Romain Legrand MS BIG-Data

Project SD701 : movie recommendation

1. Project description

The aim of the project is to provide movie recommandations, using the Movie Lens database. I want to propose alternative metrics and methodologies compared to what was done in TP1. Specifically, I develop two metrics : a similarity index (part 1), and a ratings system (part 2) based on linear regression. The project uses Spark, with extensive use of RDDs for part 1 and dataframes for part 2.

2. A first methodology : a similarity-based recommandation system

By contrast with TP1 where recommandations were based on user similarities, I propose here an alternative index based on movie similarities. Specifically, for any given user, I create a benchmark movie profile from his ratings history. I then calculate a similarity index bewween this benchmark and any other movie in the database. The recommandations obtain as the movies with the highest similarity index.

a) benchmark features

The calculation of the benchmark implies four features :

1/ the average release year of the movies rated by the user. This carries information about both the age of the user, and the period at which was released the kind of movies he likes. The database does not propose the release year as a base variable. It can however be obtained indirectly from the column « title » after some parsing, as each movie title is followed by the release year.

2/ the average of the ratings delivered by the user. This can look questionable as it may mostly reveal how strict the user is in his ratings. I believe however that it does carry useful information about tastes. For instance, a user systematically producing high ratings is likely to be little demanding and watch mostly blockbusters and wide audience movies. By contrast, a user producing a larger number of low ratings may be identified as a risk-seeking user, looking for less mainstream productions, with some disappointment. This assumes that ratings don’t represent only scores, but also contain information about how much mainstream a movie can be.

3/ the average number of ratings on the movies rated by the user. This also reflects how much mainstream or « wide audience » a movie is, and thus contains information about the type of movies appreciated by the user.

4/ the genres of movies rated by the user. This is self-explanatory : if a user tends to watch mostly movies from a few given genres (like action and horror), serving him with the same genre of movies will likely be more succesful than proposing a completely different genre (like romance).

b) calculation of the similarity index

The index is calculated as follows. Denote respectively by y_u , r_u and n_u the average year, rating and number of ratings of movies from the target user. Also, denote by g_u the vector of dimension 18 containing for each of the 18 possible genres the proportion of movies watched by the user corresponding to this genre[1]. For any other movie *m* in the database, denote respectively by y_m , r_m and n_m the release year, average rating and number of ratings for this movie, and by g_m the 18-entry vector containing binary (0 or 1) values depending on whether the movie matches the corresponding genre of not. Denote by w_1 , w_2 , w_3 and w_4 a set of four weight such that $w_1+w_2+w_3+w_4=1$, and by α and δ a set of positive curvature values. Then the index between *u* and *m* is given by :

$$ind_{(u,m)}=w_1\frac{\arctan(\alpha|y_m-y_u|)}{2\pi}+w_2(1-|r_m-r_u|/5)+w_3\frac{\arctan(\delta|n_m-n_u|)}{2\pi}+w_4(g_u^T g_m)/(1^T g_m)$$

The index comprises four elements. The first one considers the distance between the movie year of release y_m and the benchmark year y_u , normalised to 1 thanks to the *arctan* function. α represents the curvature of the function, and for my purpose an α value of 0.0005 proved reasonable. The second component considers the distance between the movie and benchmark ratings r_m and r_u . It is also normalised to obtain a maximum value of 1 when the rating for the movie and the benchmark are equal. The third component considers the distance between the number of ratings of the movie n_m and that of the benchmark n_u , again normalised with an *arctan* device. A curvature factor of $\delta=0.05$ proved good enough in this case. The final component is the genre similarity component. It is an average genre value compared to the benchmark. For instance, if movie *m* has genres action and adventures, and the benchmark has corresponding proportion coefficients of 0.34 and 0.42 for these two genres, then the average 0.38 similarity factor will be returned.

Finally, the set of weights w_1 , w_2 , w_3 and w_4 ponderate how much importance is granted to each component and constrain the index value between 0 and 1. For simplicity, and to consider the impact of all components, I propose $w_1=w_2=w_3=w_4=0.25$. These hyperparameters may however be tuned to apprehend different aspects of the recommandations.

