엔트로피와 Default Voting을 이용한 협력적 필터링에서의 사용자 유사도 측정

조선호* 김진수* 이정현*
*인하대학교 전자계산공학과,
g2001415@inhavision.inha.ac.kr kjspace@nlsun.inha.ac.kr jhlee@inha.ac.kr

User Simility Measurement Using Entropy and Default Voting Prediction in Collaborative Filtering

Sun-Ho Cho* Jin-Su Kim* Jung-Hyun Lee*
*Dept. of Computer Science & Engineering, Inha University

요 약

기존의 인터넷 웹 사이트에서는 사용자의 만족을 극대화시키기 위하여 사용자별로 개인화 된 서비스를 제공하는 협력적 필터링 방식을 적용하고 있다. 협력적 필터링 기술은 사용자의 취향에 맞는 아이템을 예측하여 추천하며, 비슷한 선호도를 가진 다른 사용자들과의 상관관계를 구하기 위하여 일반적으로 피어슨 상관계수를 많이 이용한다. 그러나, 피어슨 상관계수를 이용한 방법은 사용자가 평가를 한 아이템이 있을 때에만 상관관계를 구할 수 있다는 단점과 예측의 정확성이 떨어진다는 단점을 가지고 있다. 따라서, 본 논문에서는 피어슨 상관관계 기반 예측 기법을 보완하여 보다 정확한 사용자 유사도를 구하는 방법을 제안한다. 제안된 방법에서는 사용자들을 대상으로 사용자가 평가를 한 아이템의 선호도를 사용해서 엔트로피를 적용하였고, 사용자가 선호도를 표시하지 않은 상품에 대해서는 Default Voting 방법을 이용하여 보다 정확한 협력적 필터링 방식을 구현하였다.

1. 서론

협력적 추천 시스템에서 가장 중요한 것은 고객의 선호도를 분석하고 정제하여 정확한 예측으로 고객이 원하는 가장 적절한 상품을 추천해 줄 수 있는 능력이다. 이러한 협력적 필터링 방식을 이용하는 추천 시스템에는 FireFly[7], GroupLens[4][6]와 같은 시스템들이 있다. 그러나 이러한 시스템은 피어슨 상관계수 기반 예측 기법의 단점으로 인하여 협력적 필터링 기술에서 한계점을 나타낸다. 피어슨 상관계수 기반 예측 기법의 단점은 다음 세 가지로 요약할 수 있다[1][3]. 첫째, 두 고객 사이의 상관계수는 오직 두 고객 모두 선호도를 표시한 상품에 대해서만 계산된다. 둘째, 비록 두 고객이 선호도에 따른 상관관계가 높지 않더라도 다른 고객의 선호도 예측에 좋은 자료가 될 수 있으나 상관관계가 높지 않다는 이유로 이 정보는 활용되지 못한다. 셋째, 상관관계가 오직 두 고객 사이에서만 계산된다는 것이다.

본 논문에서는 피어슨 상관관계 기반 예측 기법의 단점을 보완하기 위해 사용자가 평가한 아이템에 사용자들이 갖는 평균 정보량을 표현하는 엔트로피를 적용하였고, 사용자가 선호도를 표시하지 않은 상품에 대해서는 Default Voting 예측 방법을 이용하여 보다 정확한 협력적 필터링 방식을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 협력적 필터링 기술에 대한 관련연구를 설명하고 3장에서는 본 논문에서 제시하는 사용자 유사도 측정방법에 대해서 설명한다. 4장에서는 실험을 통한 성능을 분석하며 마지막 5장에서 결론을 맺는다.

2. 관련연구

2.1 협력적 필터링의 피어슨 상관계수

협력적 필터링의 일반적인 형태로 피어슨 상관계수가 사용된다[3]. 피어슨 상관계수는 유사한 선호도를 가지는 이웃(Nearest Neighborhood)들을 정하고 예측 선호도 값을 위하여 다음과 같은 식(1)을 사용한다.

$$p_{a,j} = \overline{v_a} + \frac{\sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v_i})}{\sum_{i=1}^{n} w(a,i)} \qquad \text{식(1)}$$

$p_{a,j}$는 사용자 a의 아이템 j에 대한 선호도를 예측한 값이고, $\overline{v_a}$는 사용자 a의 선호도 평균값이다. $w(a,i)$는 사용자 a와 사용자 i의 유사도 가중치이고, n은 사용자 a와 다른 사용자간의 유사도가 0이 아닌 사용자수이다. 또한 피어슨 상관 계수를 사용했을 경우 사용자 a와 사용자 i의 유사도 가중치 $w(a,i)$는 다음과 같이 정의된다.

