협력적 여과 시스템에서 산포도를 이용한 잡음 감소

고 수 정†

요 약

협력적 여과 시스템의 사용자-아이템 행렬은 사용자들이 아이템에 대하여 평가할 경우 사용자들의 감정 상태가 일정하지 않음으로 인하여 평가 결과에 잡음을 포함할 가능성이 높다. 이러한 문제점을 해결하기 위해 본 논문에서는 산포도를 이용하여 추천 정보로서 이용하기에 부적당한 평가값들을 제외시킴으로써 사용자-아이템 행렬을 최적화시키고, 아이템 정보와 사용자 정보를 반영하여 고유의 사용자의 평가값을 기반으로 선호도를 예측하였을 때 발생하는 잡음을 감소시킨다. 산포도의 변이계수가 갖는 단점을 보완하기 위하여 백분위수를 이용하여 극한적인 평가값을 제거하고, 사용자의 변이계수와 아이템의 중위수를 병합하여 가중치가 부여된 사용자-아이템 행렬을 구성한다. 마지막으로 이를 기반으로 새로운 사용자의 선호도를 예측한다. 제안된 방법은 영화에 대해 평가한 MovieLens 시스템의 데이터베이스를 이용하여 평가되었으며, 기존의 방법보다 성능이 높음을 보인다.

키워드 : 협력적 여과 시스템, 산포도, 잡음 감소

Reducing Noise Using Degree of Scattering in Collaborative Filtering System

Su-Jeong Ko†

ABSTRACT

Collaborative filtering systems have problems when users rate items and the rated results depend on their feelings, as there is a possibility that the results include noise. The method proposed in this paper optimizes the matrix by excluding irrelevant ratings as information for recommendations from a user-item matrix using dispersion. It reduces the noise that results from predicting preferences based on original user ratings by inflecting the information for items and users on the matrix. The method excludes the ratings values of the utmost limits using a percentile to supply the defects of coefficient of variance and composes a weighted user-item matrix by combining the user coefficient of variance with the median of ratings for items. Finally, the preferences of the active user are predicted based on the weighted matrix. A large database of user ratings for movies from the MovieLens recommender system is used, and the performance is evaluated. The proposed method is shown to outperform earlier methods significantly.

Key Words : Collaborative Filtering System, Degree Of Scattering, Noise Reduction

1. 서 론

사용자들은 종종 음악, 영화, 서적 등 새로운 아이템을 찾기 위하여 가족, 친구, 동료들이 전반적으로 흥미롭거나 유용하게 판단하였던 정보를 이용한다[6]. 협력적 여과 시스템은 이와 같은 원리를 기반으로 아이템 추천의 타당성과 질에 대한 사용자들의 의견을 공유하는 개인화 추천 시스템이다[1]. 협력적 여과 기반 추천 시스템에서 입력 데이터는 m명의 사용자와 n개의 아이템에 관한 상품 거래 내역에 관한 정보를 나타낸다[5]. 이러한 정보는 통상적으로 m x n의 사용자-아이템 행렬로써 표현한다[14].

m x n의 사용자-아이템 행렬 표현이 매우 단순해 보일지라도 이를 추천에 이용한다는 것은 단순한 문제가 아니다. 행렬에 속한 모든 사용자가 모든 아이템에 대하여 평가를 할 경우에만 이 행렬이 완성되나 실제로 이러한 경우는 드물기 때문이다. 이와 같은 희박성을 해결하기 위해 최근 확률적 이론을 이용하여 결측치를 예측하는 방법[19], 활용 그래프를 이용하는 방법[12], 그리고 신뢰를 감지하는 협력적 여과 방법[7] 등이 제안되었다. 또한, 사용자들은 단지 그들이 흥미를 느끼는 아이템에 대해서만 평가를 하기도 하며, 또한 어떤 사용자는 본인이 흥미를 느끼지 않는 아이템에 대해서만 평가를 하는 경우가 발생하기도 한다. 이러한 경우, 사용자가 아이템에 대하여 평가를 하지 않는 경우에 발

※ 이 연구는 인덕대학 학술연구비 일부 지원에 의하여 수행되었음.
† 정 회 원 : 인덕대학 컴퓨터소프트웨어과 교수
논문접수 : 2007년 4월 23일, 심사완료 : 2007년 10월 14일

생하는 결측치로 인하여 사용자-아이템 행렬은 희박성을 나타낸다.

또 다른 문제점은 아이템들에 대하여 평가를 해야 하는 사용자들의 입장에서 시간이 없거나 평가에 대하여 지루함을 느끼고 있을 때, 아이템에 대하여 올바른 평가를 한다는 것은 거의 불가능하다. 이러한 데이터는 사용자의 흥미를 정확하게 반영했다고 할 수 없으므로 이러한 자료를 이용하여 다른 사용자에게 추천을 제공하는 것은 추천의 정확도를 저하시키는 일이다. 따라서 사용자-아이템 행렬에서 이러한 데이터를 제외시킴으로써 잡음을 최대한으로 감소시키는 과정이 필요하다.

본 논문에서는 사용자-아이템 행렬의 고유 평가값이 갖는 잡음을 감소시키기 위하여 사용자와 아이템 정보를 고유 평가값에 반영하는 방법을 제안한다. 사용자 정보와 아이템 정보를 반영하는 방법으로는 선호도의 분포 정도를 표현하는 산포도[18]를 이용한다. 산포도란 관련된 자료가 중심위치 즉, 대표값으로부터 어느 정도 떨어져 있는가를 나타내는 측도로써 집단의 분포 특성을 파악할 수 있다.

산포도는 절대적 산포도와 상대적 산포도로서 구분할 수 있다. 절대적 산포도는 평균이 같고 자료의 구조가 동질적인 집단을 비교하는 데 사용하며, 그 종류로는 범위, 사분편차, 평균편차, 분산 및 표준편차 등이 있다. 상대적 산포도는 평균이 다르거나 자료의 구조가 이질적인 집단을 비교하는 데 사용된다. 상대적 산포도는 절대적 산포도에 대한 대표값의 비로서 식 (1)과 같이 나타낸다[20].

$$\text{상대적산포도} = \frac{\text{절대적 산포도}}{\text{대표값}} \quad (1)$$

상대적 산포도를 이용한 방법은 범위계수, 사분편차계수, 평균편차계수, 변이계수 등이 있다. 본 논문에서 제안한 방법에서는 평균이 다르고 자료의 구조가 이질적이므로 상대적 산포도를 이용한다. <표 1>은 상대적 산포도를 이용한 방법들의 개요와 단점을 기술한다.

산포도를 측정하는 가장 좋은 방법은 각 데이터 점들에 대한 평균을 내기 전에 그 차를 제곱하는 방법이다. 이러한 산포도의 측정은 분산으로써 알려져 있으며, 그 분산에 대한 제곱의 거듭제곱은 표준편차로 알려져 있다. 표준편차와 분산은 산포도를 측정하는 데 광범위하게 사용되어 왔다. <표 1>에서 이와 같이 표준편차와 분산을 이용한 방법은 변이계수다.

따라서 본 논문에서 제안한 방법에서는 산포도를 측정하기 위하여 가장 광범위하게 사용되고 있는 변이계수를 이용한 방법을 사용한다. 변이계수를 이용한 산포도 측정 방법은 <표 1>에서와 같이 표준편차를 이용함으로 인하여 극단값의 영향을 크게 받는다. 이를 그대로 사용자의 분포를 파악하는 산포도로서 이용하였을 경우 많은 오차를 포함한다. 따라서 백분위수를 이용하여 변이계수를 이용한 방법의 단점을 보완한다.

