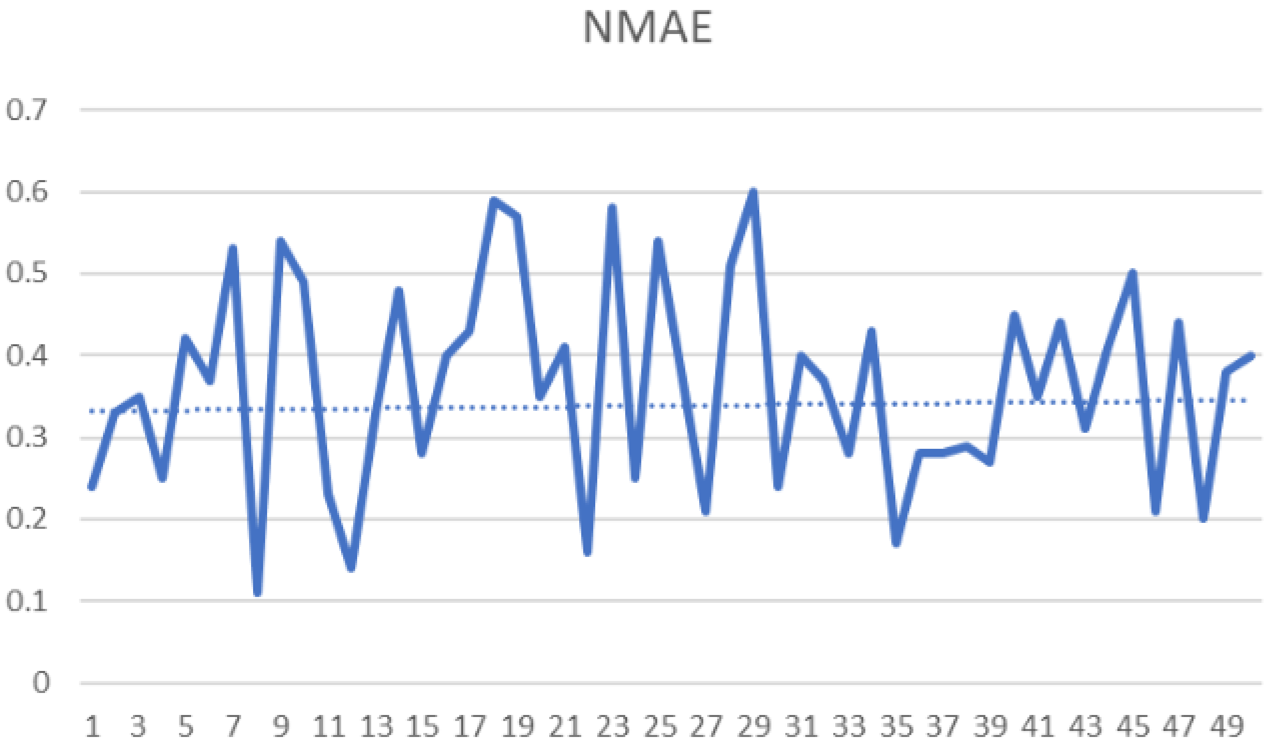
- Results evaluated by separating our dataset into a test set and a training set (10% and 90% split).
- Using the training set to predict the values in the test set.

- Normalized Mean Absolute Error = 0.33, which means our model has an accuracy around 67% on predicting.

$$MAE = \frac{\sum_{i=1}^{n} | p_i - q_i |}{n}$$

$$NMAE = \frac{MAE}{R_{\max} - R_{\min}}$$

Method evaluation



Conclusions/ideas to improve

- Accuracy improved upon adding more variables.
- We did not assign weights to different variables.
- A real application with a UI.
- We tried to generate rules based on the data. However, most of them had a very low coverage due to data sparsity, in spite of having a high accuracy.
- These rules were discarded due to overfitting.

Thank you!

- contact.yizhou@gmail.com