협동적 여과에서의 희소성 문제 해결을 위한 데이타 블러링 기법

(Data Blurring Method for Solving Sparseness Problem in Collaborative Filtering)

김 형 일 † 김 준 태 ††
(Hyungil Kim) (Juntae Kim)

요 약 추천 시스템은 사용자의 선호도를 분석하고, 아이템에 대한 사용자의 선호도를 예측하여 아이템을 추천하는 시스템이다. 다양한 추천 기법 중에 협동적 여과(collaborative filtering)는 상용화된 시스템에 성공적인 적용이 이루어진 기법이다. 그러나 협동적 여과는 데이타의 희소성 문제(sparseness problem)와 초기 추천 문제(cold-start problem)에 대해 취약점을 가지고 있다. 만약 매우 적은 양의 선호도 데이타가 존재하면 많은 유사 사용자를 찾기 어려우며, 이것은 추천 성능을 저하시키는 요인으로 작용한다. 또한 선호도 정보가 없는 새로운 사용자에게는 아이템을 전혀 추천할 수 없는 문제가 발생한다. 본 논문에서는 사용자와 아이템에 대한 추가 속성 정보를 통합하여 협동적 여과의 희소성 문제와 초기 추천 문제를 해결하고 추천 성능을 향상시키는 기법을 제안한다. 본 논문에서 제안하는 기법은 추가 속성 정보의 확률분포를 이용하여 알려지지 않은 선호도 값을 예측함으로써 선호도 데이타를 변경하고, 변경된 선호도 데이타에 협동적 여과를 적용하여 top-N 추천을 생성하는 것이다. 이와 같은 선호도 데이타 변경 기법을 데이타 블러링(data blurring)이라 한다. 몇 가지 실험 결과를 통해 제안된 기법의 효과를 확인하였다.

키워드 : 데이타 블러링, 데이타 마이닝, 정보여과, 기계학습

Abstract Recommendation systems analyze user preferences and recommend items to a user by predicting the user's preference for those items. Among various kinds of recommendation methods, collaborative filtering(CF) has been widely used and successfully applied to practical applications. However, collaborative filtering has two inherent problems: data sparseness and the cold-start problems. If there are few known preferences for a user, it is difficult to find many similar users, and therefore the performance of recommendation is degraded. This problem is more serious when a new user is first using the system. In this paper, we propose a method of integrating additional feature information of users and items into CF to overcome the difficulties caused by sparseness and improve the accuracy of recommendation. In our method, we first fill in unknown preference values by using the probability distribution of feature values, then generate the top-N recommendations by applying collaborative filtering on the modified data. We call this method of filling unknown preference values as *data blurring*. Several experimental results that show the effectiveness of the proposed method are also presented.

Key words : Data Blurring, Data Mining, Information Filtering, Machine Learning

1. 서 론

추천 시스템은 사용자의 선호도를 분석하고, 아이템들에 대한 사용자의 선호도를 예측하여 책, 영화, 음악, 새로운 기사, 웹 페이지 등과 같은 아이템들을 추천한다. 사용하는 정보나 알고리즘에 따라 추천 시스템은 협동적 여과기반 추천 시스템, 내용기반 추천 시스템, 인구통계학적 추천 시스템으로 나뉜다.

내용기반 추천 시스템은 각 아이템에 관한 서술 등의 아이템 내용 정보를 이용한다[1-4]. 내용기반 추천 시스템은 알려진 선호 아이템들의 집합에 대해 시스템은 내

· 이 연구는 2003학년도 동국대학교 연구년 지원에 의하여 이루어졌음
† 학생회원 : 동국대학교 컴퓨터공학과
hikim@dongguk.edu
†† 종신회원 : 동국대학교 컴퓨터공학과 부교수
(Corresponding author 임)
jkim@dongguk.edu
논문접수 : 2004년 12월 30일
심사완료 : 2005년 4월 9일

용을 분석하고, 사용자의 선호도를 나타내는 사용자 프로파일(profile)을 생성한 후, 사용자의 프로파일과 아이템 사이의 유사도를 계산하여 사용자가 선호할 아이템을 예측한다. 내용기반 방법은 뉴스 기사나 웹 페이지와 같이 내용 정보가 풍부한 경우에 적합하다. 그러나 영화나 음악과 같은 멀티미디어 정보처럼 내용 분석이 어려운 경우 유사도 측정 및 아이템 추천이 어렵다는 단점이 있다.

인구통계학적 추천 시스템은 사용자들의 나이, 성별, 직업과 같은 개인 정보를 이용한다[5]. 사용자들의 인구통계학적 정보를 활용하여 사용자들 사이의 거리를 측정하고, 이웃한 사용자들을 찾아 특정 아이템에 대한 이웃 사용자들의 평균 선호도를 계산하여 사용자의 선호도를 예측한다. 아이템 범주에 대한 개인적인 성향은 인구통계학적 속성에 의해 쉽게 구별될 수 있기 때문에 인구통계학적 정보를 기반으로 하는 추천 시스템은 백화점과 같은 다양한 종류의 아이템이나, 인구통계학적 집단 성향이 잘 나타나는 영역에 적합하다.

협동적 여과기반 추천 시스템은 아이템들에 대한 각 사용자들의 평가 정보를 이용한다[6-10]. 가장 일반적인 접근 방식은 사용자들 사이의 평가 정보를 비교하여 유사 사용자를 추출하고, 아이템에 대한 유사 사용자의 선호도를 기반으로 특정 아이템에 대한 사용자의 선호도를 예측하는 것이다. 협동적 여과 기법은 아이템의 내용 정보를 필요로 하지 않기 때문에 내용을 분석하기 어려운 음악이나 영화 같은 아이템을 추천할 수 있다.

협동적 여과는 많은 장점이 있으며, 다양한 응용 시스템에 성공적으로 적용된 기법이지만 데이타의 희소성 문제(sparseness problem)와 초기 추천 문제(cold-start problem)에 취약하다. 만약 매우 적은 선호도 정보가 존재한다면 많은 유사 사용자를 찾기 어려우며, 따라서 추천의 정확도는 낮아지게 된다. 이러한 취약점은 특히 시스템의 초기나 새로운 사용자에게 아이템을 추천할 경우에 결정적인 단점으로 작용된다. 새로운 사용자는 아이템에 대한 선호도 정보가 없기 때문에 유사 사용자를 추출할 수 없으며, 따라서 시스템은 아이템을 전혀 추천할 수 없게 되는데, 이를 초기 추천 문제라 한다.

데이타 희소성 문제와 초기 추천 문제를 해결하기 위한 하나의 방법은 사용자와 아이템의 추가 속성 정보를 활용하는 것이다. 본 논문에서는 협동적 여과에서 데이타 희소성 문제와 초기 추천 문제를 해결하기 위해 속성 정보를 통합하는 기법을 제안한다. 제안한 방법에서는 먼저 속성 값의 확률분포를 이용하여 알려지지 않은 선호도 값을 예측하고, 수정된 데이타를 이용하여 협동적 여과를 수행함으로써 아이템을 추천한다. 이와 같이 선호도 값을 생성하는 기법을 데이타 블러링(data blurring)이라 한다.

본 논문의 2장에서는 협동적 여과를 중심으로 관련 연구를 정리하고, 3장에서는 본 논문에서 제안한 데이타 블러링의 개념과 블러링 알고리즘에 대해 설명한다. 4장에서는 제안한 기법의 실험 결과를 분석하고, 5장에서는 결론과 향후 연구에 대해 기술한다.

