추천 시스템의 예측 정확도 향상을 위한 전처리 방법

박석인[0] 김택헌 류영석 양성봉
연세대학교 컴퓨터과학과
(psi93[0], kimthun, ryu, yang)@mythos.yonsei.ac.kr

Preprocessing Methods for Improving Prediction Accuracy in Recommender Systems

Seok-In Park[0], Taek-Hun Kim, Young-Suk Ryu, Sung-Bong Yang
Dept. of Computer Science, Yonsei University

요 약

협력적 여과(collaborative filtering) 방법을 사용하는 추천 시스템에서 예측 정확도를 높이는 방법들 중 하나로 군집화(clustering) 방법이 있다. 군집화 방법은 선호도가 유사한 사용자들을 미리 같은 군집으로 만들고, 군집 내에 속한 사용자들을 이웃으로 선정하여 예측을 수행하기 때문에 군집화의 결과가 예측의 정확도에 직접적인 영향을 주게 된다. 본 연구에서는 군집화 결과의 향상을 위해 데이터를 전 처리하는 두 가지 방법과 군집화의 특성을 이용한 새로운 예측식을 제안하고, 기존 연구 방법과의 비교 실험을 통해 실험 결과를 분석한다.

1. 서론

협력적 여과 방법[1]을 사용하는 추천 시스템은 사용자 사이의 선호도를 구하고, 이를 이용하여 특정 아이템을 추천하거나 예측된 선호도 값을 제시한다.

협력적 여과 방법의 예측 정확도를 높이는 방법에는 k-Nearest Neighbor(k-NN) 방법[2]과 군집화 방법 등이 있다. 군집화 방법으로는 k-Means 알고리즘[3]이 주로 사용되고 있다. 두 가지 방법은 모두 선호도가 비슷한 사용자들을 이웃으로 선정하고, 그 이웃들을 기반으로 선호도를 예측하는 방법으로, k-Means 알고리즘을 사용한 방법이 k-NN 방법보다 예측 정확도가 더 높다.

k-Means 알고리즘은 각 사용자의 선호도 집합을 다차원 공간 상의 점으로 간주하고, 그 점들의 거리를 계산함으로써 군집화를 수행한다. 그러나, 단순히 거리를 계산하는 특성 때문에 두 사용자의 선호도가 유사해도 같은 군집 안에 포함되지 않을 수 있는 단점을 갖게 된다.

본 논문에서는 이런 단점을 보안하기 위해 사용자의 선호도 값들을 먼저 등수(rank)나 편차(deviation)로 치환한 후, 그 값들을 군집화에 이용함으로써 선호도가 유사한 사용자들이 같은 군집 내에 들 수 있도록 하는 방법을 제안한다.

그리고, 기존 협력적 여과 방법에서 사용하는 예측식은 사용자간의 유사도를 반영하여 값을 계산하는데, 군집화를 이용할 경우, 이미 각 군집이 유사한 선호도를 갖는 사용자들로 분류되기 때문에 유사도에 대한 중복성을 갖는다. 본 논문에서는 이러한 경우, 예측식에서 유사도를 반영하지 않아도 유사한 기능을 수행할 것으로 보고, 새로운 예측식을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서 추천시스템에 관련된 기존 연구들에 대해 살펴보고, 3장에서 군집화 방법의 전처리 단계 및 새로운 예측식에 대해 본 논문에서 제안한 방법을 기술한다. 5장에서 제안한 방법들에 대한 실험 결과를 분석하고, 마지막으로 6장에서 결론을 맺는다.

2. 관련연구

2.1. 협력적 여과

협력적 여과 방법은 아이템에 대한 사용자의 선호도 집합을 통해 다른 사용자와의 유사성을 계산하고, 이를 기반으로 새로운 아이템을 추천하는 방법이다.

사용자 a와 사용자 k의 유사도 $w_{a,k}$의 계산은 아래 식 (1)의 Pearson 상관관계 계수식을 통해 계산된다. 유사도 값은 - 1.0 에서 1.0 사이의 값을 가지며, 큰 값일수록 유사도가 높음을 의미한다. 계산된 유사도를 식 (2)에 적용하여 새로운 아이템 i에 대한 선호도를 예측한다.

$$w_{a,k} = \frac{\sum_i (r_{a,i} - \overline{r_a})(r_{k,i} - \overline{r_k})}{\sqrt{\sum_i (r_{a,i} - \overline{r_a})^2 \times \sum_i (r_{k,i} - \overline{r_k})^2}} \quad (1)$$

$$P_{a,i} = \overline{r_a} + \frac{\sum_k \{(r_{k,i} - \overline{r_k}) \times w_{a,k}\}}{\sum_k |w_{a,k}|} \quad (2)$$

여기서, j는 사용자 a와 k가 동시에 선호도 점수를 준 아이템을 의미하고, $r_{a,j}$는 사용자 a의 아이템 j에 대한 선호도를 말하고, $\overline{r_a}$는 사용자 a의 평균 선호도를 나타낸다.

