협업 필터링 추천시스템에서의 취향 공간을 이용한 평가 예측 기법

(Improving Collaborative Filtering with Rating Prediction Based on Taste Space)

이 형 동 † 김 형 주 ††
(Hyungdong Lee) (Hyoungjoo Kim)

요 약 협업 필터링은 정보 과잉 문제를 해결하기 위한 정보 필터링의 주요 기법이며, 전자 상거래 분야에서 추천 시스템과 같은 응용 프로그램에서 널리 사용된다. 협업 필터링 시스템은 사용자들의 대상 항목에 대한 평가를 수집한 후 취향이 서로 비슷한 사용자들의 의견을 바탕으로 아직 평가되지 않은 항목에 대해 예측을 수행한다. 시스템의 예측 성능은 사용자들에 의해 공통적으로 평가된 항목들의 개수에 좌우된다. 그러므로 대상 항목들이 수시로 추가되거나 제거되는 동적 컬렉션의 경우 협업 필터링 알고리즘을 그대로 적용하기 어렵다. 본 논문에서는 동적 컬렉션에 대한 협업 필터링 적용 방법을 제시한다. 제안한 방법에서는 SVD 기법을 이용하여 항목들의 취향 공간을 생성한 후 과거 항목들과 새로운 항목들 간의 연관성을 구하기 위해 핵심 항목들의 클러스터를 구성한다. 이를 평가하기 위해서 사용자 평가 데이타베이스를 시간에 의해 두 부분으로 나누고, 동적으로 추가되는 상황을 시뮬레이션하여 시스템의 예측 성능을 분석했다. 이를 통해 본 방법이 동적 컬렉션에 효과적으로 적용됨을 보인다.

키워드 : 협업 필터링, 추천 시스템, 정보 필터링

Abstract Collaborative filtering is a popular technique for information filtering to reduce information overload and widely used in application such as recommender system in the E-commerce domain. Collaborative filtering systems collect human ratings and provide predictions based on the ratings of other people who share the same tastes. The quality of predictions depends on the number of items which are commonly rated by people. Therefore, it is difficult to apply pure collaborative filtering algorithm directly to dynamic collections where items are constantly added or removed. In this paper we suggest a method for managing dynamic collections. It creates taste space for items using a technique called Singular Vector Decomposition (SVD) and maintains clusters of core items on the space to estimate relevance of past and future items. To evaluate the proposed method, we divide database of user ratings into those of old and new items and analyze predicted ratings of the latter. And we experimentally show our method is efficiently applied to dynamic collections.

Key words : Collaborative filtering, Recommender system, Information filtering

1. 서 론

인터넷과 WWW이 폭발적으로 성장함에 따라 정보 과잉 문제가 대두 되었다. 이러한 문제점을 해결하기 위해 사용자에게 불필요한 정보를 제거해 필요한 정보만을 제공해주는 정보 필터링이 제안되었고, 추천 시스템과 같은 응용 프로그램이 그 역할을 수행하게 되었다. 예를 들면, Amazon.com(www.amazon.com)과 같은 온라인 서점에서는 사용자들의 행동패턴을 분석하여 해당 사용자가 선호할 것 같은 책을 예측하여 추천해준다. 또한 추천시스템은 영화, 뉴스, 쇼핑몰의 제품 추천 등 다양한 분야에 걸쳐 적용되고 있고 점차 그 분야를 넓혀가고 있다. 이러한 정보 필터링 및 추천을 위해 널리 사용되는 기법으로 협업 필터링(collaborative filtering)이 있다. 협업 필터링을 이용한 추천 시스템에서 사용자들

· 본 연구는 BK-21 정보기술사업단과 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성, 지원사업(IITA-2005-C1090-0502-0016)의 연구결과로 수행되었음
† 학생회원 : 서울대학교 컴퓨터공학과
hdlee@idb.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학과 교수
hjk@snu.ac.kr
논문접수 : 2003년 10월 20일
심사완료 : 2007년 5월 20일

은 항목들에 대해 선호도 평가 점수를 부여한다. 그러면 시스템은 사용자들의 평가 점수를 이용해서 선호도에 따른 사용자들 간의 유사도를 구하고, 특정 사용자와 유사한 취향을 가진 다른 사용자들의 평가 정보를 바탕으로 아직 평가하지 않은 항목들에 대한 평가 점수를 예측한다. 협업 필터링은 사람이 부여하는 해당 항목에 대한 질적인 측면과 취향에 의해 내린 평가를 기반으로 하기 때문에 컴퓨터에 의해 자동으로 분석된 정보를 이용하는 방식보다 일반적으로 좋은 성능을 보인다.