2. 관련 연구

협동적 여과기반 추천 시스템은 많은 사용자로부터 추천이 이루어진 아이템에 대한 평가를 활용한 기법이다. 근접이웃기반 협동적 여과는 사용자들 사이의 평가 정보를 비교하여 유사 사용자들을 추출하고, 유사 사용자들의 선호도를 기반으로 사용자의 아이템 선호도를 예측한다. 사용자 유사도는 피어슨 상관관계[8]나 벡터 유사도[11]와 같은 통계적 기법에 의해 계산된다. 유사도가 계산되고 나면 식 (1)에서와 같이 아이템에 대해 유사한 사용자들의 평가에 대한 가중 평균을 계산하여 사용자에 대한 아이템 선호도를 예측한다. 식 (1)에서 $P_{a,i}$는 아이템 i에 대한 사용자 a의 예측된 선호도 값이고, $r_{u,i}$는 사용자 u의 아이템 i에 대한 평가 값이다. $S_{a,u}$는 사용자 a와 사용자 u사이의 유사도이고, n은 사용자들의 총수이다.

$$P_{a,i}=\frac{\sum_{u=1}^{n}(s_{a,u}\cdot r_{u,i})}{\sum_{u=1}^{n}s_{a,u}} \quad (1)$$

GroupLens[8]와 같은 협동적 여과 시스템에서는 사용자의 선호도를 예측하는데 상관관계(correlation)기반 기법이 사용되었고, 다양한 변형된 기법들이 추천 시스템의 정확도 향상을 위해 제안되었다. Billsus와 Pazzani[6]는 충분한 정보가 없을 때 사용자의 선호도 예측을 위해 속성 추출 기술을 적용하였으며, 사용자-아이템 선호도 행렬의 차원을 줄이기 위하여 SVD (Singular Value Decomposition)을 사용하였다. Breese 등[11]은 사용자로부터의 평가 정보가 없는 아이템에 대해 기본 평가 값을 사용하였고, 각 아이템에 얼마나 많은 사용자가 평가를 수행했느냐에 따라 서로 다른 가중치를 적용하였다. Herlocker 등[7]은 다양한 유사도 계산 방식과 유사도 가중치 방법에 대한 실험을 수행하였다. 유사도 계산에는 피어슨 상관계수, 스피어만 상관계수, 벡터 유사도, 엔트로피를 이용하였고, 선호도 값을 구하는 방법으로 평균 가중치, 유사 사용자의 아이템 선호도의 가중치 합, z 평균 방법을 이용하였다. Shardanand[9]는 사용자 사이의 유사도가 아닌 아이템-아이템 유사도를 이용하는 방법을 제안한 바 있다.

협동적 여과에 의한 추천의 장점은 유사한 사용자 평

가를 기반으로 아이템을 추천하기 때문에 내용기반 방법에 비해 정확하며, 아이템 자체에 대한 내용 정보가 없어도 아이템 추천이 가능하다는 것이다. 그러나 전술한 바와 같이 협동적 여과는 데이타 희소성과 초기 추천 문제를 가지고 있다. 예를 들어, 만약 사용자의 선호도가 구매이력에 의해 기록되고 각 사용자는 수천 개의 아이템들 중에 몇 개의 아이템만을 구매하였다면 사용자들 사이의 벡터 유사도는 대부분의 경우가 0이 될 것이다. 그러나 일반적으로는 사용자와 아이템에 대한 몇 가지의 속성 정보를 가지고 있는 경우가 대부분이며, 이러한 경우에는 속정 정보를 이용하여 희소성 문제를 완화시킬 수 있다.

이러한 목적으로 협동적 여과에 단순 선호도 이외의 정보를 결합한 다양한 시도가 있었다. Basu[12], Claypool[13] 등은 협동적 추천과 내용기반 추천 결과를 결합하는 방식의 복합 추천을 제안하였고, Pazzani[14]는 사용자의 내용기반 프로파일 사이의 유사도로 사용자 사이의 유사도 계산함으로써 협동적 여과와 내용기반 여과를 혼합하는 시도를 하였다. 또한 통합적인 추천값 예측 모델을 시도한 예로서 Condliff 등[15]은 베이지안 혼합 효과 모델에서 사용자의 평가 값과 사용자와 아이템의 속성 값을 통합하는 시도를 하였고, Popescul 등[16]은 three-way aspect 모델에서 내용 정보와 사용자와 아이템의 동시 발생 데이타를 결합한 통합 확률 모델을 제안한 바 있다.

본 논문에서 제안하는 방식과 가장 유사한 연구로는 데이타의 희소성을 줄이기 위해 평가 값을 생성하는 내용기반 소프트웨어 에이전트를 사용한 Good 등[17]의 연구와, 선호도가 나타난 아이템의 평가 정보와 내용 정보를 이용하여 선호도가 나타나지 않은 아이템에 대해 선호도 예측값을 생성한 후, 원시 데이타에 선호도 예측값을 추가하여 가상 데이타를 생성하고, 가상 데이타를 활용하여 협동적 추천을 수행하는 Melville 등[18]의 연구가 있다. 그러나 이들의 방법은 추천하고자 하는 아이템에 대하여 기술한 텍스트 등 아이템의 내용정보만을 사용하는데 비하여, 본 연구에서는 아이템과 사용자의 일반적인 속성들을 모두 사용하여 알려지지 않은 아이템의 선호도 값을 예측하는 통합된 방법을 제안하였으며, 이러한 통합된 방법에 의하여 원시 데이타를 변형한 후 변형된 데이타를 협동적 추천에 이용한다는 점에서 차이가 있다.

3. 데이타 블러링

3.1 데이타 블러링의 개념

많은 실질적인 추천 시스템 응용 분야에서는 선호도 데이타 외에 여러 가지의 사용자 정보와 아이템 정보가 존재한다. 예를 들어 사용자에 대해서는 성별, 나이, 직업 등 인구통계학적인 정보가 있을 수 있고, 아이템에 대해서는 영화의 장르와 같은 분류 정보 등이 있을 수 있다. 이와 같은 정보를 활용하면 데이타 희소성 문제와 초기 추천 문제를 완화시킴으로써 협동적 여과의 성능을 개선할 수 있다.

예를 들어 사용자의 선호도 데이타가 표 1과 같다고 하자. 표 1에서 'U'는 사용자를 의미하고, 'I'는 아이템을 의미하며, '1'은 사용자가 해당 아이템을 선호한다는 것을 의미한다. 표 1에 따르면 U1과 U2는 아이템들에 대하여 일치하는 선호도가 없기 때문에 유사하지 않은 사용자로 나타난다. 그리고 U5는 선호도 벡터가 0이기 때문에 유사한 사용자가 전혀 없다. 따라서 U1과 U5에 대해 협동적 여과를 기반으로 하는 추천은 불가능하다. U1의 경우는 데이타 희소성 문제의 예라고 할 수 있으며, U5의 경우는 초기 추천 문제의 예라고 할 수 있다.

이제 사용자와 아이템이 표 2와 같은 속성을 갖는다고 가정하자. 이러한 속성들을 보면 U1과 U2는 동일한 장르의 아이템들을 선호하기 때문에 서로 유사한 사용자라는 결론을 얻을 수 있으며, U3, U4, U5는 동일한 사용자 속성 값 'F'를 갖기 때문에 유사한 사용자라는 결론을 얻을 수 있다. 이러한 정보들을 적절히 사용한다면 전술한 협동적 추천의 문제들도 해결할 수 있을 뿐 아니라 추천의 정확도도 높일 수 있다.

본 논문에서 제안하는 방법은 이러한 추가적인 속성 정보를 이용하여 알려지지 않은 선호도를 채워 넣는 것이다. 이러한 방법을 데이타 블러링이라고 한다. 즉 데이타 블러링은 추가적인 속성 정보로부터 계산될 수 있는 확률분포에 기반하여 알려지지 않은 선호도 데이타를 예측하여 선호도 데이타의 희소성을 없애는 것이다.