$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v_a})(v_{i,j} - \overline{v_i})}{\sqrt{\sum_j (v_{a,j} - \overline{v_a})^2 \sum_j (v_{i,j} - \overline{v_i})^2}} \qquad \text{식(2)}$$

$v_{a,j}$는 사용자 a가 아이템 j에 대해서 보여준 선호도이고, $\overline{v_a}$는 사용자 a가 선호도를 입력한 아이템들에 대한 선호도 평균값이다. j는 사용자 a와 i가 공통으로 선호도를 입력한 아이템들이다.

2.2 Default Voting 방법

Default Voting[2]은 Correlation기반의 알고리즘을 확장한 것으로 기준이 되는 특정 사용자와 그 사용자와의 유사도가 있는 사용자들이 공통으로 선호도를 보이는 아이템이 상대적으로 적을 경우에 사용한다. Default Voting은 더 나아가 사용자 중 어떤 사람도 선호도를 입력하지 않은 새로운 아이템에 대해서 기본값을 우선적으로 적용함으로써 추천이 가능하도록 할 수 있다. 대부분의 경우 Default Voting값 d는 중립적이거나 다소간 비 선호의 값을 사용하는 경우가 많다. Default Voting값은 사용자 a와 사용자 i의 평균 선호도나 전체 사용자의 평균 선호도를 사용할 수 있고 암묵적 선호도 데이터의 경우 웹 페이지의 방문 여부나 어떤 제품의 구매 여부 등과 같이 방문이나 구매를 1로 볼 수 있는 경우 웹 페이지의 방문 횟수나 제품군의 제품 구매 횟수 등에서도 마찬가지로 Default Voting 값은 방문 횟수나 제품군의 구매 횟수를 0으로 설정할 수 있다. Default Voting은 사용자에 의해서 선호도가 입력되지 않은 아이템에 대해서 보완점을 가질 수 있지만 사용자 a와 사용자 i가 공통으로 구매한 아이템의 수가 일정한 기준을 넘는 경우에 사용하는 것이 좋다.

2.3 확률 벡터의 정보량

확률 벡터의 정보량은 섀논의 정보이론에 근거하여 계산하였다. 섀논은 불확실성의 크기를 엔트로피로 측정하였는데, 이것이 사용

자가 갖는 평균 정보량이 된다[8]. 식(3)은 평균 정보량을 구하기 위한 식이다.

$$E(p) = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad \text{식(3)}$$

사용자 i가 갖는 정보량 $\log_2 p_i$는 선택될 확률벡터 p_i에 의해 결정된다. 따라서, 식(3)에서 $E(p)$는 n개의 사용자가 갖는 평균 정보량이며, p_i는 i번째 아이템이 선택될 확률을 나타낸다. $E(p)$는 다음과 같은 특성을 갖는다.

(i) $0 \le E(p) \le \log_2 N$

(ii) $E(p) = \log_2 N$, if $p_1 = p_2 = \dots = p_n = 1/N$

(iii) $E(p) = 0$, if $p_i = 1$, $p_i = 0$ $(1 \le j \le N, j \ne i)$ 식(4)

3. 사용자 유사도 측정방법

사용자 유사도를 계산하기 위해 아이템들이 갖는 평균 정보량인 엔트로피를 적용하는 과정에서의 확률벡터 연산과정은 다음과 같다.

3.1 비 선호도에 대한 Default Voting의 적용

기존의 협력적 필터링에서 많이 사용되는 피어슨 상관계수의 문제점은 사용자가 선호도를 모두 표시해야만 한다는 것이다. 이를 해결하기 위해 Default Voting 방법을 적용한다. 사용자 선호도를 표시한 아이템에 대해서 공통으로 표시한 아이템이 아닌 사용자가 어느 하나라도 선호도를 아이템에 대해서 표시하지 않았다면 표시하지 않은 아이템에 대한 선호도 값은 식(5)와 같이 정의할 수 있다.

$$d = \frac{S_i}{n} \qquad \because S_i = (p_{ui,1} + p_{ui,2} + p_{ui,n}) \qquad \text{식(5)}$$

식(5)에서 n은 사용자가 선호도를 보인 아이템의 개수이며 S_i는 사용자 i에 대한 아이템들의 합이다. 식(5)에 의해 d 값이 구해지면 Default Voting에 적용함으로써 사용자가 선호도를 표시하지 않은 아이템에 대해서 값을 표시하고 예측 선호도 값의 정확도를 높일 수 있다. 사용자가 선호도를 표시하지 않으면, 활용되지 않았던 기존의 피어슨 상관계수 방식에서 Default Voting의 d값을 구함으로써 선호도 값을 표시하지 않았던 아이템의 정보까지 활용하였다.