c) Running the recommandation programme

Consider for example user 1 in the database. A sample of movies rated by this user is as follows :

```
"Usual Suspects, The            1995            Crime|Mystery|Thriller
From Dusk Till Dawn             1996            Action|Comedy|Horror|Thriller
Braveheart              1995            Action|Drama|War
Canadian Bacon          1995            Comedy|War
Billy Madison           1995            Comedy
Tommy Boy               1995            Comedy
Forrest Gump            1994            Comedy|Drama|Romance|War
Dazed and Confused              1993            Comedy
"Three Musketeers, The          1993            Action|Adventure|Comedy|Romance
Tombstone               1993            Action|Drama|Western
Dances with Wolves              1990            Adventure|Drama|Western
Pinocchio               1940            Animation|Children|Fantasy|Musical
Fargo           1996            Comedy|Crime|Drama|Thriller
Mission: Impossible             1996            Action|Adventure|Mystery|Thriller
"Rock, The              1996            Action|Adventure|Thriller
Twister         1996            Action|Adventure|Romance|Thriller
"Ghost and Mrs. Muir, The               1947            Drama|Fantasy|Romance
Escape to Witch Mountain                1975            Adventure|Children|Fantasy
"Three Caballeros, The          1945            Animation|Children|Musical
"Sword in the Stone, The                1963            Animation|Children|Fantasy|Musical
```

1 The sum of the vector entries may exceed 1 given that most movies have several genres listed.

This user favours movies from the 1990's, and a few older movies from the 1940's and 1960's. He has a taste for action, adventure, comedy, and animation movies. If certain movies in this list are very famous blockbusters, others are considerably less known : this user is open-minded and considers movies out of the main stream, with possibly a smaller audience.

Using my similarity-based recommander system, I obtain the following top 20 recommandations :

```
0.794   In the Line of Fire     1993    Action|Thriller
0.792   "Avengers, The  2012    Action|Adventure|Sci-Fi|IMAX
0.791   This Is Spinal Tap      1984    Comedy
0.788   "Fish Called Wanda, A   1988    Comedy|Crime
0.787   "Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il)     1966    Action|Adventure|Western
0.785   Traffic 2000    Crime|Drama|Thriller
0.78    "Royal Tenenbaums, The  2001    Comedy|Drama
0.78    Harry Potter and the Goblet of Fire     2005    Adventure|Fantasy|Thriller|IMAX
0.779   Erin Brockovich 2000    Drama
0.778   "American President, The        1995    Comedy|Drama|Romance
0.778   Django Unchained        2012    Action|Drama|Western
0.777   Scarface        1983    Action|Crime|Drama
0.772   City of God (Cidade de Deus)    2002    Action|Adventure|Crime|Drama|Thriller
0.769   Die Hard 2      1990    Action|Adventure|Thriller
0.769   Robin Hood: Men in Tights       1993    Comedy
0.769   Pirates of the Caribbean: Dead Man's Chest      2006    Action|Adventure|Fantasy
0.768   Mary Poppins    1964    Children|Comedy|Fantasy|Musical
0.767   Amadeus 1984    Drama
0.764   Lost in Translation     2003    Comedy|Drama|Romance
0.764   Big Fish        2003    Drama|Fantasy|Romance
```

The recommandations follow closely the genres appreciated by the user : action, adventure and comedy. Animation movies seem to be left somewhat apart. The release years of the recommanded movies are also consistent with user 1 preferences : mostly around the 1990's, with a bit of 1960's and 1980's. Here again, a few blockbusters (Avengers, Pirates of the Caribbean) side with other less famous movies. Overall, I would say this recommender system produces adequate recommandations.

3. A second methodology : a linear regression recommender system

The first methodology mostly focused on similarities between movies, and paid limited attention to the rating. As an alternative, I also propose a system based on ratings, building on the user personal preferences.

a) the model

I use a simple linear regression model which produces personalised The methodology goes as follows : each user is assumed to rate movies according to his own tastes or features. Those features are :

- the average rating r_m received for this movie. Logically, the higher the average rating, the more likely is any specific user to like the movie as well. This amounts to explaining a rating with other ratings, and it may sound tautological. However, the average ratings represents only one factor among others, and the weight granted to this factor may vary from one user to another.

- the number of ratings n_m . A user may appreciate movies with many ratings, reflecting a wide audience (blockbuster amateur) or on the contrary prefer movies which attract fewer people (author movie amateur). To remain scalable, this variable is switched to log value.
- the standard deviation s_m of ratings on this movie. This carries information about the kind of movies appreciated : consensual movies with small standard deviation on ratings, or more controversial movies with larger standard deviation.

- the year of release y_m of the movie. This reflects information on the age and tastes of the user, along with the style of the movie. To remain scalable, years are formulated as difference from reference year 2000 (so for instance th value for a 1996 movie is -4) .

- the genres of the movie. The database identifies 18 of them, resulting in 18 dummy variables $g_1 \ldots g_{18}$. Because there is one dummy for each genre, the constant is omitted in the regression.