표 1 기본적인 사용자 선호도 예제

	I1	I2	I3	I4	I5	I6
U1	1	0	1	0	0	0
U2	0	1	0	1	0	0
U3	0	0	0	0	1	1
U4	0	0	0	0	1	1
U5	0	0	0	0	0	0

표 2 속성을 추가한 사용자 선호도 데이타의 예제

	Genre	A	A	C	C	D	D
Sex		I1	I2	I3	I4	I5	I6
M	U1	1	0	1	0	0	0
M	U2	0	1	0	1	0	0
F	U3	0	0	0	0	1	1
F	U4	0	0	0	0	1	1
F	U5	0	0	0	0	0	0

표 3 블러링된 사용자 선호도 예제

	Genre	A	A	C	C	D	D
Sex		I1	I2	I3	I4	I5	I6
M	U1	1	0.5	1	0.5	0	0
M	U2	0.5	1	0.5	1	0	0
F	U3	0	0	0	0	1	1
F	U4	0	0	0	0	1	1
F	U5	0	0	0	0	0.67	0.67

기본적인 개념은 사용자가 속성 값이 X인 아이템을 선호한다면, 그 사용자는 속성 값이 X인 다른 아이템들에 대해서도 어느 정도 선호도를 갖는다고 가정하는 것이다. 표 2의 예를 보면 사용자 U1은 I1을 선호하며, I1의 장르가 A라는 것을 바탕으로 역시 장르가 A인 I2에 대한 선호도 값이 1/2이라고 예상하는 것이다. 1/2은 장르가 A인 아이템들에 대한 U1의 평균 선호도이다. 이 방법은 사용자 속성에도 적용할 수 있다. 아이템 I6은 성별이 F인 사용자에 의해 주로 선호되어진다는 것을 바탕으로 성별이 F인 U5의 아이템 I6에 대한 선호도를 2/3으로 예측할 수 있다. 2/3은 I6에 대한 성별 F인 사용자들의 평균 선호도이다. 표 3은 이러한 방식에 의한 블러링 결과를 나타낸다. U1과 U2에 대한 선호도 벡터는 아이템 속성을 사용하여 블러링되었으며, U5의 선호도 벡터는 사용자 속성 값을 사용하여 블러링되었다.

그림 1은 실험에 활용된 원시 데이타에서 사용자 100명과 아이템 100개를 랜덤하게 추출하여 선호도 정보를 시각화한 것이다. x축은 아이템을 나타내고, y축은 사용자를 나타내며, z축은 불리언으로 변환된 선호도 값을 나타낸다. 그림 1을 보면 알려진 선호도는 매우 희소함을 알 수 있다. 그림 2는 그림 1의 데이타를 블러링한 것으로서, 알려지지 않은 선호도를 예측한 후에는 선호도 데이타가 전혀 희소하지 않음을 보여준다. 이와 같이 선호도 데이타의 희소성을 없앰으로서 블러링된 데이타로 협동적 여과기반 추천을 효과적으로 수행할 수 있다.

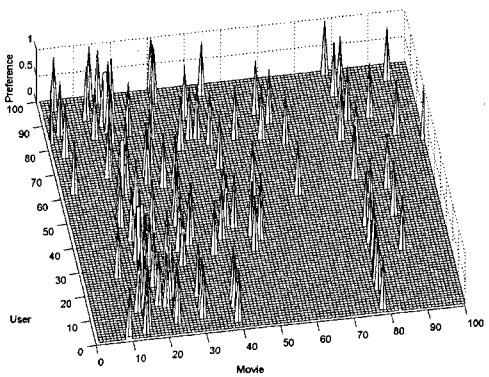

그림 1 원시 데이타

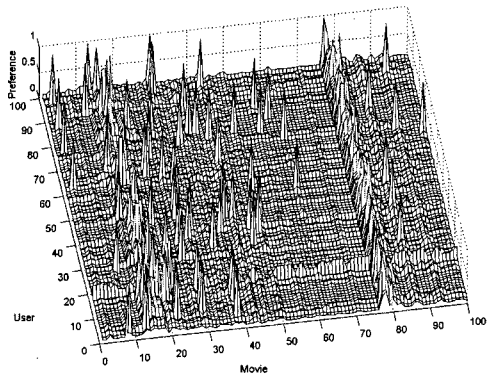

그림 2 블러링된 데이타

3.2 블러링 알고리즘

이 장에서는 본 논문에서 제안하는 속성 기반 데이타 블러링 기법을 자세히 기술한다. 블러링 계산 과정을 설명하기 위해 사용하는 기호들은 다음과 같이 정의한다.

- P는 불리언 값으로 나타난 사용자와 아이템 사이의 선호도 행렬이다. 사용자 U_i가 아이템 I_j를 선호할 경우 P_{ij}=1이고, 그 이외의 경우는 P_{ij}=0이다.
- 사용자는 속성이 $f_1, f_2, \ldots, f_n$인 n개의 속성들을 가진다.
- 아이템은 속성이 $g_1, g_2, \ldots, g_m$인 m개의 속성들을 가진다.
- 사용자 U_i의 속성 벡터는 $X_i=(x_1, x_2, \ldots, x_n)$이다. x_k는 사용자 U_i의 f_k 속성 값이다.
- 아이템 I_j의 속성 벡터는 $Y_j=(y_1, y_2, \ldots, y_m)$이다. y_k는 아이템 I_j의 g_k 속성 값이다.
- $P_{ij}=0$ 일 경우에 블러링을 적용하여 선호도 값을 채워 넣기로 한다.

P_{ij}의 값은 아이템의 속성 정보나 사용자의 속성 정보를 이용하여 예측할 수 있으며, 전자를 행-방향 블러링, 후자를 열-방향 블러링이라 한다.

행-방향 블러링은 아이템의 속성 정보를 이용하여 알려지지 않은 P_{ij}의 값을 생성한다. $SI(Y_j)$를 I_j와 같은 속성 벡터 Y_j를 갖는 아이템들의 집합이라 하자. 예를 들어, 만약 아이템 속성이 장르와 제작사이고, I_j의 속성 벡터가 Y_j=<Comedy, MGM>라면 $SI(Y_j)$는 장르가 comedy이고, 제작사는 MGM인 모든 아이템들의 집합이다. 사용자 U_i가 아이템 I_j를 선호할 확률은 사용자 U_i가 $SI(Y_j)$(속성 벡터 <Comedy, MGM>을 갖는 아이템들)를 선호할 확률과 $SI(Y_j)$ 중에서 아이템 I_j가 선호될 확률을 곱한 것이라고 할 수 있다. 따라서

사용자 U_i가 아이템 I_j를 선호할 확률을 $P(I_j|U_i)$라 하면, 이 확률은 아이템 속성의 확률분포를 이용하여 다음과 같이 계산될 수 있다.

$$P(I_j \mid U_i) = P(I_j \mid y_1, y_2, \ldots, y_m) \cdot P(y_1, y_2, \ldots, y_m \mid U_i) \quad (2)$$

$$\approx P(I_j \mid y_1, y_2, \ldots, y_m) \cdot \prod_{k=1}^{m} P(y_k \mid U_i) \quad (3)$$

$$\approx \frac{N(I_j)}{N(y_1, y_2, \ldots, y_m)} \cdot \frac{N(U_i, y_1)+1}{N(U_i)+k_1} \cdot \frac{N(U_i, y_2)+1}{N(U_i)+k_2} \cdot \ldots \cdot \frac{N(U_i, y_m)+1}{N(U_i)+k_m} \quad (4)$$

식 (3)은 아이템 속성들이 서로 독립이라는 가정에 의해 확률 값을 근사시킨 것이다. 각 확률 값들은 행렬 P에서 해당하는 1의 개수에 의해 추정된다. 식 (4)에서 $N(I_j)$는 아이템 I_j에 대한 선호도 총수이며(행렬 P에서 j번째 열의 1의 개수), $N(y_1, y_2, \ldots, y_m)$은 속성 값이 $y_1, y_2, \ldots, y_m$인 아이템에 대한 알려진 선호도 총수를 의미한다. $N(U_i)$는 사용자 U_i의 선호도의 총수를 의미하고(행렬 P에서 i번째 행의 1의 개수), $N(U_i, y_1)$는 속성 값이 y_1인 아이템에 대한 사용자 U_i의 선호도 수를 나타낸다. 확률 추정값이 0이 되는 경우를 배제하기 위하여 라플라스(Laplace) 근사값을 사용한다. 즉, k_1은 아이템 속성 g_1에 대한 서로 다른 값의 개수이다.

