Metody Odkrywania Wiedzy
Sprawozdanie końcowe z projektu

Wojciech Klicki
Konrad Starzyk

21 stycznia 2009

1 Zadanie

Zadanie składa się z dwóch części : implementacyjnej oraz badawczej. Część implementacyjna polega na implementacji metody dokonującej predykcji w środowisku GNU R. Na część drugą składa się szereg testów testujących skuteczność algorytmu.
Realizacja zadania polega na wykonaniu poniższych etapów:

- Postawienie pytań dotyczących danych wejściowych oraz algorytmu kooperatywnej filtracji, na które odpowiedź powinniśmy uzyskać przy pomocy Slope-One.
- Analiza danych – wybranie kategorii oraz atrybutów, które będą analizowane przez nasz algorytm.
- Implementacja algorytmu.
- Wykonanie eksperymentów, a następnie ocena jakości klasyfikatora.

2 Opis algorytmów

2.1 Kooperatywna filtracja

Kooperatywna filtracja polega na przewidywaniu ocen jakie otrzymają produkty od poszczególnych użytkowników na podstawie już ocenionych produktów. Zgadnięcie oceny polega na założeniu pewnego podobieństwa oceny jaką wystawi użytkownik do ocen już wystawionych.

2.2 Technika Slope-One

Przewidywanie preferencji za pomocą techniki Slope-One opiera się na założeniu, że ocenę użytkownika można przybliżyć za pomocą wzoru $f(x) = x + b$, który wyznacza średnią różnicę pomiędzy ocenami dwóch użytkowników którzy

dokonali oceny tego samego elementu. Jest to oczywiście daleko idące uproszczenie – w rozwinięciu tej techniki można korzystać z predyktorów o wzorach $f(x) = ax + b$ lub nawet $f(x) = ax^2 + bx + c$. Jak się jednak okazuje, nawet taki predyktor jest w stanie trafnie przewidywać preferencje użytkowników.

Oznaczmy przez v_i i w_i tablice ocen dla dwóch różnych użytkowników, gdzie $i = 1..n$ jest indeksem przedmiotu. Wtedy $v_i - w_i$ jest różnicą ocen tego samego przedmiotu przez dwóch użytkowników. Spróbujmy znaleźć wartość która najlepiej przybliża różnicę w ocenach dawanych przez tych użytkowników.
Minimalizując wyrażenie: $\sum_i (v_i + b - w_i)^2$ ze względu na parametr b otrzymujemy $b = \dfrac{\sum_i v_i - w_i}{n}$.

Mając zbiór testowy κ oraz dowolne dwa oceniane przedmioty i oraz j, wraz z ich ocenami u_i oraz u_j możemy określić średnie odchylenie przedmiotu i względem j jako:

$$dev_{i,j} = \sum_{u \in S} \frac{u_j - u_i}{|S|}$$

gdzie jako S oznaczymy zbiór ocen które zawierały obydwa przedmioty.
Biorąc pod uwagę, że nieznaną wartość oceny przedmiotu j możemy przewidywać jako $u_j = dev_{i,j} + u_i$, sensowny predyktor mógłby być średnią takich przewidywań:

$$P(u)_j = \frac{1}{|R_j|} \sum_{i \in R_j} (dev_{i,j} + u_i)$$

gdzie R_j jest zbiorem wszystkich przedmiotów i które zostały ocenione i dla których istnieje wyznaczona wartość średniego odchylenia względem przedmiotu j równa $dev_{i,j}$. Co więcej, jeśli dodatowo można zaobserwować, że zbiór danych jest gęsty, czyli że prawie każda z par filmów posiada pewną ocenę, to można przyjąć, że $R_j = S(u)$ A ponieważ

$$\bar{u} = \sum_{i \in S(u)} \frac{u_j}{card(S(u))} = \sum_{i \in R_j} \frac{u_j}{card(R_j)}$$

to możemy przedefiniować predykcję P:

$$P(u)_j = \bar{u} + \frac{1}{|R_j|} \sum_{j \in R_j} (dev_{i,j})$$

Wzór tej postaci jest nieco szybszy do przeliczania (zakładając, że znamy średnie oceny użytkowników).

2.3 Implementacja algorytmu

Przebieg działania programu można podzielić na następujące etapy:

- Wybór testu: algorytm normalny/uproszczony/losowy

- Wczytanie danych ze zbiorów treningowych
- Utworzenie dwuwymiarowej tablicy użytkownik/film, przechowującej oceny dla każdej z kombinacji
- Utworzenie macierzy dewiacji, przechowującej średnie odchylenia ocen między dwoma filmami
- Wczytanie zbioru testowego
- Wyznaczenie ocen dla danych znajdujących się w zbiorze testowym
- Porównanie ocen i wyznaczenie średniego błędu absolutnego (MAE)

3 Plan eksperymentów

3.1 Pytania

- Jaka będzie przewidywana ocena danego filmu przez danego użytkownika wyznaczona przez algorytm?
- Jaka będzie trafność tej oceny?