〈표 1〉 상대적 산포도의 종류와 개요

	개요	단점
범위 계수	- 데이터 집합에 나타난 값의 최대값과 최소값의 차를 이용	최대값과 최소값에 대한 정보만을 제공할 뿐이고, 이 범위 안에 속한 값들에 대한 일체의 정보도 나타낼 수 없음
사분 편차 계수	- 자료 중의 극단값에 의해 영향을 받지 않음	자료의 수가 많아지면 정확도가 낮아짐
평균 편차 계수	- 각 데이터 점과 평균 사이의 차에 대한 평균을 계산하고, 이 값을 평균 편차 계산을 위한 각 점의 수로 나누는 방법을 이용 - 계산이 어렵지 않음	연속적인 통계분석에서는 그 수식이 매우 복잡함
변이계수	- 각 데이터 점들에 대한 평균을 내기 전에 그 차를 제곱하는 방법을 이용한 방법 - 평균과 표준편차를 대상으로 다양한 실험을 거쳐 테스트 하였을 경우, 데이터 값들의 변동 정도를 나타냄 - 표준편차를 산술 평균으로 나눈값으로 정의됨	자료의 극단값을 더욱 중요시하여 극단값의 영향을 매우 많이 받음.

백분위수를 이용하여 사용자-아이템 행렬의 잡음을 제거하고, 이를 대상으로 사용자의 변이계수와 아이템 중위수를 계산한다. 다음으로, 사용자의 변이계수와 아이템의 중위수를 병합함으로써 가중치가 부여된 사용자-아이템 행렬을 생성함으로써 사용자가 평가한 정보에 사용자와 아이템의 정보를 반영한다. 마지막으로 가중치가 부여된 행렬을 기반으로 새로운 사용자의 선호도를 예측한다.

제안된 방법은 사용자가 영화에 대하여 평가한 MovieLens 추천 시스템의 데이터베이스를 이용하여 평가되었다.

2. 잡음 감소를 위한 시스템 구성도

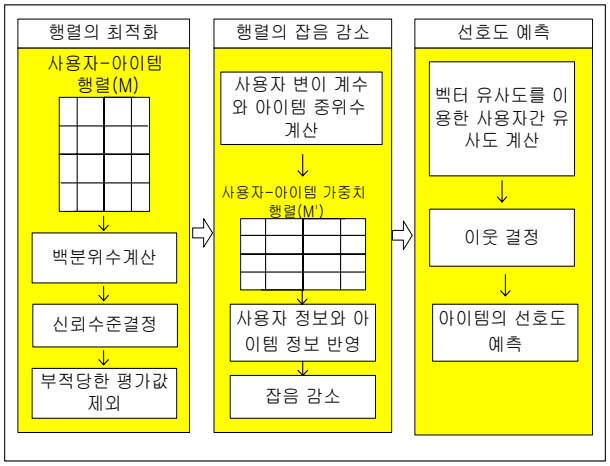
<표 2>는 사용자-아이템 행렬의 예로, MovieLens 추천 시스템에서 사용된 데이터로부터 31명의 사용자가 14개의 아이템에 대해 평가한 값을 무작위로 추출한 결과이다. <표 2>의 열은 아이템을 나타내며, 행은 사용자를 나타낸다.

<표 2>에서 사용자129나 사용자188은 대부분의 아이템들에 대해 각각 1이나 0.8 등의 값으로 평가를 하였다. 이와 같이 일률적으로 평가한 선호도의 분포는 사용자의 흥미를 정확하게 반영했다고 할 수 없으므로 이 평가값은 잡음을 포함한다고 할 수 있다. 본 논문에서는 사용자-아이템 행렬의 잡음을 감소시키기 위하여 행렬의 분포 정보를 이용한다. 사용자 분포 정보를 수치화하기 위하여 변이계수를 사용하며, 아이템 분포 정보는 아이템 중위수를 사용한다. 그런데 변이계수는 표준편차를 이용하기 때문에 사용자의 분포를 파악하는 산포도로서 이용하였을 경우에 오차를 포함하는 결과를 발생시킨다. 따라서 이러한 오차를 줄이기 위하여 행렬을 최적화하는 과정이 필요하며, 다음으로 사용자와 아이템의 정보를 반영함으로써 잡음을 감소시키는 과정, 마지막으로 이를 기반으로 선호도를 예측하는 과정이 필요하다.

〈표 2〉 사용자-아이템 행렬의 예

아이템 / 사용자	1	2	10	13	17	18	19	21	25	31	32	34	36	39
10	0.8		0.8		0.2	0.8	0.8	1		0.8	0.8		1	0.8
27	1	0.8	0.8		1	0.2	0.6	0.8	0.8		0.8	0.8	1	0.8
34	1	0.8	0.8		0.8	0.8	0.6	0.6	0.8	0.6	0.8	0	1	1
129	1	1	0.8		1		0	1	0.6		1	1	1	1
142		0.4	0.4		0.2		0	1	0.4	0.2	0.4	0	0.4	0.4
157	1	0.8	0.8		0.8		1	0.8	0.8		1	1	0.8	
188		0.8	0.8				0			0.6			0.6	0.8
245	0.8		0.8			0.8	0.6	0.8	0.6		1	0.8	0.6	0.8
254	1	0.6	0.4				0	0.8	1	0	0.8		0.8	
489	0	0.6	0.6	0.4			0.2	0.6	0.8		0.6		0.8	0.4
661	1	0.2	0.8	0.8	0	0.8	0	1	1	0	1	1	1	1
753	0.8		0.4		1			0.8	0.8	0	1	1	1	0.8
833	1		0.8		1			0	0.8	0.8	0.6	0.8	0.8	1
938	1	1	1		1			0.6	0.8	0.4	0.8	0.8	0.8	0.6
1161	1	0.8	1		0.8		0.2	1	1	0.8				1
1294	1	0.8	0.8		1		0.2	1	0.2		0.8	1	0.8	
1388	1	0.6	0.8			0		0.8	0.8		0.8		1	1
1400	0.8		0.8	1			0.8	0.8		0.8	0.8	1	0.2	
1498	0.8	0.8	0.8		0.8		0	0.8	0.8		1			1
1516	1	0.8			1			0.8	1	0	0.8	0.8	1	0.8
1624	1	1	0.8		0.8			0	0.6	1	0.6	0.8	1	0.8
1812	1		1	0		0.8	0	0.8	1		1		0	
1918		0.8	0.6		0.8	0.6	0	0.6			1	1	1	0.8
2110		0			0.8			0.8		0.6	0.6			0.4
2138	0.6	0	0.8		1			1			0.8			1
2200	0.6	0.6	1		0.8		0	0.8	0.2		0.8	0.8	0.6	0.8
2487	1	1	1			0.2	0.8	1			1	0.6	0.8	0.8
2662	0	0.6				0.6	0.8		0.8		1			1
2756	0.6	0.6	0.6		0.8	0.2		0.6	0.4		0.8	0.6	0.6	0.4
3086	0.8	1	0.8		0		0.6	0.8	0.8	0.6		1		
3267	0.2				1			0.8	0.8		0.4		0.8	0.8

(그림 1)은 추천 시스템에서 산포도를 이용하여 잡음을 감소시키기 위한 시스템 구성도를 기술한다. 이 시스템은 행렬의 최적화 단계, 가중치를 부여함으로써 잡음을 감소시키는 단계, 마지막으로 선호도를 예측하는 단계로, 총 3 단계로 구성된다. (그림 1)에서 사용자-아이템 행렬은 M, 가중치가 부여된 사용자-아이템 행렬은 M'로 정의한다.



(그림 1) 산포도를 이용하여 잡음 감소시키기 위한 시스템 구성도

사용자-아이템 행렬의 최적화 단계에서는 사용자 변이계수 계산을 위한 전처리로서 변이계수의 계산 결과에 오류를 가져오는 평가값을 제거한다. 평가값의 최상위와 최하위 부분에 속하는 값들은 변이계수의 계산에 미치는 영향이 매우 커서 사용자의 선호도 분포를 올바르게 측정할 수 없게 한다. 따라서 최상위와 최하위 부분의 극한값을 제거하기 위하여 백분위수를 사용한다. 사용자-아이템 행렬에서 사용자가 평가한 평가값의 집합을 R라고 정의하였을 때 집합 R로부터 제외되어야 하는 평가값 집합을 R2로, 나머지 평가값 집합을 R1이라고 정의한다.