열-방향 블러링은 사용자의 속성 정보를 이용하여 P_{ij}의 값을 생성한다. $SU(X_i)$를 U_i와 같은 속성 벡터 X_i를 갖는 사용자의 집합이라 하자. 사용자 속성이 성별와 나이이고, U_i의 속성 벡터가 X_i=<Male, 20s>라면 $SU(X_i)$는 성별은 male이고, 나이는 20s인 모든 사용자의 집합이다. 아이템 I_j가 사용자 U_i에 의해 선호될 확률은 아이템 I_j가 $SU(X_i)$(속성 벡터 <Male, 20s>을 갖는 사용자들)에 의해 선호될 확률과 $SU(X_i)$ 중에서 사용자 U_i가 선택될확률을 곱한 것이라고 할 수 있다. 따라서 아이템 I_j가 사용자 U_i에 의해 선호될 확률을 $P(U_i|I_j)$라 하면, 이 확률은 사용자 속성의 확률분포에 의해 다음과 같이 계산될 수 있다.

$$P(U_i \mid I_j) = P(U_i \mid x_1, x_2, \ldots, x_n) \cdot P(x_1, x_2, \ldots, x_n \mid I_j) \quad (5)$$

$$\approx P(U_i \mid x_1, x_2, \ldots, x_n) \cdot \prod_{k=1}^{n} P(x_k \mid I_j) \quad (6)$$

$$\approx \frac{N(U_i)}{N(x_1, x_2, \ldots, x_n)} \cdot \frac{N(I_j, x_1)+1}{N(I_j)+l_1} \cdot \frac{N(I_j, x_2)+1}{N(I_j)+l_2} \cdot \ldots \cdot \frac{N(I_j, x_n)+1}{N(I_j)+l_n} \quad (7)$$

식 (7)에서 $N(x_1, x_2, \ldots, x_n)$은 속성 값이 $x_1, x_2, \ldots, x_n$인 사용자들의 알려진 선호도의 총수를 의미하며, $N(I_j, x_1)$는 속성 값이 x_1인 사용자들이 아이템 I_j를 선호한 총수를 나타낸다.

행-방향 블러링과 열-방향 블러링의 결과로 최종 값 $Blurring(i, j)$을 아래와 같이 결정할 수 있으며, α_r과 α_c는 각각 행-방향 블러링 인수와 열-방향 블러링 인수이다.

$$Blurring(i, j) = \alpha_r \cdot P(I_j \mid U_i) + \alpha_c \cdot P(U_i \mid I_j) \quad (8)$$

모든 알려지지 않은 선호도 값을 예측하고 나면, 변형된 선호도 행렬을 이용하여 협동적 여과를 수행한다.

3.3 예 제

이 장에서는 위에서 제시한 블러링 계산의 예를 설명한다. 표 4는 원시 데이타의 예로서, 사용자의 아이템에 대한 선호도를 나타내고 표 5는 원시 데이타에 나타난 선호도 값들을 이용하여 사용자 유사도를 계산한 것이다. 표에서 U1과 U2, U1과 U3, U2와 U3는 공통된 선호도가 없기 때문에 사용자 유사도는 모두 0으로 나타난다.

표 4 원시 데이타 예제

		Genre	C	C	C	A
		Director	w	x	y	z
Sex	Age		I1	I2	I3	I4
M	20	U1	0	0	0	1
M	30	U2	1	0	1	0
M	30	U3	0	1	0	0
F	30	U4	1	1	0	1

표 5 원시 데이타에서의 사용자 유사도

	U1	U2	U3	U4
U1	1	0	0	0.58
U2	0	1	0	0.41
U3	0	0	1	0.58
U4	0.58	0.41	0.58	1

표 6은 행-방향 블러링과 열-방향 블러링에 대해 블러링 인수를 0.5로 하여 원시 데이타를 블러링한 결과를 나타낸다. 사용자 속성은 성별과 나이를 이용하여 블러링하고, 영화 속성은 장르와 감독을 이용하여 블러링한 것이다. 예를 들어 U2의 I2에 대한 선호도는 식 (4), 식 (7), 식 (8)에 의해 다음과 같이 계산된다.

P(I2|U2) = P(I2|C, x) P(C, x|U2) = P(I2|C, x) P(C|U2) P(x|U2)

= 2/2 * (2+1)/(2+2) * (0+1)/(2+4) = 1/8

P(U2|I2) = P(U2|M,30) P(M,30|I2) = P(U2|M, 30) P(M|I2) P(30|I2)

= 2/3 * (1+1)/(2+2) * (2+1)/(2+2) = 1/4

Blurring(2,2) = 0.5 * 1/4 + 0.5 * 1/8 = 0.188

표 6에 나타낸 블러링 데이타를 이용하여 사용자 유사도를 계산하면 표 7의 결과를 얻을 수 있다. 표 7에 나타난 바와 같이 블러링을 수행함으로써 공통된 선호도가 나타나지 않은 사용자들 사이에 연관성이 나타나게 되어 사용자 유사도를 구할 수 있다. 사용자 U2와 U3는 원시 데이타에서 유사도는 0이지만, 원시 데이타를 블러링하여 유사도를 측정하면 U2와 U3의 유사도는 0.32가 된다. U2와 U3는 장르가 C인 아이템을 선호하고, U2와 U3 모두 나이가 30대인 사실이 사용자 유사도 계산에 반영된 것이다.

표 6 블러링 데이타

Genre			C	C	C	A
Director			w	x	y	z
Sex	Age		I1	I2	I3	I4
M	20	U1	0.096	0.144	0.258	1
M	30	U2	1	0.188	1	0.104
M	30	U3	0.129	1	0.141	0.075
F	30	U4	1	1	0.154	1

표 7 블러링 데이타에서의 사용자 유사도

	U1	U2	U3	U4
U1	1	0.32	0.25	0.70
U2	0.32	1	0.32	0.58
U3	0.25	0.32	1	0.69
U4	0.70	0.58	0.69	1

4. 성능 실험

본 장에서는 제안된 블러링 기법에 대한 실험 결과를 기술한다. 각 실험에서는 원시 데이타를 활용한 추천 정확도와 블러링 데이타를 활용한 추천 정확도를 비교한다.

4.1 실험 방법

실험에서 사용한 실험 데이타는 DEC(Digital Equipment Corporation)사의 EachMovie 데이타이다. EachMovie 데이타는 웹 기반 영화 추천 사이트에서 1996년부터 18개월 동안 72,916명의 사용자가 1,628개의 영화들에 대해 선호도 평가를 수행한 것으로 사용자의 명시적 평가가 이루어진 데이타 집합이다. 사용자들은 0.0부터 1.0까지의 범위에서 6단계로 된 평가를 수행하였으며, 평가된 총수는 2,811,983건이다. EachMovie 데이타는 사용자 id, 나이, 성별과 같은 사용자 정보를 포함하고 있으며, 영화 제목, 장르, 웹 주소 등의 아이템 정보를 포함하고 있다. 장르에는 action, animation, art-foreign, classic, comedy, drama, family, horror, romance, thriller로 총 10개의 영화 장르들이 존재한다.