3.2 확률벡터 연산 과정(Entropy)

사용자가 선호도를 보인 아이템들의 선호도 값을 확률 벡터로 표현할 수 있다. 사용자 i의 확률벡터 U_i는 다음과 같이 표현된다.

$$U_i = (P_{U_{i,1}}, P_{U_{i,2}}, \dots, P_{U_{i,j}}) \qquad \text{식(6)}$$

$P_{U_{i,j}}$는 사용자 i가 평가를 한 모든 선호도의 값에 대한 아이템 j의 선호도의 비율이다. 평균 정보량을 계산하기 전에 우선, 아이템들의 유한 집합 $T=(t_1, t_2, \dots, t_m)$이 주어졌을 때, 다음과 같은 확률함수를 갖는다.

$$p_i = P(u_i|t), \text{ for } i = 1,2,\dots,N,\ p_i \ge 0,\ \sum_{i=1}^{n} P_i = 1 \qquad \text{식(7)}$$

이러한 경우 t에 대한 엔트로피는 다음과 같다.

$$E(P_t) = E(p_1, p_2, \dots, p_N) = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad \text{식(8)}$$

이와 같은 방법으로 측정된 $E(P_t)$값은 확률벡터들과의 곱으로서 다음과 같은 식(9)가 성립되고 새로운 테이블을 생성할 수 있다. 식(9)는 사용자 a가 아이템 j에 대해 평가한 선호도의 확률벡터($P_{a,j}$)와 아이템에 대한 엔트로피($E(P_t)$)를 곱한 값 $H(a,j)$를 계산한다. $P_{a,j}$는 사용자 a에 대한 아이템 j의 확률벡터 값이다.

$$H(a,j) = E(P_t) * P_{a,j} \qquad \text{식(9)}$$

3.3 피어슨 상관계수와 Entropy

기존의 협력적 필터링의 피어슨 상관계수는 특정 아이템에 대한 이웃들의 선호도와 이웃들의 선호도 평균과의 거리를 이웃들과의 유사도로 가중 평균함으로써 특정 사용자의 아이템에 대한 선호도를 예측하는 것이다. 모든 사용자의 선호도 값은 근사적으로 같은 분포를 가지며 같은 분산을 가진다고 가정하지만 현실적으로 같은 분포를 가지며 동일한 분산을 가질 것이라는 확신을 할 수 없다. 따라서 이 사용자 선호도 값을 식(7)을 이용하여 확률 벡터로 표현하고, 이 사용자 확률벡터를 식(8)을 이용하여 평균 정보량을 구하여 식(9)에 의해 기존의 테이블을 새로운 테이블로 다시 표현을 할 수 있다. 따라서 아래와 같은 식으로 새로운 사용자 유사도 가중치 계산식을 정의할 수 있다.

$$w(a,i) = \frac{\sum_j (H_{a,i} - \overline{H_a})(H_{i,j} - \overline{H_i})}{\sqrt{\sum_j (H_{a,i} - \overline{H_a})^2 \sum_j (H_{i,j} - \overline{H_i})^2}} \qquad \text{식(10)}$$

$H_{a,i}$ 는 사용자 a가 아이템 i에 대해서 보여준 선호도이고, $\overline{H_a}$는 사용자 a가 선호도를 입력한 아이템들에 대한 선호도 평균값이다. j는 사용자 a와 i가 공통으로 선호도를 입력한 아이템들이다.

3.4 사용자 유사도 측정 알고리즘

[알고리즘 1]은 Default Voting과 엔트로피에 의한 사용자 유사도 측정 알고리즘이다.