영화나 도서와 같은 분야의 경우 새로운 항목이 추가되는 빈도가 비교적 낮고 추가된 항목이 오랫동안 지속된다. 반면 TV 프로그램이나 뉴스와 같은 분야의 경우 새로운 항목의 추가가 빈번하고 시간의 흐름에 따라 항목의 가치가 급격하게 떨어진다. TV 프로그램의 경우를 생각해보면 지나간 프로그램에 대한 추천은 무의미하다. 이와 같이 동적으로 새로운 항목이 계속 추가되고 시간이 지나면 가치가 사라져버리는 특성을 가진 항목들을 동적 컬렉션(dynamic collection)이라고 한다.

협업 필터링 기법은 동일 항목에 대한 여러 사용자들의 평가 점수를 바탕으로 사용자의 취향을 분석하고 아직 평가되지 않은 항목에 대한 평가 점수를 예측한다. 따라서 사용자들이 많은 항목들에 대해 평가를 해주어야만 추천의 질이 높아진다. 하지만 사용자가 시스템 내의 모든 항목들에 대해 평가를 하지는 못하므로 협업 필터링 시스템의 근본적인 문제점으로 항상 존재하는 것이 평가 희소성(rating sparsity)이다. 이러한 협업 필터링 시스템의 평가 희소성 문제는 적용 도메인이 동적 컬렉션인 경우 더욱 심화되며, 이런 경우 시스템의 예측 품질은 더욱 나빠진다. 그러므로 협업 필터링 기법을 동적 컬렉션에 그대로 적용하면 공통 항목에 대한 평가가 매우 적어 시스템의 예측 품질이 낮아지게 된다. 이러한 문제를 해결하기 위해서 과거에 평가된 항목들과 향후에 추가되는 항목들 간의 연관관계를 찾아서 평가 점수를 공유하여 예측의 질을 높이는 방법이 필요하다.

본 논문에서는 동적으로 추가되는 항목에 대한 예측에 초점을 두어 협업 필터링의 적용 방안을 살펴본다. 제안하는 방법은 사용자의 항목에 대한 취향을 Singular Value Decomposition(SVD, 이하 SVD) 기법으로 분석하여 *취향 공간*을 생성한 후, 취향에 기반한 항목 클러스터를 생성 유지한다. 새로운 항목이 추가되면 과거 항목과의 연관성을 찾기 위해 취향 공간 상의 값으로 변환하고 해당 항목이 어느 클러스터에 속하는지 찾아낸 후 과거 항목들과 평가 정보를 공유한다.

본 논문은 다음과 같이 구성된다. 관련 연구에서는 간략하게 기존의 협업 필터링 시스템들에 대해 알아보고 예측 성능을 높이기 위한 노력들에 대해 알아본다. 3장에서는 협업 필터링 알고리즘에 대해 자세히 살펴보고 협업 필터링이 동적 컬렉션에 적용되는 경우의 문제점을 인식한다. 그런 다음 SVD에 대한 소개 및 SVD 기법의 협업 필터링에 대한 적용 방안을 살펴본 후 우리의 구체적인 적용 방법을 제시한다. 4장에서는 제안한 방법을 평가하기 위한 실험 환경 및 결과를 제시하고 결론 및 향후 작업으로 마무리 한다.

2. 관련 연구

[1]은 처음으로 이웃 기반 기법(neighborhood based method)에 의한 자동화된 협업 필터링 시스템이다. 이웃 기반 기법은 추천하고자하는 사용자와 취향이 비슷한 사용자 그룹을 형성 한 후, 사용자 그룹 내의 평가 점수들을 가중 합산하여 해당 사용자의 평가 점수를 예측한다. 사용자들 간의 유사도 계산을 위해서는 피어슨 상관계수(Pearson correlation coefficient)가 사용되었다.

그 후 협업 필터링에서의 평가 희소성 문제를 해결하기 위해 여러 가지 방법들이 시도되었다. [2]는 여러 종류의 평가 에이전트(agent)를 두어 사용자가 아직 평가되지 않은 항목들에 대해 자동으로 평가를 행하게 하였다. [3]은 계층적 클러스터링(Hierarchical clustering) 기법을 이용하여 사용자와 항목에 대해 각각 클러스터링을 실시한 후 유사한 항목들 간에 평가 점수를 공유함으로써 평가 희소성을 극복하고자 하였다. [4]는 SVD 기법을 사용하여 사용자-항목 평가 행렬의 차원을 줄여 평가를 예측하는 방법을 제안하고, 영화 및 상거래 데이타를 이용하여 기존의 협업 필터링 알고리즘에 의한 예측 결과와 비교하였다.