이와 같은 정의를 기반으로 $R \supseteq R1, R \supseteq R2$ 로 기술하면, 식 (2)의 정의를 얻을 수 있다.

$$R = R1 \cup R2 \tag{2}$$

식 (3)는 최적의 평가값 집합 R1을 구하기 위해서는 집합 R로부터 R2를 제외시켜야 함을 나타낸다.

$$R1 = R - R2 \tag{3}$$

식 (4)는 집합 R1과 R2에 속한 평가값이 공유될 수 없음을 나타낸다.

$$R1 \cap R2 = \phi \tag{4}$$

행렬의 잡음 감소 단계에서는 사용자의 변이계수와 아이템 중위수를 계산하고, 이를 병합함으로써 사용자-아이템 가중치 테이블을 구성한다. 가중치는 최적화된 평가값 집합 R1을 기반으로 아이템에 대한 중위수를 계산한 후에, 그 결과를 사용자들의 변이계수와 병합시킨 결과이다. 사용자-아이템 가중치 행렬 M'의 요소는 m'_{ij}로 정의한다.

선호도 예측 단계에서는 사용자-아이템 가중치 행렬 M'을 기반으로 벡터 유사도를 이용하여 사용자간의 유사도를 계산하고, 새로운 사용자에 대한 이웃을 결정한다. 마지막으로, 가장 보편적인 메모리 기반의 선호도 예측 방법을 이용하여 선호도를 예측한다.

3. 사용자-아이템 행렬의 최적화

본 장에서는 산포도를 계산하는 변이계수의 단점을 해결하기 위하여 백분위수를 사용하여 사용자-아이템 행렬의 극단값을 제거하는 방법을 기술한다.

<표 2>에서 사용자27은 아이템18에 대하여 0.2로 평가하였으며, 이러한 결과는 중위값 0.8과의 차이가 매우 큰 0.6이다. 반면, 아이템18을 제외한 모든 아이템은 중위값의 차이가 ± 0.2의 차이를 갖는다. 이와 같은 극단값을 포함하여 변이계수를 계산하였을 경우, 중위값과의 차이가 매우 크므로 변이계수 측정에 방해를 준다. 따라서 이와 같이 변이계

수를 측정하는 데 방해를 주는 평가값을 제외하고 변이계수를 측정하기 위하여 백분위수를 사용한다.

백분위수는 자료의 크기를 나열하여 100등분할 때 만들어지는 값이다. 제 k분위수 P_k는 주어진 자료를 크기순으로 나열할 경우 적어도 k%의 자료들이 그 값보다 작거나 같고, 또한 적어도 (100-k)%의 자료들이 그 값보다 크거나 같게 되는 것이다. 식 (5)는 제 k백분위수 P_k 를 정의한다[17].

$$P_k = x_L + (\frac{k \cdot n}{100} - F_{-1}) \cdot \frac{c}{f}, k = 1,2,...99 \quad (5)$$

식 (5)에서 x_L은 제 k백분위수 P_k가 위치하는 계급의 하한값이며, n은 자료의 총개수이다. F_{-1}은 제 k백분위수 P_k가 위치하는 계급 직전의 누적도수이며, c는 제 k백분위수 P_k가 위치하는 계급의 간격이다. 마지막으로 f는 제 k백분위수 P_k가 위치하는 계급의 도수이다.

<표 3>은 식 (5)를 이용하여 <표 2>의 평가값을 대상으로 제 k백분위수 P_k의 값을 계산한 결과이다. <표 3>에서 각 열은 P_k의 값을, 행은 사용자를 나타낸다.

〈표 3〉 제 k백분위수 Pk의 값

pk / 사용자	P1	P3	P5	P7	P9	P10	P20	P40	P60	P80	P91	P92	P93	P95
10	0.254	0.362	0.47	0.578	0.686	0.794	0.8	0.8	0.8	0.84	1	1	1	1
27	0.244	0.332	0.42	0.508	0.596	0.642	0.8	0.8	0.8	0.96	1	1	1	1
34	0.072	0.216	0.36	0.504	0.6	0.6	0.6	0.8	0.8	0.92	1	1	1	1
129	0.06	0.18	0.3	0.42	0.54	0.6	0.8	1	1	1	1	1	1	1
142	0	0	0	0	0	0	0.2	0.4	0.4	0.4	0.46	0.52	0.58	0.7
157	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.88	1	1	1	1	1
188	0.03	0.09	0.15	0.21	0.27	0.33	0.6	0.6	0.8	0.8	0.8	0.8	0.8	0.8
245	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.8	0.8	0.8	0.838	0.856	0.874	0.91
254	0	0	0	0	0	0	0.24	0.64	0.8	0.88	1	1	1	1
489	0.018	0.054	0.09	0.126	0.162	0.18	0.36	0.52	0.6	0.64	0.8	0.8	0.8	0.8
661	0	0	0	0	0	0	0.12	0.8	1	1	1	1	1	1
753	0.036	0.108	0.18	0.252	0.324	0.396	0.72	0.8	0.88	1	1	1	1	1
833	0.054	0.162	0.27	0.378	0.486	0.594	0.76	0.8	0.8	1	1	1	1	1
938	0.42	0.46	0.5	0.54	0.58	0.6	0.6	0.8	0.8	1	1	1	1	1
1161	0.248	0.344	0.44	0.536	0.632	0.728	0.8	0.84	1	1	1	1	1	1
1294	0.2	0.2	0.2	0.2	0.2	0.2	0.68	0.8	0.88	1	1	1	1	1
1388	0.048	0.144	0.24	0.336	0.432	0.528	0.72	0.8	0.8	1	1	1	1	1
1400	0.248	0.344	0.44	0.536	0.632	0.728	0.8	0.8	0.8	0.88	1	1	1	1
1498	0.064	0.192	0.32	0.448	0.576	0.704	0.8	0.8	0.8	0.88	1	1	1	1
1516	0.072	0.216	0.36	0.504	0.648	0.792	0.8	0.8	0.88	1	1	1	1	1
1624	0.06	0.18	0.3	0.42	0.54	0.6	0.6	0.8	0.8	1	1	1	1	1
1812	0	0	0	0	0	0	0	0.8	0.96	1	1	1	1	1
1918	0.054	0.162	0.27	0.378	0.486	0.594	0.6	0.72	0.8	1	1	1	1	1
2110	0.02	0.06	0.1	0.14	0.18	0.22	0.4	0.6	0.6	0.8	0.8	0.8	0.8	0.8
2138	0.036	0.108	0.18	0.252	0.324	0.396	0.64	0.8	0.92	1	1	1	1	1
2200	0.02	0.06	0.1	0.14	0.18	0.24	0.6	0.6	0.8	0.8	0.82	0.84	0.86	0.9
2487	0.236	0.308	0.38	0.452	0.524	0.596	0.76	0.8	1	1	1	1	1	1
2662	0.036	0.108	0.18	0.252	0.324	0.396	0.6	0.68	0.8	0.96	1	1	1	1
2756	0.22	0.26	0.3	0.34	0.38	0.4	0.4	0.6	0.6	0.6	0.8	0.8	0.8	0.8
3086	0.048	0.144	0.24	0.336	0.432	0.528	0.6	0.8	0.8	0.88	1	1	1	1
3267	0.212	0.236	0.26	0.284	0.308	0.332	0.48	0.8	0.8	0.8	0.892	0.904	0.916	0.94

<표 3>과 같이 정의한 제 k백분위수 P_k에서 어떠한 k을 기준으로 극단값을 정의하느냐가 문제이다. 즉, 변이계수 계산을 위한 전처리로써 극단값을 지정하고 이 값을 평가값으로부터 제외함으로써 평가값의 신뢰도를 높일 수 있기 때문이다. 이를 위하여 평가값에 대한 신뢰구간을 설정하고, 이 신뢰구간의 값만을 대상으로 변이계수를 측정한다.