Denoting by r_u the rating attributed to movie y_m by user u , this produces the following regression :

$$r_u = \beta_1 r_m + \beta_2 n_m + \beta_3 s_m + \beta_4 y_m + \sum_{i=1}^{18} \delta_i g_i + \varepsilon$$

b) estimation

For a given user, the model is trained on the movies rated by the user, using his personal ratings as r_u . Once the set of coefficients $\beta_{1,} \beta_{2,} \beta_{3,} \beta_4$ and the series of δ_i coefficients are estimated, the model can be applied to all the movies not visioned by the user. Predicted ratings are then obtained, and recommandations can be produced based on the highest values.

c) Running the recommandation programme

As a comparison exercise, it is interesting to consider again user 1. After training the model, one obtains the following set of coefficients :

```
beta_1  =  0.919      delta_5 =  -0.107      delta_13 =  0.066
beta_2  = -0.08       delta_6 =  -0.011      delta_14 = -0.173
beta_3  =  0.685      delta_7 =   0.0        delta_15 = -0.2
beta_4  = -0.006      delta_8 =  -0.114      delta_16 = -0.176
delta_1 = -0.007      delta_9 =  -0.144      delta_17 = -0.131
delta_2 =  0.015      delta_10 =  0.277      delta_18 = -0.195
delta_3 =  0.065      delta_11 = -0.915
delta_4 =  0.103      delta_12 = -0.093
```

The coefficients are informative on their own. First, $\beta_1 = 0.919$ shows that user 1 attributes significant weight to other user ratings, but nevertheless reacts less than one to one. Thus, other factors matter for his personal ratings. $\beta_2 = -0.07$ and $\beta_3 = 0.68$ show that user 1 likes less movies rated by many people and more movies with high variance on the ratings. This is clearly the sign of a profile oriented towards controversial, or author movies. The slightly negative coefficient $\beta_4 = -0.006$ finally establishes that user 1 prefers older movies, though this is not very strong. The set of genre coefficients δ_i eventually reveals a user appreciating the genres « animation », « children », « film noir » and « mystery » ($\delta_{3,} \delta_{4,} \delta_{10}$ et δ_{13}) .

Using the trained model, the top 20 recommandations for user 1 are :

prediction	title	year	genres
6.896993514812584	Ivan's Childhood (a.k.a. My Name is Ivan) (Ivanovo detstvo)	1962	Drama|War
6.838854832899267	Fanny and Alexander (Fanny och Alexander)	1982	Drama|Fantasy|Mystery
6.584029584480806	Lassie	1994	Adventure|Children
6.439741589808314	Play Time (a.k.a. Playtime)	1967	Comedy
6.417288349375511	Fury	1936	Drama|Film-Noir
6.40235945681551	"Zed & Two Noughts, A	1985	Drama
6.360582930482634	Titanic	1953	Action|Drama
6.298316480478787	"Room, The	2003	Comedy|Drama|Romance
6.199129175735221	Troll 2	1990	Fantasy|Horror
6.15596740468558	Burnt by the Sun (Utomlyonnye solntsem)	1994	Drama
6.009410936784207	Gulliver's Travels	1939	Adventure|Animation|Children
6.008702819041224	Kind Hearts and Coronets	1949	Comedy|Drama
5.956643521665349	"Hard Way, The	1991	Action|Comedy
5.94904300726737	Junior and Karlson	1968	Adventure|Animation|Children
5.937933055030355	Karlson Returns	1970	Adventure|Animation|Children
5.928762764316524	Winnie Pooh	1969	Animation|Children
5.928762764316524	Gena the Crocodile	1969	Animation|Children
5.921268126674832	On the Trail of the Bremen Town Musicians	1973	Adventure|Animation|Children
5.883641440401528	Louis C.K.: Shameless	2007	Comedy
5.864708522130931	Girls About Town	1931	Comedy

The recommandations are overall consistent with the identified profile of user 1 : movies that are not very recent, many of them entering the categories children or animation. Hardly any « film noir » or « mystery » movies are yet to be found in this top 20, a sign that other factors play a significant role. Also, only a few blockbusters appear on the list, confirming the taste of user 1 for more undergournd movies. In this respect, the propositions are consistent with those produced in part 1 of the exercise, though this second recommendation system seems definitiely more « animation » and « children » oriented for user 1.

4. Overall conclusion

I have proposed two alternative recommender systems for movies based on the movie lens database. The two systems seem consistent in terms of the type of movies they recommend, even though the specific titles proposed are not the same.

I can find both strenghts and weaknesses in these two recommender models. Unlike many other systems, they stick closely to the user profile and don't seem biased towards succesful commercial blockbusters. In this respect, they support diversity more than competing recommenders. But this may in fact also represent their main weakness. From a commercial point of view, the movies recommended are not necessarily very famous and little likely to generate high volume of sales.

For movie amateurs, these systems are probably adequate, but for professional from the movie industry, they probably offer too limited commercial perspectives.