3.2 Charakterystyka zbiorów danych

Dane używane do testów pochodzą z serwisu Movielens. Dostępne są dwa zestawy danych: pierwszy składa się z 100,000 ocen 1682 filmów wystawionych przez 943 użytkowników. Drugi zawiera około miliona ocen 3900 filmów wystawionych przez 600 użytkowników. Każdy film został oceniony w skali od 1 do 5.

Do naszych badań zostanie wykorzystany pierwszy zbiór zawierający 100000 rekordów. Został on już wstępnie podzielony na podzbory : trenujący (80000 ocen) oraz testowy (20000).

3.3 Parametry algorytmów których wpływ na wyniki będzie badany

Algorytm nie zawiera parametrów którymi można sterować jego działaniem.

3.4 Sposób oceny jakości modeli

Aby ocenić trafność przewidywań porównamy je z danymi ze zbioru testowego. Do porównania użyjemy miary zwanej Mean Average Error (MAE).

$$MAE = \frac{1}{card(X')} \sum_{u \in X'} \frac{1}{card(S(u))} \sum_{i \in S(u)} |P(u_i) - u_i|$$

Gdzie X' jest zbiorem wszystkich ewaluacji użytkowników, natomiast u zbiorem ewaluacji użytkownika. W wyniku otrzymujemy liczbę będącą średnią średnich różnic pomiędzy predykcją, a właściwą odpowiedzią ze zbioru wszystkich ewaluacji danego użytkownika.

3.5 Testy

Po implementacji algorytmu pierwszym testem było porównanie wyników działania dla prostego zbioru danych z modelem matematycznym zbudowanym w arkuszu kalkulacyjnym.

	A	B	C	D	E
1		Item 1	Item 2	Item 3	
2	User 1	1	2	3	
3	User 2	4	5	1	
4	User 3	2	3	4	
5					
6					
7		Item 1	Item 2	Item 3	
8	Item 1	**0**	**1**	**0,33**	
9	Item 2	**1**	**0**	**0,67**	
10	Item 3	**0,33**	**0,67**	**0**	
11					
12					
13	Predict	Item 1	Item 2	Item 3	
14	User 1	2,44	2,56	2,33	
15	User 2	3,78	3,89	3,67	
16	User 3	3,44	3,56	3,33	
17					

To pozwoliło potwierdzić poprawność implementacji. Następnym krokiem było uruchomienie algorytmu slope-one dla pięciu zbiorów testowych pochodzących z archiwów MovieLens. Przed predykcją algorytm budował na podstawie danych treningowych (różnych dla każdego z pięciu zbiorów) macierz odchyleń ocen. Pierwsze pięć przebiegów zostało wykonanych przy użyciu nieuproszczonej wersji algorytmu. Kolejne pięć, na tych samych danych, zostało wykonane algorytmem ze zoptymalizowanym wzorem predykcji.

Wyniki dla algorytmu nieuproszsczonego:

MAE dla T1 = 0.2353901 MAE dla T2 = 0.2247096 MAE dla T3 = 0.2238898 MAE dla T4 = 0.2321149 MAE dla T5 = 0.2222222

Wyniki dla algorytmu uproszczonego:

MAE dla T1 = 0.2353901 MAE dla T2 = 0.22 MAE dla T3 = 0.22 MAE dla T4 = 0.23 MAE dla T5 = 0.22

Ostatnim krokiem było przeprowadzenie testów dla zbioru losowych predykcji. Zbiór losowy został wygenerowany przy użyciu programu napisanego w środowisku Java (posłużył do tego pakiet java.math). Efektem tego losowania były indeksy w tablic ocen użytkowników. Dzięki temu można było wygenerować losowo oceny filmów zgodnie z rozkładem losowym ocen ze zbioru treningowego. Następnie tak wygenerowane predykcje zostały wprowadzone do algorytmu liczącego MAE. W wyniku tego działania wygenerowane zostały następujące dane:

MAE dla T1 = 0.3105243 MAE dla T2 = 0.3094126 MAE dla T3 = 0.3062914 MAE dla T4 = 0.3078320 MAE dla T5 = 0.3085444

3.6 Wnioski

Z powyższych danych liczbowych o błędzie MAE można wywniskować, że nasza implementacja algorytmu slope-one dała lepsze wyniki niż losowe zgadywanie ocen, które zachowywało ich rozkład. Co więcej uzyskaliśmy także pewne różnice pomiędzy implementacją uproszczoną a pełną algorytmu predykcji. Algorytm pełny, zgodnie z przewidywaniami okazał się nieznacznie lepszy. Testy okazały się niezwykle czasochłonne. Wynikało to głównie z przyjętego sposobu implementacji opartego na pętlach, które to są niestety bardzo niewydajnym narzędziem w obrębie środowiska GNU R.