Mateusz Lewko, Martyna Siejba

System rekomendujący na podstawie zestawu danych *MovieLens*

Projekt z Eksploracji Danych

Prowadzący: Piotr Wnuk-Lipiński

Wrocław, 11 lutego 2018

Spis treści

1. Opis problemu

Zestaw danych *MovieLens* zawiera 26 milionów ocen w skali od 0.5 do 5.0 dla 27 tysięcy filmów. Naszym celem było zbudowanie modelu przewidującego oceny filmów wystawione przez użytkowników, na podstawie pozostałych ich ocen.

2. Pierwsze podejście

Zdecydowaliśmy się zaimplementować *Collaborative Filtering*, gdyż to podejście działa dla produktów dowolnego typu. Problem *nowego użytkownika* nas nie martwił, ponieważ dane zawierają jedynie użytkowników z co najmniej dwudziestoma ocenami. Przetestowaliśmy kilka wariantów *Collaborative Filtering*: zwykły *user–user*, *item–item* oraz *user–user* gdzie brany jest pod uwagę fakt, że niektórzy użytkownicy częściej daję wyższą lub niższą ocenę niż inni. Dla porównania, użyliśmy prostego modelu, gdzie przewidywana ocena danego filmu to średnia arytmetyczna wszystkich ocen (z danych testowych).

Na podstawie ocen stworzyliśmy macierz $user\_rating$, gdzie $user\_rating_i^j$ to ocena $j - tego$ filmu wystawiona przez użytkownika numer i. We wszystkich podejściach należy zdefiniować miarę podobieństwa między użytkownikami lub filmami. Użyliśmy dwóch najpopularniejszych: odległość kosinusów i wskaźnik Pearsona. Użytkownik jest reprezentowany przez wektor swoich ocen, a film przez wektor ocen użytkowników dla tego filmu. Odległość między dwoma użytkownikami to odległość miedzy odpowiadającymi im wektorami. Analogicznie dla filmów.

Nasz model działał w sposób następujący. W standardowej wersji *user–user* przewidywana ocena filmu to średnia ważona ocen innych użytkowników (dla tego filmu). Wagi stanowiły wartości podobieństwa między użytkownikami. W wersji *item-item* sposób był analogiczny. Wynik to była średnia ocen danego użytkownika, z wagami będącymi podobieństwem między filmami.

3. Rozwinięcie *Collaborative Filtering*

W kolejnym podejściu wykorzystaliśmy pomysł, że niektórzy użytkownicy zazwyczaj dają wyższe lub niższe oceny niż pozostali. W zwykłym podejściu *user–user*, użytkownik A, który lubianym filmom wystawia ocenę 4 istotnie różni się od użytkownika B, który lubianym filmom wystawia ocenę 5, nawet jeśli podobają im się te same filmy. W rzeczywistości mają podobny gust. Postanowiliśmy wziąć to pod uwagę. Zamiast brać średnią z ocen innych użytkowników wystawionych danemu filmowi, bierzemy średnią ważoną różnic między średnią wszystkich ocen danego użytkownika, a jego oceną dla danego filmu. Tym sposobem skupiamy się na relatywnej ocenie filmu przez innych użytkowników, a nie konkretnej wartości tej oceny. Na końcu należy dodać średnią ocen dla przewidywanego użytkownika. Wagi średniej relatywnych ocen to tak jak wcześniej podobieństwa między użytkownikami. Przetestowaliśmy obie miary podobieństwa.

Ostatni pomysł to sposób na ulepszenie modelu *item–item*. Macierz *user_rating*, na podstawie której wyliczane jest podobieństwo między filmami jest rzadka, więc wyznaczone podobieństwa można spróbować poprawić. Zestaw danych *MovieLens* zawiera także informacje o gatunkach danego filmu. Wprowadziliśmy dodatkową miarę podobieństwa: liczba gatunków które się pokrywają między dwoma filmami. Ulepszony model dla wersji *item–item* jako wynik predykcji podaje średnią arytmetyczną wyniku z poprzedniego modelu *item–item* oraz z jego analogicznej wersji z nową miarą.

4. Implementacja

Implementacja wszystkich rozwiązań i testowania znajduje się w pliku *movielens−recommender.ipynb*. Krótki spis ważniejszych funkcji zawartych w notebooku:

1. *gen_test_data* – generowanie danych testowych
2. *simple_mean_predict* – prosty model predykcji, wynik to średnia arytmetyczna ocen
3. *item_based_predict* – podstawowa wersja *item–item Collaborative Filtering*
4. *user_based_predict* – podstawowa wersja *user–user Collaborative Filtering*
5. *item_based_with_genres_predict* – rozszerzona wersja *item–item* z nową miarą odległości
6. *user_relative_predict* – rozszerzona wersja *user–user Collaborative Filtering*

5. Testowanie

Z podanych ocen filmów wybraliśmy losowo 25% jako dane testowe. Dane testowe zostały wymazane z macierzy R i zastąpione zerami. Nie kolidowało to z wystawionymi ocenami – minimalna ocena to 0.5. Należało jednak uważać przy obliczeniach, aby nie policzyć wyzerowanych elementów. Jakość modelu sprawdzaliśmy obliczając błąd $RMSE$ dla danych testowych.

6. Wyniki

$RMSE$ dla danych testowych	
Metoda	Błąd
średnia arytmetyczna ocen	0.95028
podstawowy *user–user*, miara cosinusów	0.94220
podstawowy *user–user*, wskaźnik *Pearsona*	0.94146
rozszerzony *user–user*, miara cosinusów	0.87340
rozszerzony *user–user*, wskaźnik *Pearsona*	0.87265
podstawowy *item–item*, miara cosinusów	0.97895
podstawowy *item–item*, wskaźnik *Pearsona*	0.98414
rozszerzony *item–item*	0.97076

7. Wnioski

Rozszerzona wersja *user–user* okazała się dawać najdokładniejsze wyniki (istotnie lepsze od pozostałych). Zapewne dlatego, że pomysł z różnym podejściem użytkowników do systemu ocen, występuje w praktyce. Podstawowy model *user–user* wciąż był lepszy od *item–item* i od modelu testowego.

Wynik modelu *item–item* był najsłabszy i dawał gorsze wyniki niż najprostszy pomysł ze średnią arytmetyczną ocen (testowy model). Głównym powodem była najpewniej jakość miary podobieństwa filmów. Pomysł z zastosowaniem średniej z dwóch miar (gdzie druga to liczba wspólnych gatunków), trochę poprawił wynik.

We wszystkich przypadkach zastosowanie wskaźnika *Pearsona* dla *user–user* dawało trochę lepsze rezultaty (o około 0.001). Dla *item–item*, wskaźnik *Pearsona* był gorszy o 0.01.