[알고리즘 1] Default Voting과 엔트로피에 의한 사용자 유사도 측정 알고리즘

```
M; // 한 사용자에 대한 아이템의 개수
N; // 한 아이템에 대한 사용자의 수
S ← 0; // 선호도 값의 합
E ← 0; // 엔트로피      EN = 0; // 빈 아이템의 개수
/* Default Voting 값 부여 */
for(i=1; i<=N; i++) {
   for(j=1; j<=M; j++) {
      if(P[Ui][j]==0)   EN ← EN + 1; // 비 선호도 값 체크
      else      S ← S + P[Ui][j]; } // 표시된 선호도의 합
   d ← S / (N - EN); }
/* Entropy 값 측정 */
for(j=1; j<=M; j++)
   for(i=1; i<=N; i++)
      Ej ← Ej + (-(P[Ui][j]*log2P[Ui][j])); // 가중치 값
/* 새로운 테이블 생성 */
for(j=1; j<=M; j++)
   for(i=1; i<=N; i++)
      NewTable[Ui][j] ← Ej * P[Ui][j];
/* 사용자 a와 사용자 i의 유사도 측정 */
Ha // 사용자 a의 가중치가 부여된 선호도의 평균값
Hi // 사용자 j의 가중치가 부여된 선호도의 평균값
Pcc1 = Pcc2 = Pcc3 ← 0;
while(k∈Ua && k∈Ui) { // 사용자 a와 사용자 i가 공통으로
                       표시되어 있는 아이템 k
   Va,k // 사용자 a가 아이템 k에 대해서 보여준 선호도
   Vi,k // 사용자 i가 아이템 k에 대해서 보여준 선호도
   Ha,k ← P(Ej) * Va,k;
   Hi,k ← P(Ej) * Vi,k;
   Pcc1 ← Pcc1 + (Ha,k - Ha)(Hi,k - Hi);
   Pcc2 ← Pcc2 + (Ha,k - Ha)^2;
   Pcc3 ← Pcc3 + (Hi,k - Hi)^2;
   k++; }
Ha,i ← Pcc1 / sqrt(Pcc2 * Pcc3); // 사용자 a와 사용자 i의 유사도
Assign ( Ha,i);
```

4. 실험 및 결과

4.1 데이터 집합

본 논문의 실험을 위한 구현환경은 Visual C++ 6.0과 Pentium II 333MHz, 256MB Ram 환경에서 실험을 실시하였다. 실험을 위한 데이터로는 EachMovie 데이터 집합을 사용하였는데 이 데이터 집합은 컴팩 연구소에서 18개월 동안 협력적 필터링 알고리즘을 연구하기 위해서 영화에 대한 사용자의 선호도를 조사한 데이터이다[5]. 데이터 중에서 400명을 샘플링하여 Nearest Neighborhood방식

[4]을 이용해서 10개의 이웃들로 나누었다.

4.2.성능 평가 기준

본 논문에서는 예측값의 정확성 측면에서 성능을 평가하기 위해서 MAE, Weighted mean 평가방법[2]을 사용하였다.

[1]Mean Absolute Error(MAE)

Error는 실제 선호도 값과 예측된 선호도 값과의 차이로 정의되고 MAE는 Error의 절대값들의 평균을 의미한다. MAE는 절대적으로 알고리즘이 얼마나 정확하게 예측을 했는지를 알 수 있다.

[2] Weighted mean

Weighted mean은 실제 사용자의 선호도와 예측된 선호도와의 차이값인 Error에 실제 사용자의 선호도에서 사용자의 평균 선호도를 뺀 값을 곱함으로써 사용자의 평균 선호도에서 멀리 떨어진 아이템에 대해서 얼마나 잘 예측을 하는지를 보여주는 척도이다.

4.3 실험방법 및 결과

EachMovie 데이터 중에서 400명을 샘플링하여 Nearest Neighborhood 방식을 이용해서 10개의 이웃들로 나누었다. 10개의 이웃들 중 하나인 데이터만을 추출하여 식(5)에 의해 Default Voting 값을 부여하고 식(8)에 의해 엔트로피를 계산한 결과는 [표1]과 같다. 흑색으로 표시된 부분이 Default Voting 부여값이다.

[표 1] Default Voting과 엔트로피 계산 값

사용자 / item	Nearest Neighborhood 의한 이웃선정							Entropy
	2	7	15	49	23	51	···	
3	0.025	0.1	0.05	0.05	0.075	0.025	···	1.836
11	0.125	0.05	0.1	0.1	0.075	0.05	···	1.934
29	0.075	0.075	0.075	0.075	0.1	0.1	···	2
35	0.05	0.125	0.075	0.125	0.05	0.05	···	1.908
43	0.1	0.025	0.075	0.025	0.05	0.025	···	1.7
:	:	:	:	:	:	:		:

[표2]는 식(9)를 이용하여 각각의 확률벡터 값과 엔트로피를 곱한 결과값으로 갱신된 테이블이다.