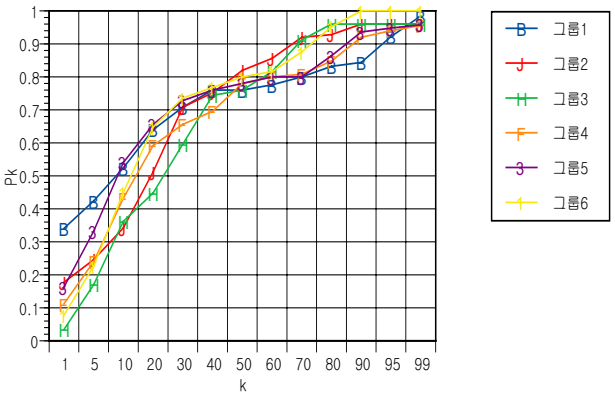
어떤 평가값 m_{ij}가 θ_L과 θ_U사이에 있을 확률이 $1-\alpha$라는 것을 식 (6)과 같이 정의한다.

$$P(\theta_L \prec m_{ij} \prec \theta_U) = 1 - \alpha \quad (6)$$

식 (6)에서 $\theta_L \sim \theta_U$는 평가값 m_{ij}에 대한 $100(1-\alpha)$의 신뢰구간(confidence interval)이다. 여기서, θ_L은 신뢰구간의 하한(lower limit of confidence interval)이며, θ_u 는 신뢰구간의 상한(upper limit of confidence interval)이다. $1-\alpha$는 신뢰구간이 평가값 m_{ij}를 포함할 확률로서 신뢰수준(confidence level), 신뢰계수(confidence coefficient), 또는 평가값의 신뢰도라고 표현한다.

반면, 평가값에 대한 신뢰수준을 정하는 것이 중요하다. 평가값들에 대한 신뢰수준을 정하는 기준에 따라 제외되는 평가값의 범위가 달라지기 때문이다. MovieLens 추천 시스템에서 사용된 데이터로부터 60명을 무작위로 추출하여 10명씩 그룹을 형성한 후, 신뢰수준을 0%로부터 99%까지로 하여 P_K의 값을 계산하였을 경우, (그림 2)와 같이 상승곡선을 나타낸다.

(그림 2)에서 보는 바와 같이 0<k<10에서 P_k의 값이 가장 급격하게 상승되는 것을 볼 수 있다. 반면, 90<k<99에서 P_k의 값은 가장 변화가 없다고 할 수 있다. 그러한 이유는 대부분의 사용자들이 아이템에 대하여 평가한 값들의 중위값이 0.5~0.8사이에 속하기 때문이다. <표 2>의 전체 평가값들의 중위값은 0.7이다. 이러한 결과는 신뢰수준을 90%에서 99%사이로 정하는 것이 가장 바람직함을 알 수 있다. 신뢰수준이 90%에서 99%사이에 해당하는 k의 범위 중에서 90<k<99에 해당하는 P_k의 값은 변이계수에 계산에 잡음이



(그림 2) 신뢰수준 0%로부터 99%의 PK

될만한 값을 포함하지 않기 때문에 이를 제외한다. 0<k<10에서의 P_k값만을 대상으로 이들 중 어떤 P_K값을 신뢰수준의 한계값을 정할 것인가를 결정한다.

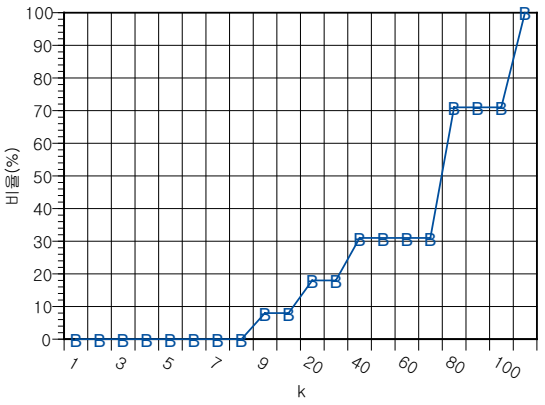
예를 들어, 신뢰수준을 99%로 설정하면 <표 3>에서 $P_1<m_{ij}<P_{99}$인 평가값 범위만으로 변이계수를 계산하며, 95%로 정하면 $P_5<m_{ij}<P_{95}$인 평가값 범위로 계산된다. <표 3>의 사용자27을 대상으로 신뢰수준을 정하고자 할 때, 신뢰수준을 99%라고 가정하자. P_1의 값이 0.244이므로, 0.2보다 작은 평가값을 갖는 아이템18은 변이계수 측정의 대상으로부터 제외가 된다. 이와 같은 원리를 이용하여 변이계수 측정에 방해가 되는 극단값을 제거할 수 있다.

<표 4>는 <표 2>의 전체 평가값을 대상으로 0<k<100인 범위의 k에 대하여 P_k값을 계산한 결과이다. 또한, 전체 평가값 중에서 각 P_k보다 작은 값이 존재하는 평가값의 비율도 나타낸다.

<표 4>와 (그림 3)은 $P_k>0$인 k를 기반으로 신뢰수준을 정하는 것이 가장 합당함을 보인다. 선호도를 평가하려고 하는 데이터 집합의 평균은 모두 다르므로 신뢰수준을 획일적으로 정하는 것이 아니고 각 집합의 특성을 고려하여 신뢰수준을 정하는 것이다. <표 4>와 (그림 3)에서는 k=9일 때 0에서 0.2로 P_k의 값이 변하므로 신뢰수준 92%에서 정하는 것이 바람직하다.

〈표 4〉 0<k<100인 범위의 k에 대한 Pk와 평가값의 비율

k	Pk	mij>Pk인 평가값의 수	해당 평가값의 비율(%)
1	0	0	0
2	0	0	0.000
3	0	0	0.000
4	0	0	0.000
5	0	0	0.000
6	0	0	0.000
7	0	0	0.000
8	0	0	0.000
9	0.2	26	8.638
10	0.2	26	8.638
20	0.6	55	18.272
30	0.6	55	18.272
40	0.8	96	31.894
50	0.8	96	31.894
60	0.8	96	31.894
70	0.8	96	31.894
80	1	214	71.096
90	1	214	71.096
91	1	214	71.096
92	1	214	71.096
93	1	214	71.096
94	1	214	71.096
95	1	214	71.096
96	1	214	71.096
97	1	214	71.096
98	1	214	71.096
99	1	214	71.096
K>100		301	100.000



(그림 3) k에 대한 평가값의 비율

(그림 3)은 <표 4>를 기반으로 전체 평가값 중 각 k에 해당하는 평가값의 비율을 나타낸다.

이와 같이 92%의 신뢰수준을 사용하여 92%의 신뢰수준보다 작은 값을 나타내는 평가값을 변이계수 계산의 대상으로부터 제외시킨다. <표 3>에서 각 사용자에 대한 P_9와 P_{91}의 값을 지정하고, <표 2>에서 $m_{ij} < P_9$인 m_{ij}를 제거한다. 예를 들어, <표 2>의 사용자34의 경우 P_9=0.6이다. 따라서 $m_{ij} < 0.6$ 이 되는 아이템34의 값 0은 행렬로부터 제거된다. <표 5>는 <표 2>의 사용자-아이템 행렬을 92%의 신뢰수준으로, <표 3>의 값을 기준으로 최적화시킨 결과이다.