실험을 위해 다음 절차에 따라 평가 값을 불리언 값으로 변환하였다. 각 사용자의 평가 값의 평균을 계산한 후, 평균보다 큰 평가 값은 1로 변환하고, 나머지는 0으로 변환하였다. 따라서 변환된 데이타 집합은 오직 양성의 선호도 정보를 갖는다. 즉 데이타 집합에서 '1'은 '선호한다'는 의미이고, '0'은 '선호도를 알 수 없다'의 의미이다. 이러한 변형은 아이템을 구매하는 행위나 웹 페이지를 클릭하는 행위 등 묵시적인 양성 평가만 얻어질 수 있는 일반적인 환경에서 제안된 기법을 테스트하기 위하여 수행되었다.

모든 실험에서 데이타 집합은 10개의 동일한 크기의 사용자 집합들로 나누어, 하나의 집단은 테스트 집합으로 사용하고, 다른 9개의 집단은 훈련 집합으로 사용하였고, 이와 같은 실험을 모든 10개의 집합에 대해 반복하여 결과를 평균하였다(10-fold cross validation). 블러링 인수는 행-방향 블러링과 열-방향 블러링 모두 0.5로 하였다. 블러링 데이타에서 추천 정확도는 다음과 같이 측정하였다.

테스트 집합의 각 사용자에 대하여 알려진 선호도를 하나씩 제외한 상태에서 블러링을 수행하고 협동적 추천에 의해 top-N 추천 아이템을 구한 후 제외된 선호도가 추천 리스트에 포함되는가를 보았다. 즉,

step 1. 속성 정보를 이용하여 훈련 집합을 블러링한다(선호도 데이타에 나타난 모든 '0'을 블러링 값으로 채운다-식 (8)). 아이템 속성은 장르를 이용하고, 사용자 속성은 나이, 성별을 이용한다.

step 2. 테스트 집합의 각 사용자에 대해 각각의 '1'을 차례로 제거하면서('1'을 '0'으로 변환한다) step 3~7을 수행한다. 제거된 '1'에 해당하는 아이템을 '목표 아이템'이라 한다.

step 3. 훈련 집합에서 획득된 속성들의 확률분포를 이용하여 테스트 사용자의 선호도 벡터를 블러링한다.

step 4. 훈련 집합에 있는 모든 사용자와 테스트 사용자 사이의 유사도를 계산한다. 유사도 계산에는 벡터 유사도를 사용한다.

step 5. 가장 유사한 K명의 사용자를 선택하고, K명의 유사 사용자의 선호도 벡터의 가중평균을 이용하여 알려지지 않은 선호도(블러링 이전에 0인 선호도)에 대해 아이템 선호도를 예측한다.

step 6. 높은 예측값을 갖는 상위 N개의 아이템을 찾아서 추천한다.

step 7. 상위 N개의 아이템 리스트에서 '목표 아이템'

을 찾았다면 추천은 성공한 것이다. 이것을 '적중'이라 한다.

최종적인 추천의 정확도는 다음과 같은 적중률로 나타낸다. 이러한 적중률을 K와 N의 값을 변화시키면서 구하여 블러링되지 않은 원시 데이타에 의한 추천 성능과 블러링된 데이타에 의한 추천 성능을 비교 실험하였다.

$$Hit\ ratio = \frac{total\ number\ of\ hits}{total\ number\ of\ tests(=total\ number of\ '1')} \quad (9)$$

4.2 실험 결과

성능 측정을 위해 수행한 실험은 총 3가지로 구성된다. 첫 번째는 데이타 블러링에 의한 일반적인 성능 향상 측정을 위한 실험이고, 두 번째는 데이타 희소성에 따른 블러링의 효과에 대한 실험이며, 세 번째 실험은 초기 추천에 대한 실험이다.

4.2.1 일반적인 성능 향상

본 실험에서는 EachMovie 데이타에서 1000명의 사용자를 랜덤하게 추출하고 여러 장르에 속하는 아이템들을 제외하여 1000×1000 사이즈의 실험용 데이타 집합을 설정하였다. 선호도를 불리언 값으로 변환한 후 사용자의 아이템에 대한 선호도 총수는 15,226개로 전체 데이타에서 1.5%를 점유한다. 선호도 값들은 고른 분포를 이루지 않으며, 특정한 아이템들이 많은 선호도를 차지하고 있다. 사용자들이 많이 선호하는 아이템을 기준으로 top-10에 해당하는 아이템들이 전체 선호도 데이타의 16%, top-20에 해당하는 아이템들이 전체 선호도 데이타의 28%를 차지한다.

표 8은 이러한 데이타를 가지고 수행한 원시 데이타와 블러링된 데이타에서의 top-N 추천 적중률 실험 결과이다. 그림 3은 원시 데이타에서 K값의 변화에 따른 추천 성능 누적 그래프이고, 그림 4는 블러링 데이타에서 K값의 변화에 따른 추천 성능 누적 그래프이다.

그림 3과 그림 4를 보면 대체로 K(유사 사용자 수)가 증가할수록 추천 정확도는 높아지지만, K=50 이상이 되면 정확도는 더 이상 증가하지 않고 오히려 top-10 추천에서는 정확도가 감소한다는 것을 알 수 있다. 그림 3의 원시 데이타에서 나타난 결과를 보면 K=1에서는 N이 증가하더라고 추천 정확도 향상이 크게 나타나지 않

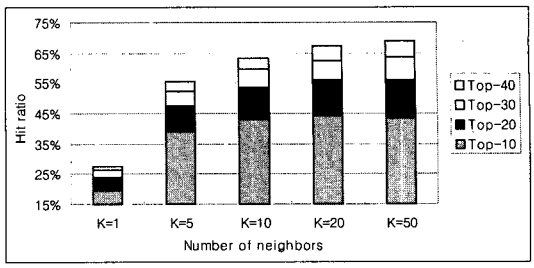

그림 3 원시 데이타에서 추천 성능 누적 그래프

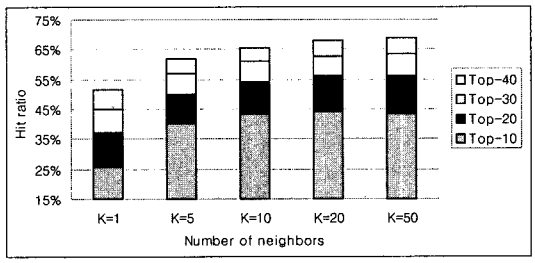

그림 4 블러링된 데이타에서 추천 성능 누적 그래프

고 있다. 이는 원시 데이타에서의 선호도 데이타가 희소하여(1.5%) 단일한 유사 사용자만을 이용하는 경우 목표 아이템이 아예 추천 대상이 되지 않는 경우가 많기 때문이다. 그림 4의 블러링 데이타에서는 K=1에서도 N의 증가에 따라 성능 향상이 크게 나타난다. 이와 같은 결과는 데이타를 블러링함으로써 단일한 유사 사용자만으로도 목표 아이템이 추천될 수 있기 때문이다. K=1일 때 블러링 데이타는 원시 데이타에 비하여 top-10 추천에서 6.1%, top-40 추천에서 23.9%의 성능 향상을 나타내었다.

그림 5는 top-20 추천에서 원시 데이타와 블러링 데이타의 추천 정확도를 비교한 것으로서 블러링 데이타를 활용한 추천이 원시 데이타를 활용한 추천에 비해 모든 경우에서 더 높은 적중률을 나타낸 것을 보여준다. 성능의 향상은 K 값이 작을수록 높게 나타났으며, 이는 블러링된 데이타에는 알려지지 않은 선호도가 존재하지 않으므로 적은 유사 사용자를 활용하여도 모든 아이템에 대한 추천 값을 얻을 수 있기 때문이다.