2.2. k-Nearest Neighbor

이 방법은 식 (1)을 사용하여 사용자 a와 유사도가 가장 높은 k 명을 정하고, 선정된 k 명만을 이웃으로 인정하여 위의 식 (2)를 통해 선호도를 예측하는 방법이다.

2.3. k-Means 알고리즘

k-Means 알고리즘은 거리 기반 군집화 방법으로 사용자의 선호도를 다차원 공간상의 점으로 표시하고, 거리를 계산함으로써 전체 사용자들의 집합을 k개의 군집으로 나눈다. 사용자 a와 k사이의 거리는 식 (3)과 같이 계산하고, 식에서 a_i는 사용자 a의 속성(차원) i에 대한 선호도 값을 의미한다.

$$d_{a,k} = \sqrt{\sum_i (a_i - k_i)^2} \qquad (3)$$

군집화의 결과 선호도가 비슷한 사용자들이 같은 군집에 속하게 된다. 추천 시스템에서는 사용자와 유사도가 가장 높은 군집을 이웃으로 인정하여 예측함으로써 정확도를 높일 수 있다[4].

3. 전처리 방법과 새로운 예측식

3.1. 전처리 방법

추천 시스템에 적용되는 k-Means 군집화 알고리즘은 거리 기반 군집화 방법의 특성 때문에 표 1과 같이 두 사용자간의 선호도가 유사해도 둘 간의 거리의 크기가 크기 때문에 두 사용자는 다른 군집에 속하게 된다. 표 1에서 두 사용자 A, B의 유사도를 식 (1)을 통해 계산하면 1.0 이고, 이 것은 두 사용자의 선호도가 일치함을 의미한다.

표 1. 사용자 A, B의 선호도

사용자	속성별 선호도			평균
A	0.2	0.4	0.3	0.3
B	0.8	1.0	0.9	0.9

이런 문제점은 본 연구에서 제안하는 등수 방법(RANK)과 편차 방법(DEV)을 사용하여 사용자의 정보를 전처리 함으로써 보완 할 수 있다.

RANK 방법은 단어에서 의미하는 대로 사용자의 선호도 값들을 순위로 변환하는 방법이고, DEV 방법은 해당 속성의 선호도와 전체 선호도에 대한 평균값의 차이로 변환하는 방법이다.

표 2와 표 3는 표 1을 각각 RANK 방법과 DEV 방법으로 전처리 한 후의 두 사용자의 정보 집합(선호도)이 서로 일치하게 조정됨을 보여준다. 결과적으로 전처리 방법을 거친 후, 군집화를 수행하게 되면 두 사용자는 하나의 군집에 속할 수 있게 된다.

표 2. 전처리: RANK를 이용한 선호도 변환

사용자	속성별 선호도			평균
A	3	1	2	0.3
B	3	1	2	0.9

표 3. 전처리: DEV를 이용한 선호도 변환

사용자	속성별 선호도			평균
A	-0.1	0.1	0.0	0.3
B	-0.1	0.1	0.0	0.9

3.2. 새로운 예측식

군집화를 이용한 추천 시스템에서는 사용자들에 대한 선호도 집합을 사용하여 유사도가 높은 사용자의 군집들로 나눈 후, 테스트 사용자의 해당 군집을 이웃으로 하여 예측 및 추천을 하게 된다. 이 때, 기존 추천 시스템에서 사용하는 예측식은 식 (1)과 같이 사용자 사이의 유사도를 반영하여 예측을 수행하게 되는데, 군집화와 예측 시에 유사도 반영이 중복적으로 발생 함을 알 수 있다. 따라서, 예측 시 유사도를 반영하지 않아도 기존 예측식과 유사한 결과를 보일 것으로 기대된다. 이런 가정을 근거로 추천 시스템에서의 선호도 예측식을 새롭게 제안한다.

식 (4)는 예측 아이템에 대해 군집내의 이웃들이 점수를 준 선호도 값의 평균을 의미한다. 여기서 n은 예측아이템 i에 점수를 준 이웃들의 수를 나타낸다.

$$P_{a,i} = \frac{\sum_k r_{k,i}}{n} \qquad (4)$$

식 (5)는 사용자 a의 평균 선호도와 모든 이웃들간의 예측아이템에 대한 편차(deviation)의 합을 이용하여 예측을 수행한다. 이 식은 식 (1)에서 사용자간의 유사도 부분을 제외한 간략화 된 형태로 되어있다.

$$P_{a,i} = \overline{r_a} + \frac{\sum_k (r_{k,i} - \overline{r_k})}{n} \qquad (5)$$

4. 실험 및 결과

4.1 실험 환경 및 데이터

본 논문의 실험에는 1997년 Digital Equipment Corporation에서 공개한 EachMovie[5] 데이터 셋을 사용하였다. 이 것은 총 72916명의 사용자가 1628개의 각 영화에 대해 평가한 선호도로 구성되어 있으며, 선호도는

0.0 부터 1.0 사이의 값으로 0.2의 차이를 두고 명시적으로 평가되어 있다. 그리고, 영화의 장르는 액션, 애니메이션, 코미디 등 총 10 가지로 구분되어 있다.