3. 동적 컬렉션에 대한 협업 필터링의 적용

이번 장에서는 협업 필터링 분야에서 사용되는 여러 기법들에 대해 살펴보고, 우리가 제안하는 방법이 어떤 식으로 적용될 수 있는지 자세히 알아본다. 이를 위해서 협업 필터링 알고리즘과 SVD 기법에 대한 설명부터 시작한다.

3.1 협업 필터링 알고리즘(Collaborative Filtering Algorithm)

협업 필터링 시스템에서의 데이타베이스는 사용자, 항목, 그리고 사용자의 항목에 대한 평가 점수로 구성된다 : <user, item, rating>. 협업 필터링 시스템은 사용자의 항목에 대한 선호도 정보를 바탕으로 사용자들 간의 친밀도를 나타내는 유사도를 측정하고, 특정 사용자의 아직 평가하지 않은 항목에 대한 요구가 있을 때 다른 사용자들의 평가 정보와 그 사용자와의 유사도 정보를 이용하여 평가 점수를 예측한다. 그림 1은 협업 필터링 시스템에서 일반적으로 사용되어지는 이웃 기반 알고리

즘을 설명하고 있다. 이웃 기반 알고리즘은 먼저 활동 사용자(active user)와 다른 사용자들과의 거리를 계산하고 거리가 가장 가까운 k명의 사용자들을 이웃으로 선별한다. 사용자들 간의 거리를 구하기 위해서는 여러 가지 척도가 사용되어질 수 있는데, 이에는 피어슨 상관 계수(Pearson correlation coefficient), 제곱 평균 차이(mean-square-difference), 벡터 유사도(vector similarity) 등이 있다. [5]에서 피어슨 상관계수가 벡터 유사도 보다 좋은 결과를 생성함으로 보였고, [6]에서 피어슨 상관 계수가 제곱 평균 차이보다 좋은 결과를 낸다는 것을 보였다. 뿐만 아니라, [6]은 너무 작거나 너무 큰 이웃의 수를 선택하면 예측 성능이 떨어진다는 것을 보였다. 사용자 간의 거리가 구해지면 항목에 대한 예측 점수는 다음의 식과 같이 다른 사용자들의 평가 점수를 거리 비율에 따라 합해줌으로써 얻어진다.

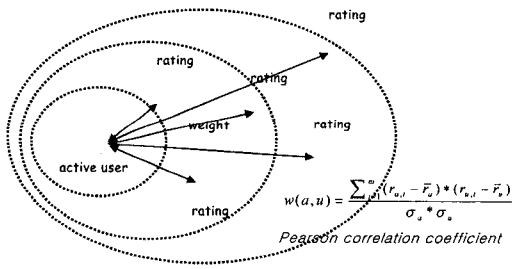

그림 1 협업 필터링 알고리즘

$$pa,i=\bar{r}_a+\frac{\sum_{u=1}^{n}(r_{u,i}-\bar{r}_u)*w_{a,u}}{\sum_{n=1}^{n}w_{a,u}}$$

$p_{a,j}$는 활성 사용자 a의 항목 i에 대한 예측을 나타낸다. n은 이웃 사용자수이며 $r_{u,i}$는 사용자 u의 항목 i에 대한 평가 점수이고 $w_{a,u}$는 활성 사용자와 이웃간의 유사도 가중치이다[1].

협업 필터링 시스템의 성능에 영향을 주는 가장 큰 요인은 공통된 항목에 대한 사용자들의 평가수이다. 공통 항목은 사용자간의 유사도 측정에 사용될 뿐만 아니라 평가 점수 예측에도 직접적으로 사용되므로, 공통 항목이 적은 경우 예측의 질이 떨어질 수밖에 없다. 이 문제를 평가 희소성(rating sparsity)이라고 하며, 이를 해결하기 위해서 앞서 관련 연구에서 살펴보았듯이 여러 방법들이 제안되었다.

3.2 동적 컬렉션(Dynamic Collection)

동적 컬렉션은 다음과 같은 특징을 가진다.

- 시간이 중요한 요인으로 작용한다. TV 프로그램의 예를 들어보면, 매일 각 채널 별로 수십 편의 프로그램들이 추가되며 추가된 프로그램은 계속 축적되어 남는 것이 아니라 지난 프로그램은 대부분 사라진다.
- 항목들에 대한 상대적인 가치가 정적인 항목들에 비해 낮다. 예를 들어 대상 항목이 영화나 도서인 경우 지속적으로 가치가 유지되나 TV 프로그램과 같은 경우 지나간 프로그램에 대한 평가는 상대적으로 가치가 낮다.
- 과거와 새로운 항목들 간에 연관성이 존재한다. 계속 새로운 항목들이 추가되지만 완전히 새로운 내용의 항목들이 추가되는 것은 아니다. TV 프로그램의 시리즈 같은 경우 한번 시청한 사람은 계속 그 프로그램을 시청하려는 경향이 강하고 뉴스 프로그램을 선호하는 사람은 지속적으로 고정된 뉴스 프로그램에 대해 선호도가 높다.