그림 6은 원시 데이타와 블러링 데이타에서의 목표

표 8 원시 데이타와 블러링된 데이타에서의 적중률(1000X1000 데이타 집합)

K	N=10		N=20		N=30		N=40	
	Original	Blurred	Original	Blurred	Original	Blurred	Original	Blurred
1	19.5	25.6	24.1	37.1	26.4	45.1	27.6	51.5
5	39.1	40.0	47.5	50.1	52.5	56.7	55.7	61.7
10	43.0	43.3	53.3	54.2	59.5	61.0	63.4	65.6
20	44.1	44.2	55.9	56.1	62.5	62.8	67.4	68.0
50	43.5	43.4	56.1	56.3	63.6	63.7	68.8	68.8

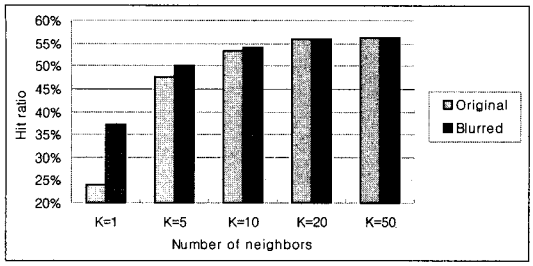

그림 5 Top-20에서 원시 데이타와 블러링 데이타의 적중률 비교

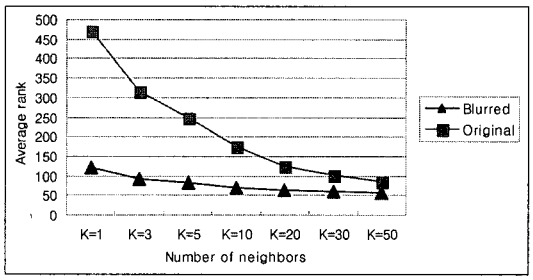

그림 6 K값에 따른 목표 아이템의 평균 추천 순위

아이템의 평균 추천 순위를 나타낸 그래프이다. K=1에서 원시 데이타에서의 평균 순위는 471을 나타내고, 블러링 데이타를 이용한 추천에서는 평균 순위가 122이다. K=20 이상에서 top-N 적중률에 있어서는 많은 차이가 나지 않지만 평균 순위에서는 현저한 차이를 보임으로써 일반적으로 블러링 데이타를 이용했을 때 목표 아이템을 상위에 추천하게 됨을 알 수 있다.

4.2.2 희소성에 따른 성능 향상의 차이

협동적 여과에서의 중요한 문제 중에 하나는 희소성 문제이다. 만약 매우 적은 양의 선호도 값이 존재한다면 유사 사용자들을 추출하기 어렵기 때문에 추천 정확도는 낮아진다. 이러한 경우 데이타 블러링을 희소성이 높은 선호도 데이타에 적용하면 추천 정확도 향상에 매우 효과적이다.

본 실험을 위해 세 개의 서로 다른 희소성을 갖는 500×500 크기의 데이타 집합을 생성하였다. 세 개의 실험 데이타 집합은 각각 아이템을 2개만 선호한 경우, 3개만 선호한 경우, 4개만 선호한 경우에 해당하는 500명의 사용자들을 랜덤하게 추출하고, 아이템은 선호도가 많은 순으로 500개를 추출하여 생성하였다. 이 실험용 데이타 집합들은 매우 적은 수의 선호도만을 알고 있는 사용자들로 구성되어 데이타 희소성이 매우 높은 집단이다. 표 9, 10, 11은 각각 선호도가 4개, 3개, 2개만 있는 사용자 집합에서의 추천 성능 실험 결과이며, 그림 7, 8, 9는 top-20 추천에서 원시 데이타와 블러링 데이타의 추천 정확도를 비교한 것이다.

표 9~11 결과를 보면 희소성이 더 낮은 표 8의 결과보다 대부분의 경우에 오히려 높은 적중률을 나타내는데 이는 일반적인 현상이 아니라 EachMovie 데이타의 경우 희소한 데이타일수록 선호도가 적은 수의 특정한 아이템에 집중되어 있기 때문에 나타나는 현상이다.

그림 7~9를 보면 모든 경우에 있어서 블러링 데이타를 활용한 경우가 우수한 결과를 나타낸다. 그림 5의 그래프와 비교하면 원시 데이타와 블러링 데이타의 정확도 차이가 더 크며, 특히 그림 7, 8, 9로 갈수록 높은 K

표 9 선호도 값을 4개만 소유한 사용자 집합에서의 적중률 (500×500)

K	N=10		N=20		N=30		N=40	
	Original	Blurred	Original	Blurred	Original	Blurred	Original	Blurred
1	22.4	60.2	22.4	67.30	22.4	77.5	22.4	82.0
3	42.0	60.2	42.0	68.40	42.0	77.7	42.0	82.1
5	50.2	60.3	50.5	69.15	50.5	78.5	50.5	81.9
10	57.9	58.2	60.8	69.20	60.9	78.1	61.0	81.9
20	61.0	58.0	67.9	71.15	68.9	77.5	67.0	82.4
50	61.9	61.4	71.1	72.90	76.4	79.2	77.3	83.6

표 10 선호도 값을 3개만 소유한 사용자 집합에서의 적중률 (500×500)

K	N=10		N=20		N=30		N=40	
	Original	Blurred	Original	Blurred	Original	Blurred	Original	Blurred
1	17.8	42.9	17.8	67.3	17.8	76.3	17.8	81.9
3	36.7	44.9	36.7	68.8	36.7	77.6	36.7	83.7
5	44.8	43.7	44.9	70.5	44.9	77.7	44.9	83.8
10	55.8	44.5	57.5	69.8	57.5	77.8	57.5	83.7
20	56.1	51.5	60.4	67.7	60.7	78.5	60.7	84.1
50	53.2	57.6	58.3	72.1	59.8	79.4	60.5	84.6

표 11 선호도 값을 2개만 소유한 사용자 집합에서의 적중률 (500×500)

K	N=10		N=20		N=30		N=40	
	Original	Blurred	Original	Blurred	Original	Blurred	Original	Blurred
1	11.8	75.2	11.8	84.1	11.8	88.7	11.8	92.1
3	43.2	74.9	43.2	84.5	43.2	89.7	43.2	92.2
5	54.0	74.9	54.0	84.4	54.0	90.1	54.0	92.3
10	64.0	74.9	64.0	84.1	64.0	89.8	64.0	92.8
20	63.5	76.2	63.6	84.0	63.6	90.3	63.6	93.1
50	60.4	76.4	61.4	84.0	61.4	90.2	61.4	93.1