〈표 5〉 92%의 신뢰수준을 기준으로 최적화시킨 행렬

아이템 / 사용자	1	2	10	13	17	18	19	21	25	31	32	34	36	39
10	0.8		0.8			0.8	0.8	1		0.8	0.8		1	0.8
27	1	0.8	0.8		1		0.6	0.8	0.8		0.8	0.8	1	0.8
34	1	0.8	0.8		0.8	0.8	0.6	0.6	0.8	0.6	0.8		1	1
129	1	1	0.8		1			1	0.6		1	1	1	1
142		0.4	0.4		0.2		0	1	0.4	0.2	0.4	0	0.4	0.4
157	1	0.8	0.8		0.8		1	0.8	0.8		1	1	0.8	
188		0.8	0.8							0.6			0.6	0.8
245	0.8		0.8			0.8	0.6	0.8	0.6		1	0.8	0.6	0.8
254	1	0.6	0.4				0	0.8	1		0.8		0.8	
489	0	0.6	0.6	0.4			0.2	0.6	0.8		0.6		0.8	0.4
661	1	0.2	0.8	0.8	0	0.8	0	1	1	0	1	1	1	1
753	0.8		0.4		1			0.8	0.8		1	1	1	0.8
833	1		0.8		1				0.8	0.8	0.6	0.8	0.8	1
938	1	1	1		1			0.6	0.8	0.4	0.8	0.8	0.8	0.6
1161	1	0.8	1		0.8			1	1	0.8				1
1294	1	0.8	0.8		1		0.2	1	0.2		0.8	1	0.8	
1388	1	0.6	0.8					0.8	0.8		0.8		1	1
1400	0.8		0.8	1			0.8	0.8		0.8	0.8	1		
1498	0.8	0.8	0.8		0.8			0.8	0.8		1			1
1516	1	0.8			1			0.8	1		0.8	0.8	1	0.8
1624	1	1	0.8		0.8				0.6	1	0.6	0.8	1	0.8
1812	1		1	0		0.8	0	0.8	1		1		0	
1918		0.8	0.6		0.8	0.6		0.6			1	1	1	0.8
2110					0.8			0.8		0.6	0.6			0.4
2138	0.6		0.8		1			1			0.8			1
2200	0.6	0.6	1		0.8		0	0.8	0.2		0.8	0.8	0.6	0.8
2487	1	1	1				0.8	1			1	0.6	0.8	0.8
2662		0.6				0.6	0.8		0.8		1			1
2756	0.6	0.6	0.6		0.8			0.6	0.4		0.8	0.6	0.6	0.4
3086	0.8	1	0.8				0.6	0.8	0.8	0.6		1		
3267					1			0.8	0.8		0.4		0.8	0.8

4. 행렬의 잡음 감소와 선호도 예측

본 장에서는 아이템의 중위수와 사용자들의 변이계수를 병합하여 사용자-아이템 가중치 행렬을 구성하고, 이를 이용하여 선호도를 예측하는 방법을 기술한다.

4.1 사용자-아이템 가중치 행렬 구성

선호도를 예측하기 위하여 <표 5>에 나타낸 사용자 들의 변이계수와 아이템에 대한 중위수를 사용하여 사용자-아이템 가중치 행렬을 구성한다.

사용자가 아이템에 대해 평가한 값의 분포정보와 아이템의 중위수를 병합시킴으로써 사용자가 평가한 값만으로 추천을 하는 경우에 나타나는 잡음을 줄일 수 있다. 즉, 사용자가 아이템에 대하여 평가한 값에 아이템 정보와 사용자 정보를 반영한 형태이다. 이를 위하여 사용자 변이계수와 아이템 중위수를 평가값에 곱하고, 그 값을 가중치로 정의한다.

식 (7)은 <표 5>의 사용자-아이템 행렬의 사용자를 cu_i로 정의하고, 그 사용자가 n개의 아이템에 대하여 평가한 경우에 있어서 그 사용자의 표준편차 S_{in} [20]를 나타낸다.

$$S_{in} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(m_{ij} - \overline{m_i})^2} \tag{7}$$

식 (7)에서 m_{ij} 는 j번째 아이템에 대하여 평가한 값을 나타내며, $\overline{m_i}$는 사용자 cu_i가 n개의 아이템에 대하여 평가한 평가값의 평균을 나타낸다. 식 (8)는 사용자 cu_i의 변이계수를 나타낸다.

$$CV_i = \frac{S_{in}}{\overline{m_i}} x100 \tag{8}$$

<표 6>은 식 (8)을 이용하여 <표 5>에 있는 사용자들의 변이계수를 계산한 결과이다.

사용자 cu_i의 아이템 d_j에 대한 평가값 m_{ij}의 가중치 m'_{ij}는 식 (9)를 사용하여 계산한다.

$$m'_{ij} = m_{ij} \cdot \sqrt{CV_i} \cdot w_j \tag{9}$$

식 (9)에서 w_j는 아이템 d_j의 중위수이며, CV_i는 사용자 cu_i의 변이계수다. CV_i의 값을 곱할 경우 그 값이 너무 작음으로 인하여 평가값에 상당히 큰 영향을 준다. 따라서 이 값은 가중치로 사용하기에 부적당하므로 루트를 취한 값을 사용한다.

<표 7>은 식 (9)을 이용하여 <표 5>의 사용자-아이템 행렬에 가중치를 부여한 결과를 나타낸다.

〈표 6〉 사용자 변이계수 계산의 예

	산술평균	표준편차	변이계수
10	0.84	0.09	0.1
27	0.84	0.12	0.14
34	0.8	0.15	0.18
129	0.94	0.13	0.14
142	0.35	0.27	0.78
157	0.88	0.1	0.12
188	0.72	0.11	0.15
245	0.76	0.13	0.17
254	0.6	0.39	0.65
489	0.56	0.19	0.35
661	0.69	0.43	0.62
753	0.84	0.19	0.23
833	0.76	0.3	0.39
938	0.84	0.16	0.19
1161	0.93	0.1	0.11
1294	0.76	0.31	0.41
1388	0.85	0.14	0.17
1400	0.85	0.09	0.11
1498	0.85	0.09	0.11
1516	0.89	0.11	0.12
1624	0.84	0.16	0.19
1812	0.62	0.47	0.76
1918	0.8	0.17	0.22
2110	0.64	0.17	0.26
2138	0.87	0.16	0.19
2200	0.7	0.22	0.31
2487	0.82	0.26	0.31
2662	0.8	0.18	0.22
2756	0.6	0.13	0.22
3086	0.8	0.15	0.19
3267	0.77	0.2	0.26

〈표 7〉 가중치가 부여된 사용자-아이템 가중치 행렬

아이템 / 사용자	1	2	10	13	17	18	19	21	25	31	32	34	36	39
10	0.259	0	0.207	0	0	0.207	0.155	0.259	0	0.155	0.207	0	0.259	0.207
27	0.38	0.243	0.243	0	0.304	0	0.137	0.243	0.243	0	0.243	0.243	0.304	0.243
34	0.43	0.275	0.275	0	0.275	0.275	0.155	0.206	0.275	0.155	0.275	0	0.344	0.344
129	0.379	0.303	0.243	0	0.303	0	0	0.303	0.182	0	0.303	0.303	0.303	0.303
142	0	0.283	0.283	0	0.141	0	0	0.707	0.283	0.106	0.283	0	0.283	0.283
157	0.343	0.219	0.219	0	0.219	0	0.206	0.219	0.219	0	0.274	0.274	0.219	0
188	0	0.25	0.25	0	0	0	0	0	0	0.14	0	0	0.187	0.25
245	0.326	0	0.261	0	0	0.261	0.147	0.261	0.196	0	0.326	0.261	0.196	0.261
254	0.803	0.386	0.257	0	0	0	0	0.514	0.643	0	0.514	0	0.514	0
489	0	0.284	0.284	0.142	0	0	0.071	0.284	0.379	0	0.284	0	0.379	0.189
661	0.79	0.126	0.505	0.379	0	0.505	0	0.632	0.632	0	0.632	0.632	0.632	0.632
753	0.384	0	0.154	0	0.384	0	0	0.307	0.307	0	0.384	0.384	0.384	0.307
833	0.623	0	0.399	0	0.499	0	0	0	0.399	0.299	0.299	0.399	0.399	0.499
938	0.433	0.347	0.347	0	0.347	0	0	0.208	0.277	0	0.277	0.277	0.277	0.208
1161	0.335	0.214	0.268	0	0.214	0	0	0.268	0.268	0.161	0	0	0	0.268
1294	0.639	0.409	0.409	0	0.511	0	0.077	0.511	0.102	0	0.409	0.511	0.409	0
1388	0.408	0.196	0.261	0	0	0	0	0.261	0.261	0	0.261	0	0.326	0.326
1400	0.264	0	0.211	0.198	0	0	0.158	0.211	0	0.158	0.211	0.264	0	0
1498	0.264	0.211	0.211	0	0.211	0	0	0.211	0.211	0	0.264	0	0	0.264
1516	0.344	0.22	0	0	0.275	0	0	0.22	0.275	0	0.22	0.22	0.275	0.22
1624	0.433	0.347	0.277	0	0.277	0	0	0	0.208	0.26	0.208	0.277	0.347	0.277
1812	0.873	0	0.698	0	0	0.558	0	0.558	0.698	0	0.698	0	0	0
1918	0	0.298	0.223	0	0.298	0.223	0	0.223	0	0	0.372	0.372	0.372	0.298
2110	0	0	0	0	0.327	0	0	0.327	0	0.184	0.245	0	0	0.164
2138	0.26	0	0.278	0	0.347	0	0	0.347	0	0	0.278	0	0	0.347
2200	0.333	0.267	0.444	0	0.356	0	0	0.356	0.089	0	0.356	0.356	0.267	0.356
2487	0.56	0.448	0.448	0	0	0.09	0.269	0.448	0	0	0.448	0.269	0.359	0.359
2662	0	0.227	0	0	0	0.227	0.227	0	0.303	0	0.378	0	0	0.378
2756	0.283	0.226	0.226	0	0.302	0	0	0.226	0.151	0	0.302	0.226	0.226	0.151
3086	0.348	0.348	0.278	0	0	0	0.156	0.278	0.278	0.156	0	0.348	0	0
3267	0	0	0	0	0.405	0	0	0.324	0.324	0	0.162	0	0.324	0.324