[표 2] 확률벡터와 엔트로피에 의한 테이블

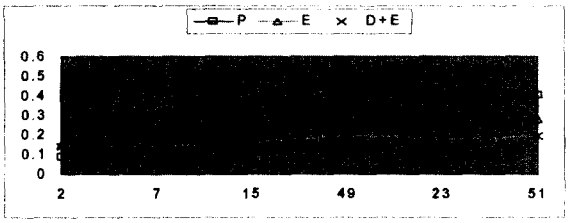
사용자 / item	Nearest Neighborhood 의한 이웃선정						
	2	7	15	49	23	51	···
3	0.046	0.184	0.092	0.092	0.138	0.046	···
11	0.242	0.097	0.193	0.193	0.145	0.097	···
29	0.15	0.15	0.15	0.15	0.2	0.2	···
35	0.095	0.239	0.143	0.239	0.095	0.095	···
43	0.17	0.043	0.128	0.043	0.085	0.043	···
:	:	:	:	:	:	:	

[표3]은 성능평가를 위하여 MAE와 Weighted mean의 방법으로 비교 평가한 결과이다. 피어슨 상관계수의 예측값과 엔트로피만 사용했을 때의 예측값, 그리고 Default Voting 후에 엔트로피를 적용했을 때의 성능비교를 평가한 예측값에 대한 평균값이다. Default Voting 후에 엔트로피를 적용했을 때 예측값이 가장 좋음을 보여주고 있다.

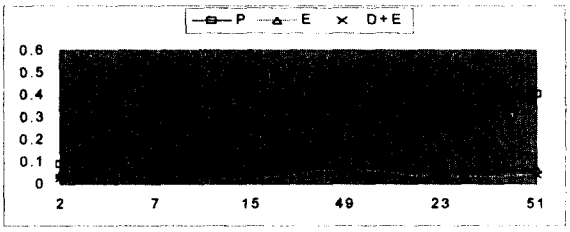
[표 3] 사용자별 MAE / Weighted mean의 평균값

MAE / Weighted mean			
사용자	피어슨 상관계수(P)	Entropy(E)	Default Voting+Entropy(D+E)
2	0.089	0.146 / 0.044	0.141 / 0.03
7	0.228	0.104 / 0.021	0.143 / 0.029
15	0.308	0.223 / 0.048	0.164 / 0.026
49	0.497	0.359 / 0.079	0.197 / 0.067
23	0.378	0.256 / 0.051	0.181 / 0.033
51	0.403	0.277 / 0.061	0.196 / 0.042

[그림1]과 [그림2]는 MAE와 Weighted mean 성능평가를 한 결과값을 그래프로 표현한 것이다.



[그림 1] MAE에 의한 성능비교



[그림 2] Weighted mean에 의한 성능비교

5. 결론

본 논문에서는 아이템을 대상으로 사용자가 선호도를 보이지 않은 값에 대하여 Default Voting을 부여하고 확률벡터로 표현하여 엔트로피를 적용하여 확률벡터로 표현된 새로운 테이블을 생성하여 새로운 사용자 유사도 측정법을 제안하였다. 제안한 방법에 대하여 기존의 협력적 필터링 기술과 성능을 비교한 결과 예측의 정확도가 향상되었기에 본 논문에서 제안한 방식이 좀 더 정확한 예측을 보여준다는 것을 알 수 있었다. 향후에는 Default Voting 값을 사용자에 근접한 선호도를 표시할 수 있는 방법을 연구하여야 할 것이며, 엔트로피를 이용해서 사용자 로그를 분석한 규칙기반에 적용하면 보다 정확한 예측값을 얻어낼 수 있을 것이다. 또한 예측의 결과를 보다 빠르게 할 수 있는 방법에 대하여 연구해야 할 것이다.

참고 문헌

[1] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Analysis of recommendation Algorithms for E-Commerce," Proc. of the ACM E-Commerce, 2000.

[2] Bress, J., Heckerman, D., and Kadie, C., "Empirical Analysis of Predictive Algorithms for collaborative filtering," Proceedings of the fourteenth Annual Conference on Uncertainty in Artificial Intelligence, 1998.

[3] Diniel Billsus and Michael J. Pazzani, "Learning collaborative Informaton Filters," Proceedings of ICML, pp.46-53, 1998.

[4] J. Konstan, et al., "GroupLens: appling collaborative filtering to usenet news," Communications of the ACM Vol. 40, No. 3, 1997.

[5] P. McJones, EachMovie collaborative filtering datasets, url : www.research.digital.com/SRC/eachmoive, 1997.

[6] Sarwar, B. M., et al., "Using Filtering Agents to Improve Prediction Quality in the GroupLens research Collaborative Filtering System," Proc. ACM CSCW, pp. 345-354, 1998.

[7] U. Shardanand and P. maes, "Social information filtering: algorithm for automating 'word mouth'," Proc. of ACM CHI Conference, 1995.

[8] 강창언, 오용선, 이명호, 정보이론 - 토딩 이론과의 접목, 생능사, pp.233-285, 1987.