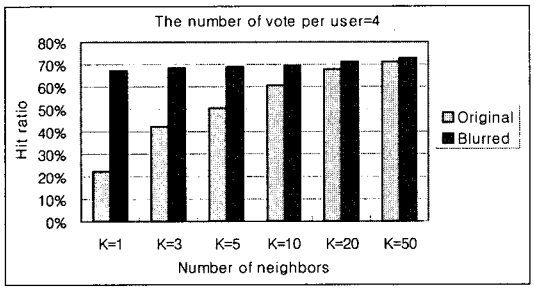

그림 7 Top-20에서 선호도 값을 4개만 소유한 사용자 집합에서의 적중률

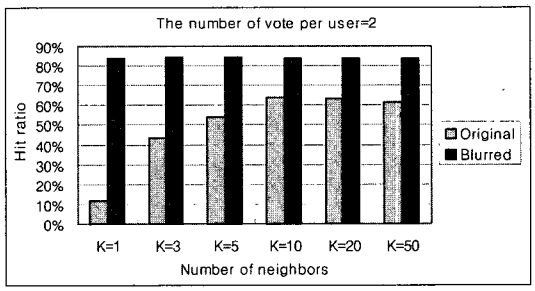

그림 9 Top-20에서 선호도 값을 2개만 소유한 사용자 집합에서의 적중률

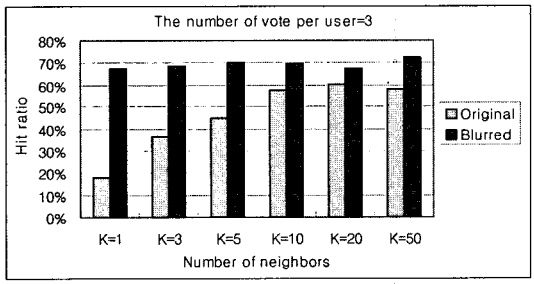

그림 8 Top-20에서 선호도 값을 3개만 소유한 사용자 집합에서의 적중률

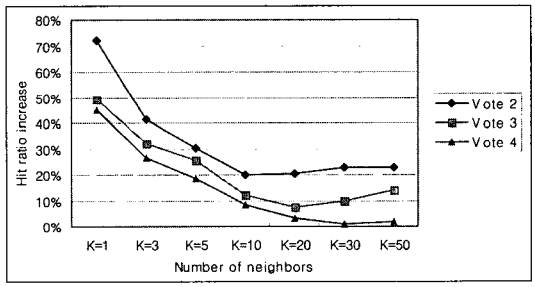

그림 10 희소성에 따른 top-20 추천의 적중률 증가

값에 대해서도 정확도의 차이가 크게 나타나는 것을 볼 수 있다. 선호한 아이템 수량이 4개, 3개, 2개로 줄어가면서 원시 데이타와 블러링 데이타를 활용한 추천에서 적중률이 가장 높은 K=50의 경우 적중률 차이를 비교하면 N=20일 경우 vote 4(사용자 선호도 수 4개)에서는 1.8%, vote 3에서는 11.7%, vote 2에서는 20.5%를 나타낸다.

그림 10은 각각의 데이타 집합을 이용한 실험에서 top-20 추천의 적중률의 증가분을 요약한 것이다. 일반적으로 K값이 작을수록 성능 향상이 크며, 모든 K에 대해서 데이타의 희소성 높을수록 성능 향상이 크다는 것을 보여준다. 이 결과는 제안한 데이타 블러링 기법이 데이타가 희소한 경우에 더욱 효과적이라는 것을 나타낸다.

4.2.3 초기 추천 문제에 대한 실험

초기 추천 문제는 협동적 여과기반 추천 시스템에서 희소성 문제와 함께 또다른 중요한 문제이다. 새로운 사용자와 같이 알려진 선호도 값이 존재하지 않을 때는 협동적 여과기반 추천 시스템은 유사 사용자들을 추출할 수 없기 때문에 아이템을 전혀 추천할 수 없다. 그러나 사용자와 아이템의 속성 정보를 기반으로 새로운 사용자의 선호도 벡터를 블러링한다면 협동적 여과기반 추천 시스템을 이용한 아이템 추천이 가능하다.

본 실험에서는 새로운 사용자를 생성하기 위해 테스트 집합에 소속된 사용자들의 선호도 를 모두 없애고('1'을 '0'으로 변환), 새로운 사용자의 선호도 벡터를 블러링한다. 블러링된 선호도 벡터를 이용하여 훈련 집합에서 유사 사용자들을 추출한 후 아이템들의 선호도를 예측하여 각 목표 아이템(변환 이전에 '1'이었던 아이템)의

적중률을 계산한다. 비교 대상인 원시 데이타는 0 벡터로서 협동적 여과를 적용할 수 없기 때문에 이 실험에서는 모든 새로운 사용자에게는 가장 인기있는 (사용자로부터 많이 선호된) top-N 아이템을 추천하는 방식을 취한다. 즉, 블러링 데이타에서는 추천 리스트 중 top-N을 추천하고 원시 데이타에서는 선호도가 가장 많은 top-N 아이템을 추천하도록 하였다. 실험 데이타는 4.2.1절의 실험에서 사용한 것과 같은 1000X1000 데이타를 사용하였다.

표 12와 그림 11은 새로운 사용자에 대한 원시 데이타와 블러링 데이타의 실험 결과이다. 선호도가 인기 있는 소수의 영화에 집중되는 경향이 있기 때문에 항상 인기 순위 top-20 아이템을 추천하여도 약 28%의 적중률을 얻을 수 있다. 실험 결과를 보면 K가 5보다 큰 경우에는 새로운 사용자를 블러링하여 아이템을 추천받은 결과가 인기도가 높은 상위 10개와 상위 20개를 추천한 결과보다 우수하게 나타난다. 이러한 결과는 새로운 사용자에 대한 아이템 추천 역시 데이타를 블러링함으로써 가능할 수 있다는 것을 나타낸다.

표 12 초기 추천에 대한 적중률

K	N=10		N=20	
	Blurred	Popular10	Blurred	Popular20
1	5.69	14.58	11.39	28.00
5	20.00	14.58	28.94	28.00
10	19.66	14.58	29.28	28.00
20	20.09	14.58	30.18	28.00
50	18.91	14.58	29.22	28.00

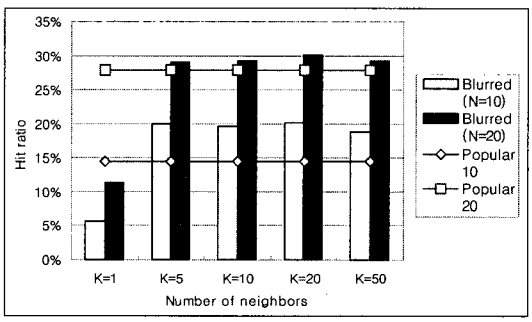

그림 11 초기 추천에 대한 적중률 그래프

4.2.4 기타

데이타 블러링에서 문제가 될 수 있는 것은 사용자의 수가 매우 많을 경우 저장하여야 할 블러링 데이타의 수가 많아지고 계산 시간이 늘어난다는 것과 사용자의 새로운 선호도 정보가 추가될 때마다 블러링을 위한 속성 기반 확률 추정 값이 변하므로 전체 데이타의 블러링 값을 새로 계산하여야 한다는 것이다. 이와 같은 문제점을 해결할 수 있는 하나의 방법은 적절한 양의 샘플 데이타로 블러링을 위한 확률 추정 값들을 계산한 뒤 추천 대상 사용자의 선호도 벡터만을 블러링하여 협동적 추천을 수행하는 것이다.

표 13은 훈련 집합에서의 속성 확률분포를 이용하여 테스트 집합의 추천 대상 사용자의 선호도 벡터만 블러링하고 전체 선호도 데이타는 블러링하지 않는 경우의 추천 성능 실험 결과이다. 추천 대상자만 블러링한 경우에도 모든 데이타를 블러링한 경우와 비교하여 0~1.3%의 적은 적중률 차이만을 나타내어 추천 정확도에 큰 영향을 주지 않음을 알 수 있다. 즉 웹에서의 응용과 같이 블러링을 위한 데이타 갱신이 문제가 되는 경우에는 추천 대상 사용자의 선호도만 블러링함으로써 문제를 해결 할 수 있다.