<표 7>에 나타난 행렬은 결측치를 갖지 아니하므로 완성 행렬(Complete matrix)이다. 통상적인 협력적 여과 사용자-아이템 행렬 M에서 추천으로 제공될 수 있는 아이템은 $(I_a \cap I_j)$의 결과이다. 여기서 I_a는 사용자 cu_a가 평가한 아이템들의 집합이고, I_j는 사용자 cu_j가 평가한 아이템들의 집합이다. 그러나 사용자-아이템 행렬의 대부분은 희박성을 나타내고, 이로 인하여 $(I_a \cap I_j)$의 결과가 공집합이 될 가능성도 있다. 그 결과, 추천을 제공할 수 없는 경우가 발생하기도 한다. 반면, <표 7>과 같이 완성행렬을 사용할 경우 추천을 위하여 전체 아이템 집합 I와 I_a의 공집합 $(I_a \cap I)$을 사용하므로 그 결과가 공집합이 나올 경우가 없다는 장점을 갖는다.

4.2 사용자의 선호도 예측

새로운 사용자의 선호도를 예측하기 위하여 4.1 절에서와 같이 구성된 사용자-아이템 가중치 행렬을 기반으로 이웃을 선정하고, 이웃들이 평가한 선호도를 기반으로 선호도를 예측한다.

협력적 여과 시스템에서 사용자간의 유사도를 계산하기 위해 사용되는 피어슨 상관 계수, 스피어맨 순위 상관 계수, 벡터 유사도 등의 방법이 사용되고 있다. 본 논문에서는 이들 중에서 행렬의 모든 요소를 대상으로 하는 벡터 유사도[11,13]를 사용한다.

식 (10)은 벡터 유사도를 이용하여 사용자 cu_i와 사용자 cu_a간의 유사도($w'_{a,i}$)를 계산하기 위한 식이다.

$$w'_{a,i} = \sum_j \frac{m'_{aj}}{\sqrt{\sum_{k \in d_a} m'^2_{ak}}} \frac{m'_{ij}}{\sqrt{\sum_{k \in d_i} m'^2_{ik}}} \quad (10)$$

<표 8>은 <표 7>의 사용자3086과 나머지 사용자간의 유사도를 식 (10)의 벡터 유사도를 이용하여 계산한 결과를 나타낸다.

식(10)을 이용하여 유사도를 계산하고 새로운 사용자의 선호도를 예측한다. 이를 위하여 가장 보편적인 메모리 기반의 선호도 예측 방법[2]을 사용한다. 추천을 받을 사용자를 cu_a라고 가정하였을 때, 이 사용자에 대한 아이템 d_j의 선호도 예측은 식 (11)을 이용한다.

$$p_{a,j} = \overline{m_a} + \frac{\sum_{b=1}^{N} (m'_{bj} - \overline{m'_b}) \times w'_{a,b}}{\sum_{b=1}^{N} w'_{a,b}} \quad (11)$$

식 (11)에서 $\overline{m_a}$는 사용자 cu_a가 아이템에 대하여 평가한 선호도의 평균을 나타낸다. $\overline{m'_b}$는 사용자-아이템 가중치 행렬 M'에서 사용자 cu_b에 해당하는 가중치들의 평균을 나타낸다. $w'_{a,b}$는 식 (10)에 의해 계산된 사용자 cu_a, cu_b의 벡터 유사도를 나타낸다.

〈표 8〉 사용자 3086과 다른 사용자간의 벡터 유사도

	분자	분모	유사도
10	0.268	0.52	0.515
27	0.525	0.7	0.75
34	0.504	0.789	0.639
129	0.545	0.752	0.725
142	0.469	0.807	0.581
157	0.506	0.62	0.815
188	0.178	0.395	0.452
245	0.427	0.648	0.659
254	0.807	1.153	0.7
489	0.373	0.657	0.568
661	1.03	1.541	0.669
753	0.481	0.82	0.586
833	0.624	1.047	0.596
938	0.599	0.779	0.769
1161	0.44	0.577	0.762
1294	0.839	1.099	0.763
1388	0.428	0.666	0.642
1400	0.35	0.483	0.726
1498	0.341	0.527	0.647
1516	0.41	0.615	0.668
1624	0.543	0.757	0.718
1812	0.847	1.354	0.626
1918	0.357	0.731	0.488
2110	0.12	0.464	0.258
2138	0.264	0.613	0.431
2200	0.58	0.839	0.691
2487	0.735	0.992	0.742
2662	0.199	0.585	0.339
2756	0.423	0.602	0.703
3086	0.644	0.644	1
3267	0.18	0.627	0.287

식 (12)는 사용자3086이 18번째 아이템에 대하여 평가한 선호도를 예측하기 위해 식 (11)를 사용하였을 때, 그 결과를 보인다.

$$p_{3086,18} = 0.67046 \quad (12)$$

5. 성능 평가

협력적 여과 시스템에서 가장 광범위하게 사용되고 있는 데이터 집합은 MovieLens 추천 시스템에서 사용한 데이터 집합[8]이다. MovieLens 시스템의 데이터 집합은 70,000 명의 사용자로부터 2백 8십만개 이상의 평가를 포함했던 EachMovie 데이터 집합으로 데이터를 정제하여 체계적으로 수집한 것으로서, 기계 학습이나 산술적인 연구 프로젝트를 위하여 사용되어 왔다. 이러한 MovieLens 데이터 집합은 여러 연구가에 의해 사용되어 왔다. 예를 들어, 초기 평가 문제를 다루고 있는 추천 시스템인 [17]과 아이템 기반의 평가를 다루고 있는 [15], 그리고 공동의 의견 추출 문제를 다루고 있는 [10], 마지막으로 협력적인 법적 제재(sanctioning)에서의 연구[9] 등을 위하여 사용하였다[4]. 본 논문에서도 제안된 방법의 성능을 평가하기 위해

Eachmovie 데이터 집합으로부터 발취한 MovieLens 시스템의 데이터 집합을 대상으로 실험하였다. 이를 위하여 훈련 집합과 테스트 집합으로 각각 1,000명, 500명의 사용자를 임의로 수집하였다. 이들 사용자 중에서 1600개의 영화에 대해 적어도 30개 이상 평가하였던 사용자만을 고려하면 훈련 집합은 883명, 테스트 집합은 437명이 실험의 대상이 되었다.

협력적 여과 시스템의 성능은 사용자가 원하는 아이템을 얼마나 효율적으로 추천하는가의 정도를 나타내며, 범위(coverage)와 정확도(accuracy)를 이용하여 측정한다. 범위는 추천 시스템이 추천을 제공할 수 있는 아이템의 퍼센트이다. 즉, 협력적 여과 시스템이 실제로 사용자가 각 아이템에 대해 평가하기 전에 사용자가 원하는 아이템을 추천할 수 있는가의 정도를 나타낸다[3].