실험에는 전체 데이터 셋 중에서 최소한 100회 이상, 각 장르 당 최소한 1회 이상 선호도를 입력한 사용자 3763명을 선택하였고, 무작위로 선택한 188명 (5%)을 테스트 사용자로, 나머지 3575명 (95%)을 훈련용으로 구분하여 사용하였다. 그리고, 테스트 사용자가 선호도를 평가한 영화들 중 무작위로 5개를 선택하여, 선호도를 예측하고 실제 값과 비교하였다. 이와 같은 방법으로 3회 반복 수행하여 각 결과의 평균 값을 실험 결과로 하였다.

4.2 성능 평가 기준

예측의 정확성을 평가하기 위해서 MAE (Mean Absolute Error) 방법을 사용하였다. 식 (6)는 MAE를 나타낸 것이며, N은 총 예측 횟수, ε는 예측된 선호도와 실제 선호도 간의 오차를 나타낸다.

$$|\overline{E}| = \frac{\sum_{i=0}^{N} |\varepsilon_i|}{N} \qquad (6)$$

4.3. 실험 결과

기존 방법과 실험 결과를 비교하기 위해 GroupLens (CF) 방법, k-Nearest Neighbor 방법(kNN)과 k-Means 방법을 통해 군집화 한 후의 협력적 여과 방법(kCF)을 실험하였다.

전처리 방법은 RANK 방법과 DEV 방법을 이용한 방법을 실험하여 기존 방법과 예측 정확도를 비교하였고, 새로운 예측식, 식 (4)(P_AVG 방법)와 식 (5) (P_MDEV 방법)에 대한 실험은 kCF 와 실험 결과를 비교하였다.

k-Means 방법을 사용한 실험에서는 최적의 k 값을 찾기 위해 10에서 30까지 모든 k 값에 대하여 실험하였다.

표 4는 기존 연구 방법과 전처리 방법을 통한 kCF방법에 대한 실험 결과이다. 여기서 k는 각 실험에서 얻은 가장 최적 k 값을 나타내고, MAE는 그 k값에서의 오차를 보여준다. 표에서 보는 바와 같이 전처리를 거친 방법들이 기존 방법들에 비해 더 낮은 오차 값을 가졌고, 그 중에서 DEV 방법이 가장 좋은 결과를 보였다.

표 4. 전처리 방법에 대한 실험 결과

실험	CF	kNN	kCF	RANK	DEV
k		50	27	15	18
MAE	0.1984	0.1963	0.1914	0.1905	0.1898

그림1은 kCF 방법과 제안된 예측식을 사용한 실험을 그래프로 나타낸 것이다. 세 실험 모두 k 값이 27인 경우에 가장 좋은 결과를 보여주고 있다. 그래프는 k의 최적 값 부근의 결과를 보여준다. P_AVG방법과 P_MDEV방법의 MAE 값은 각각 0.1883와 0.1884로 두 실험이 표 4의 다른 실험들과 비교해서 가장 좋은 결과를 보임을 알 수 있다.

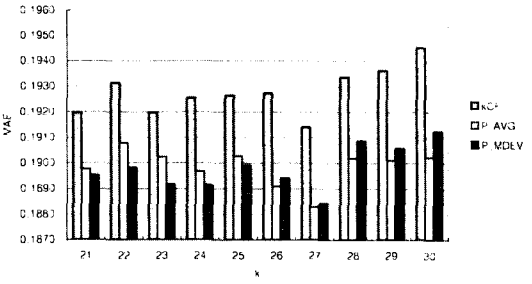

그림 1. 제안한 예측식에 대한 실험 결과

5. 결론

본 논문에서는 추천 시스템에서 k-Means 군집화 방법을 이용할 때, 군집 결과의 향상과 그에 따른 예측 정확도의 향상을 위해 전처리 방법을 제안 하였다. 또한, 같은 군집에 있는 사용자간의 높은 유사성을 이용하여 새로운 예측식을 제시하고 실험하였다.

전처리 방법은 사용자 사이의 유사도를 기존 방법들보다 잘 반영하게 함으로써 예측 오차를 줄였고, 새로운 예측식에 대한 실험은 유사도에 대한 중복 사용을 피함으로써 기존 연구들 보다 향상된 결과를 보였다.

6. 참고 문헌

[1] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, " Analysis of Recommendation Algorithms for E-Commerce," *The ACM E-Commerce 2000 Conference*, 2000.

[2] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, " An Algorithm Framework for Performing Collaborative Filtering," *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, 1999.

[3] Lyle H. Ungar and Dean P. Foster, " Clustering Methods for Collaborative Filtering," *Proceeding of the 1998 Workshop on Recommendation Systmes*, pp.114-129, 1998.

[4] Young-Suk Ryu, Taek-Hun Kim, Ji-Sun Park, Seok-In Park, and Sung-Bong Yang, " Using Content Information for Finding Neighbors in the Collaborative Filtering Framework," *Proceeding International Conference on Electronic Commerce*, 2001.

[5] P. McJones, EachMovie collaborative filtering data set, URL:http://www.research.digital.com/SRC/eachmovie/, 1997.