마지막으로 블러링 인수의 영향을 알아보기 위하여 인수를 0.1부터 5.0까지 단계적으로 조정하면서 추천 정확도를 실험하였다. 그림 12는 다양한 인수 값 설정에 따른 추천 정확도를 나타낸다. 블러링 인수가 1 미만인 경우에는 블러링의 영향을 높일 수록 정확도가 약간 상

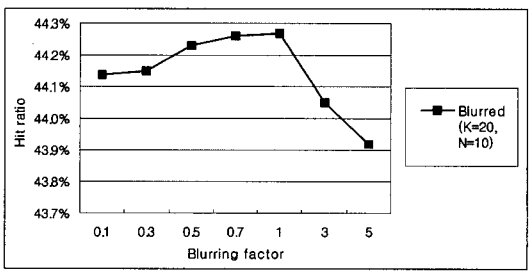

그림 12 블러링 인수 변화에 따른 적중률

표 13 추천 대상자만 블러링한 경우의 적중률

	N=10		N=20		N=30		N=40	
	Blurred-Test only	Blurred-All	Blurred-Test only	Blurred-All	Blurred-Test only	Blurred-All	Blurred-Test only	Blurred-All
K=1	25.0	25.6	36.1	37.1	43.8	45.1	50.2	51.5
K=5	39.8	40.0	49.9	50.1	56.3	56.7	61.1	61.7
K=10	43.3	43.3	54.1	54.2	60.9	61.0	65.4	65.6
K=20	44.2	44.2	56.0	56.1	62.8	62.8	67.9	68.0
K=50	43.4	43.4	56.3	56.3	63.7	63.7	68.8	68.8

승하지만 블러링 인수가 1을 초과하기 시작하면 적중률은 하락한다. 이는 블러링에 의한 선호도 예측치를 알려지지 않은 선호도에 과중하게 적용하면 원시 데이타에 노이즈로 작용될 수도 있다는 것을 뜻한다. 따라서 블러링을 적용할 때는 알려지지 않은 선호도 데이타가 블러링이 적용된 후에 알려진 선호도 값의 최대치를 넘지 않는 범위에서 데이타의 특성에 맞는 블러링 인수를 선정하여야 한다.

5. 결론 및 향후 연구

협동적 여과 방식은 추천 시스템에 성공적으로 적용된 추천 기법이다. 그러나 협동적 여과 방식은 선호도 데이타가 희소한 경우 추천 성능이 저하되며, 사용자의 선호도 정보가 존재하지 않는 초기 추천 문제에 있어서는 전혀 아이템을 추천을 할 수 없는 문제를 유발한다.

본 논문에서는 협동적 여과기반 추천 시스템의 문제점인 데이타 희소성 문제와 초기 추천 문제를 해결하기 위해 사용자와 아이템의 추가 속성 정보의 확률분포를 이용하여 알려지지 않은 선호도 데이타에 선호도 값을 부여하는 데이타 블러링 기법을 제안하였다. 데이타 블러링 기법은 사용자의 추가 속성 정보와 아이템의 추가 속성 정보의 확률분포 값을 알려지지 않은 선호도 값에 적용함으로써 데이타의 희소성을 완화시키며, 데이타의 희소성 완화로 소수의 유사 사용자를 이용한 경우에도 높은 추천 성능을 나타낸다. 실험 결과에서 확인한 것과 같이 데이타 블러링 기법을 협동적 여과기반 추천에 적용함으로써 초기 추천 문제와 데이타의 희소성 문제를 완화할 수 있다. 데이타 블러링 기법은 희소성이 높은 데이타일 경우에 더욱 큰 효과를 나타내었으며, 적은 유사 사용자들을 활용한 경우에도 높은 추천 성능을 나타내었다. 또한 선호도 정보가 전혀 없는 새로운 사용자의 경우에도 단순 인기 순위 추천에 비하여 높은 추천 정확도를 얻을 수 있었다. 따라서 본 연구에서 제안하는 방법은 실제 데이타 희소성이 문제가 되는 일반적인 전자 상거래 시스템 등에 효과적으로 적용될 수 있을 것이다.

향후 연구로서 아이템 및 사용자 속성의 중요도에 따라 블러링 인수를 조정하는 방법에 관한 연구와, 응용분야마다 다를 수 있는 여러 속성들의 다양한 확률분포 특성에 따른 데이타 블러링의 효과를 분석하기 위하여 가상 데이타를 이용한 실험이 필요하다.

참 고 문 헌

[1] R. Armstrong, D. Freitag, T. Joahims, and T. Mitchell, "WebWatcher: A Learning Apprentice for the World Wide Web," *Proceedings of the 12th National conference on Artificial Intelligence*, 1995.

[2] M. Balabanovic, and Y. Shoham, "Fab : Content-Based Collaborative Recommender," *Recommendation Communications of the ACM*, Vol.40, No.3, pp.66-77, 1997.

[3] H. Lieberman, "Letizia : An Agent That Assists Web Browsing," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[4] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting web sites," *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[5] B. Krulwich, "Lifestyle Finder: Intelligent user profiling using large-scale demographic data," *Artificial Intelligence Magazine*, Vol.18, No.2, 1997.

[6] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," *Proceedings of the 15th International Conference on Machine Learning*, Wisconsin, 1998.

[7] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," *In Proceedings of ACM SiGIR-99*, 1999.

[8] J. Konstan, B. Millr, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying Collaborative Filtering to Usenet News," *Communications of the ACM*, Vol.40, No.3, pp.77-87, 1997.

[9] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'," *Proceedings of the Conference of Human Factors in Computing Systems*, 1995.

[10] L. Terveen, W. Hill, B. Amenta, D. McDonald, and J. Creter, "PHOAKS: A System for Sharing Recommendations," *Communications of the ACM*, march 1997.

[11] J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, 1998.

[12] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as Classification: Using Social and Content-Based Information in Recommendation," *Proceedings of the 15th National Conference on Artificial Intelligence*, 1998.

[13] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes and M. Sartin, "Conbining Content-Based and collaborative Filters in an Online Newspaper," *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, 1999.

[14] M. Pazzani, "A Framework for Collaborative, Content-based and Demographic Filtering," Artificial Intelligence Review, pp.393-408, 1999.

[15] M. Condliff, D. Lewis, D. Madigan, and C. Posse,

"Baysian Mixed-Effect Models for Recommender Systems," *Proceedings of Recommender Systems Workshop at SIGIR-99*, 1999.

[16] A. Poposcul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments," *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, 2001.

[17] N. Good, J. B. Shafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations," *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.

[18] P. Melville, R. Mooney, and R. Nagarajan, "Content-Boosted Collaborative Filtering," *Proceedings of the SIGIR-2001 Workshop on Recommender Systems*, 2001.

[19] Hyungil Kim, Juntae Kim, and J. L. Herlocker, "Feature-Based Prediction of Unknown Preferences for Nearest-Neighbor Collaborative Filtering," *Proceedings of the 4th IEEE International Conference on Data Mining*, 2004.

[20] J. Ruker and M. J. Polanco, "Siteseer: Personalized Navigation for The Web," *Communications of the ACM*, Vol.40, No.3, 1997.

[21] J. Schafer, J. Konstan, and J. Riedl, "Recommender System in E-Commerce," *Proceedings of the ACM Conference on Electronic Commerce*, 1999.

김 형 일

1996년 목원대학교 수학과 졸업(이학사) 1996년~1998년 (주)경기은행. 2001년 동국대학교 대학원 컴퓨터공학과(공학석사). 2004년 동국대학교 대학원 컴퓨터공학과(공학박사). 2005년~현재 동국대학교 컴퓨터공학과 IT분야 교수요원(정보통신부). 관심분야는 지능형 에이전트, 정보검색, 기계학습, 게임

김 준 태

1986년 서울대학교 제어계측공학과 졸업(공학사). 1990년 미국 Univ. of Southern California, Electrical Engineering-Systems(M.S.). 1993년 미국 Univ. of Southern California, Computer Engineering(Ph.D.). 1994년~1995년 미국 Southern Methodist University, Computer Science and Engineering(Postdoc). 1995년~현재 동국대학교 컴퓨터공학과 부교수. 관심분야는 기계학습, 데이타마이닝, 정보검색, 지능형 에이전트