정확도는 기존의 연구에서 다양한 척도로써 이용되어 왔다. 가장 일반적인 방법은 통계적인 추천 정확도(Statistical recommendation accuracy)와 의사 지원 정확도(Decision-support accuracy)이다. 통계 추천 정확도는 사용자에 의해 평가된 값과 시스템에 의해 예측된 값과의 차이점을 측정한다. MAE(Mean Absolute Error), RMSE(Root Mean Squared Error) 등이 있다. MAE는 실제 평가값과 예측값 사이의 차이이며 협력적 여과 시스템에서 많이 사용되고 있는 평가 척도이다[3]. 각 평가값-예측값의 쌍 $<p_i,q_i>$는 이 값들 사이의 차의 절대값 $|p_i-q_i|$를 계산한다. MAE는 N개의 평가값과 예측값의 절대 오차를 합산함으로써 계산할 수 있다. 식 (13)은 MAE를 계산하기 위한 식이다. MAE가 낮아질수록 추천 엔진의 정확도는 더 높아진다.

$$MAE = \frac{\sum_{i=1}^{N}|p_i - q_i|}{N} \quad (13)$$

의사 지원 정확도는 추천 시스템이 사용자에게 높은 질의 추천이 가능한지의 정도를 측정하여, 그 종류로는 ROC(Receiver Operating. Characteristic) 민감도, PRC(Precision-Recall Curve) 민감도 등이 있다[9]. ROC 민감도는 그래프의 곡선을 이용하여 여과 시스템의 성능을 측정한다. 곡선 아래 영역은 여과 시스템이 '좋은(good) 아이템'을 보유할수록 증가한다. 여기서 좋은 아이템과 '나쁜(bad) 아이템'을 결정하는 것이 필요하다. ROC 민감도 측정은 추천 시스템이 좋은 아이템을 얼마나 많이 추천할 수 있는 가를 나타낸다. 특히, 1.0은 완전 여과라고 할 수 있으며, 0.5는 임의의 여과라고 할 수 있다[16].

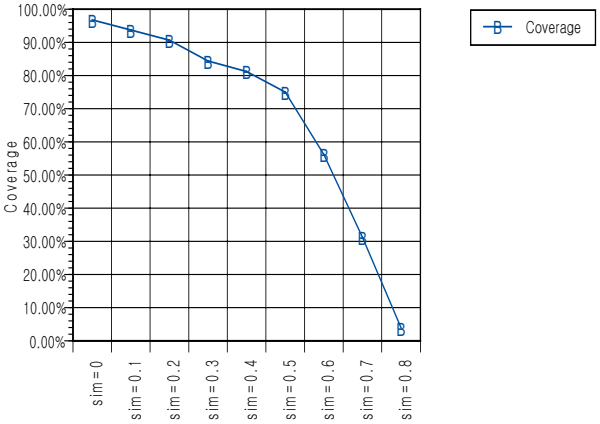
본 논문에서는 제안된 방법(Stat_CF)의 성능을 평가하기 위하여 유사도의 한계값을 지정하여 그 한계값에 따라 제안된 방법의 범위와 ROC 민감도가 변화됨을 보인다. 다음으로, 기존의 피어슨 상관 계수를 이용한 대표적인 협력적 여과 방법(CF_G), k백분위수를 사용하지 않은 방법(Stat_k_CF)과 이웃의 수를 변화시키면서 MAE를 사용하여 성능을 평가하였으며, 또한 이들 방법간의 추천 속도를 비교하였다. <표 9>는 Stat_CF, CF_G, 그리고 Stat_k_CF의 온라인과 오프라인 상의 작업을 비교하여 설명하였다.

〈표 9〉 Stat_CF, CF_G, 그리고 Stat_k_CF의 비교

	오프라인	온라인
Stat_CF	백분위수를 이용한 최적화, 가중치 부여	- 코사인유사도 - 평균 편차를 사용한 예측
Stat_k_CF	가중치 부여	- 코사인유사도 - 평균 편차를 사용한 예측
CF_G	없음	- 피어슨 상관 계수 - 평균 편차를 사용한 예측

〈표 10〉 사용자 유사도 변화에 따른 범위와 ROC 민감도

	범위	ROC-0.3
sim=0.0	96.80%	0.7323
sim=0.1	93.75%	0.7427
sim=0.2	90.62%	0.7534
sim=0.3	84.38%	0.7613
sim=0.4	81.20%	0.7681
sim=0.5	75.00%	0.7732
sim=0.6	56.20%	0.7954
sim=0.7	31.20%	0.8126
sim=0.8	3.90%	0.8964

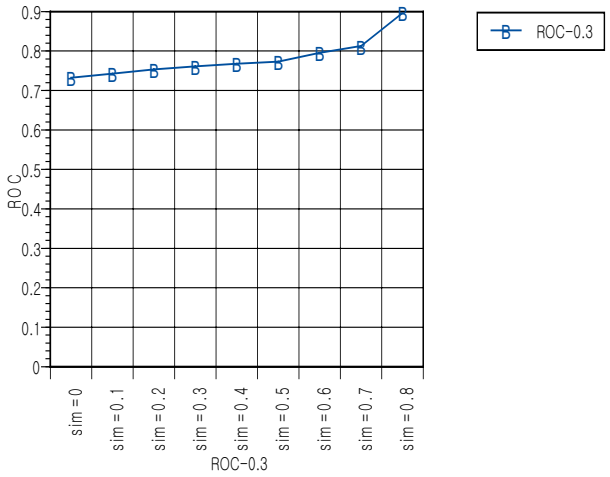


(그림 4) 사용자 유사도 변화에 따른 범위

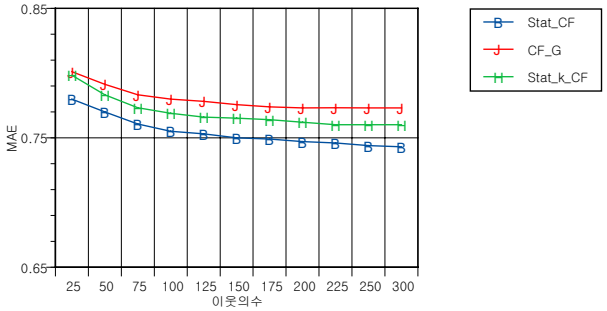
<표 10>은 제안된 방법의 사용자 유사도를 0에서 0.8까지 변화시킴에 따른 범위와 ROC 민감도를 나타낸다. 본 논문에서는 좋은 아이템과 나쁜 아이템을 구분하기 위해 사용자 자신의 평가값을 사용한다. ROC 민감도에서 좋은 아이템과 나쁜 아이템의 기준은 평가값이 0.3보다 클 경우 좋은 아이템으로, 그 외의 경우는 나쁜 아이템으로 정의하였다.

(그림 4)는 <표 10>을 기반으로 하고 있으며, 사용자 유사도 변화에 따른 범위의 변화 곡선을 나타낸다. 또한, (그림 5)는 <표 10>를 기반으로 한 ROC 민감도의 변화 곡선을 나타낸다.

(그림 4)에서 본 논문에서 제안된 방법(Stat_CF)은 유사도의 한계값이 커짐에 따라 그 값이 급강하하는 곡선을 나타낸다. 즉, 유사도의 한계값이 낮을 때는 이웃의 수가 많아서 이에 따라 추천되는 아이템의 수가 많으나 이웃의 수가 적어짐에 따라 추천되는 아이템의 수가 적어짐을 보인다.



(그림 5) 사용자 유사도 변화에 ROC 민감도



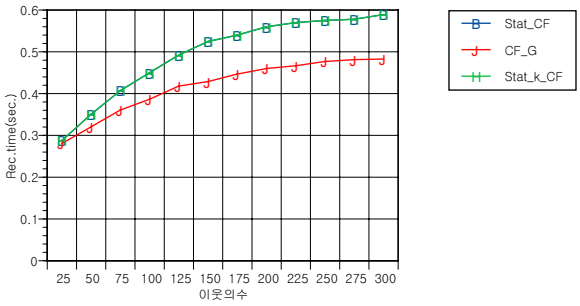
(그림 6) Stat_k_CF, CF_G, 그리고 Stat_k_CF의 MAE

(그림 5)에서 유사도의 한계값이 증가함에 따라 ROC 민감도 역시 서서히 상승하는 곡선을 보이며 유사도 한계값이 0.6이상일 때는 ROC 민감도가 급격히 상승함을 보인다. 즉, 유사도의 한계값이 높다는 것은 이웃으로 정한 기준의 신뢰도가 높다는 것으로, 근접한 이웃이 평가한 기준으로 아이템을 추천하였을 때 그 정확도 역시 높다는 것을 나타낸다.

(그림 6)은 이웃의 수를 변화시킴에 따른 제안된 방법과 대표적인 협력적 여과 방법(CF_G), k백분위수를 사용하지 않은 방법(Stat_k_CF)의 MAE 의 변화곡선을 나타낸다.

Stat_k_CF, CF_G, 그리고 Stat_k_CF은 모두 이웃의 수에 민감하지 않으며, 이웃의 수 변화에 따른 MAE는 모두 비슷한 변화 곡선을 보인다. 전반적으로 Stat_CF는 CF_G와 Stat_k_CF보다 낮은 MAE를 보이며, 특히 Stat_k_CF보다는 다소 성능이 높음을 보인다. 즉, 백분위수를 사용하여 행렬을 최적화 시키고, 사용자와 아이템의 정보를 반영함으로써 잡음을 감소시키는 방법이 사용자-아이템 행렬의 잡음이 그대로 있는 대표적인 협력적 여과 방법보다는 높은 성능을 보인다는 의미이다. 또한, 사용자 변이계수의 단점을 보완하지 않은 경우, 즉 백분위수를 사용하여 행렬을 최적화 시키지 않았을 경우보다 단점을 보완한 Stat_CF의 방법이 보다 효율적이라는 것을 보인다.

(그림 7)은 Stat_CF, CF_G, 그리고 Stat_k_CF의 추천 속도를 나타낸다. <표 9>에 나타난 오프라인과 온라인 작업



(그림 7) Stat_CF, CF_G, 그리고 Stat_k_CF의 추천 속도

중 오프라인상에서의 작업은 추천을 하기 위한 전처리 작업이므로 추천 속도를 계산하는 과정에서 제외하였다. 즉, 온라인 상에서의 추천 속도만을 측정하였다.

(그림 7)은 기존의 CF_G가 가장 추천 속도가 빠름을 보인다. 반면, Stat_CF와 Stat_k_CF는 같은 속도를 나타내며, CF_G보다 낮은 속도를 나타낸다. 이와 같은 결과는 <표 9>에서와 같이 온라인 상에서 Stat_CF와 Stat_k_CF는 같은 방법을 사용하고 있으며, 이들은 모든 아이템을 대상으로 유사도를 계산하므로 공통으로 평가된 아이템만을 대상으로 유사도를 비교하는 피어슨 상관 계수를 사용하는 CF_G와 비교했을 때 다소 낮은 속도를 나타낸다. 이와 같은 속도차는 사용자가 큰 차이로 느끼지 않으므로 성능 평가에 큰 영향을 미치지 않는다고 볼 수 있으나, 이를 보완하는 방법이 연구된다면 보다 우수한 성능을 갖는 추천 시스템이 될 것이다.

6. 결론 및 향후 과제

협력적 여과 시스템에서는 사용자가 아이템에 대하여 평가한 평가값만을 기반으로 아이템을 추천하므로 그 평가값은 중요한 가치를 가진다. 그러나 모든 사용자가 그 가치를 의식하여 평가에 대해서 신중을 기하는 것은 아니다. 따라서 평가값에는 잡음을 포함할 수 밖에 없다. 이를 보완하기 위하여 본 논문에서는 산포도를 이용하여 사용자-아이템 행렬의 잡음을 감소시키는 방법을 제안하였다. 잡음을 감소시키는 방법으로 사용자와 아이템의 정보를 행렬에 반영하는 방법을 사용하였다. 백분위수를 사용하여 행렬의 극한값을 제거하고, 이를 기반으로 사용자 변이계수와 아이템 중위수를 병합하여 가중치 행렬을 구성함으로써 잡음을 감소시킬 수 있었다. 제안된 방법과 기존의 방법과 비교하였을 경우 전반적으로 추천의 정확도를 높아짐을 보였다.

향후, 추천의 정확도를 높이기 위하여 사용자-아이템 행렬을 최적화하는 방법의 연구가 필요하다.

참 고 문 헌

[1] Basu, C., Hirsh, H., and Cohen, W. W., "Recommendation as classification:Using social and

content-based information in recommendation," In Proc. of the Fifteenth National Conference on Artificial Intelligence, 1998.
[2] Breese, John. S. and Kadie, C., "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," In Proc. of the Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.
[3] Herlocker, J., Konstan, J., Borchers, A., and Riedl, J., "An Algorithmic Framework for Performing Collaborative Filtering," In Proc. of the 1999 Conference on Research and Development in Information Retrieval, 1999.
[4] Herlocker, J., Konstan, J., Terveen, L., and Riedl, J., "Evaluating Collaborative Filtering Recommender Systems," ACM Transactions on Information Systems, Vol. 22, No. 1, 2004.
[5] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J., "GroupLens:Applying Collaborative Filtering to Usenet News," Communications of the ACM, Vol. 40, No. 3, pp. 77-87, 1997.
[6] Lee, W. S., "Collaborative learning for recommender systems," In Proc. of the Conference on Machine Learning, 1997.
[7] Massa, P. and Avesani, P., "Trust-aware Collaborative Filtering for Recommender Systems," In Proc. of International Conference on Cooperative Information Systems, 2004.
[8] MovieLens collaborative filtering data set, Hppt://www.cs.umn.edu/Research/GroupLens/index.html, GROUPLENS RESEARCH PROJECT, 2000.
[9] Mui, L., Ang, C., and Mohtashemi, M., "A Probabilistic Model for Collaborative Sanctioning,"MIT LCS Technical Memorandum 617, 2001.
[10] Reddy, P. K., Kitsuregawa, P., Sreekanth, P., and Rao, S. S., "A Graph based Approach to Extract a Neighborhood Customer Community for Collaborative Filtering," In Proc. of Databases in Networked Information Systems, Second International Workshop, Lecture Notes in Computer Science, Springer-Verlag, 2002.
[11] Rijsbergen, V. and Joost, C., *Information Retrieval*, Butterworths, London-second edition, 1979.
[12] Robu, V. and Poutre , J.A. La, "Learning the Structure of Utility Graphs Used in Multi-Issue Negotiation through Collaborative Filtering," In Proc. of the 8th International Pacific Rim Workshop on Multi-Agent Systems (PRIMA'05), 2005.
[13] Salton, G. and McGill, M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
[14] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., "Application of Dimensionality Reduction in Recommender System-A Case Study," In Proc. of ACM WebKDD Web Mining for E-Commerce Workshop, 2000.
[15] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., "Item-based Collaborative Filtering Recommendation Algorithms," In Proc. of the 10th international World Wide Web Conference(WWW10), 2001.
[16] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., "Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System," In Proc. of CSCW'98, 1998.
[17] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M., "Generate Models for Cold-start Recommendations," In Proc. of the 2001 ACM SIGIR Workshop on Recommender Systems, 2001.
[18] Spiegel, Murray R. and Stephens, Larry J., *Schaum's Outline of Statistics*, McGraw-Hill, 1998.
[19] Wang, J., Vries, Arjen P. de, and Reinders, Marcel J. T., "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion,"In Proc. of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR06), 2006.
[20] 양완연, 일반통계학, 연학사, 1995.

고 수 정
e-mail : sjko@induk.ac.kr
1990년 인하대학교 전자계산학과(학사)
1997년 인하대학교 전자계산교육전공 (석사)
2002년 인하대학교 전자계산공학과(박사)
2003년~2004년 University of Illinois at Urbana-Champaign Post Doc.
2004년~2005년 Colorado State University Research Scientist
2005년~현재 인덕대학 컴퓨터소프트웨어과 교수
관심분야 : 데이터마이닝, 정보검색, 기계학습