이와 같은 동적 컬렉션의 특징 때문에 협업 필터링 기법을 그대로 적용하기에는 무리가 따른다. 협업 필터링 기법의 근본적인 문제인 평가 희소성 문제가 동적 컬렉션인 경우 보다 심화되기 때문이다. 계속 항목이 변화되므로 사용자들 간의 공통적인 항목도 존재하기 어려워지며, 특정 항목에 대한 평가 점수 역시 해당 항목이 사라지게 되면 존재 의미가 없어진다. 또한 상대적으로 존재 가치가 떨어지는 동적 항목들 및 해당 평가 점수들을 모두 저장하여 유지하는 것 역시 저장 공간의 측면에서 부담이 크다.

3.3 협업 필터링과 Singular Value Decomposition (SVD)

우리의 목적은 새로운 항목들과 기존의 항목들 간의 연관성을 찾아 기존의 평가 정보를 이용하여 새로운 항목에 대한 평가를 예측하는 것이다. 협업 필터링에서 사용자는 항목을 평가할 때 개인의 취향에 의해 평가 점수를 부여하므로 항목들을 취향의 측면에서 분석하는 것이 필요하다. 사람의 취향은 매우 복잡하며 여러 요인에 의해 영향을 받으므로 고차원의 데이타로써 표현된다. 따라서 이러한 고차원의 데이타를 추출하고 분석하는 기법이 요구된다.

협업 필터링 시스템의 데이타베이스를 행렬의 관점에서 보면, 그림 2와 같이 사용자를 행으로 가지고 항목을 열로 가지며, 원소를 평가 점수로 갖는 행렬을 생각해 볼 수 있다. 주어진 평가 점수들을 이용해서 비어있는 평가 점수를 구하는 것이 협업 필터링이 해결해야 할 문제다. 이를 위해서는 위와 같이 구성된 행렬에서 사용자와 항목간의 연관성을 분석하여 그 정보를 바탕으로 아직 평가되지 않은 항목에 대한 점수를 예측하는 작업이 필요하다. 한편 이와 유사한 문제가 정보 검색(Information Retrieval, IR) 분야에서도 제기되어 많은 학자들이 이에 관한 연구를 해왔다. 정보검색 분야에서

$$M = U\Sigma V^T \qquad (U^TU = V^TV = I_n,\ \Sigma = diag(\sigma_1,\cdots,\sigma_n), \sigma_i > 0 for 1 \le i \le r, \sigma_j = 0 for j \ge r+1)$$

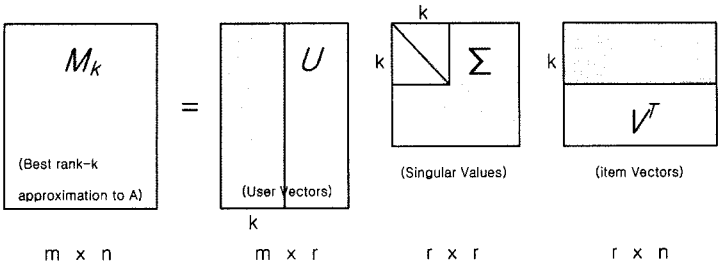

그림 2 Singular Value Decomposition (SVD)

는 문서와 문서 내에 출현하는 용어(term) 들 간의 동시 출현하는 패턴(co-occurrence)을 분석해서 서로간의 연관성을 밝혀 검색의 정확도를 높이려는 시도가 있었고 이를 위해 제시된 방법들 중 대표적인 것으로 Latent Semantic Indexing(LSI)이 있다. LSI는 용어(term)와 문서 간의 연관성을 분석하고 잠재적인 의미 정보를 얻기 위해 고차원의 초기 용어-문서 행렬을 SVD 기법을 이용하여 저차원의 행렬을 생성한다[7]. 이와 유사하게 협업 필터링의 데이타를 문서-용어 데이타 대신 사용자, 항목 간의 행렬로 보고 이들의 상관 관계를 구하는데 SVD 기법을 사용할 수 있다. 이번 장의 나머지 부분은 SVD 기법에 대한 소개 및 우리가 제안하는 예측 방법에 대해 자세히 알아본다.

3.4 Singular Value Decomposition (SVD)

SVD는 m개의 행과 n개의 열을 가지는 초기 행렬 M을 다음과 같이 세 개의 행렬로 분해하는 잘 알려진 행렬 인수분해(matrix factorization) 기법이다.

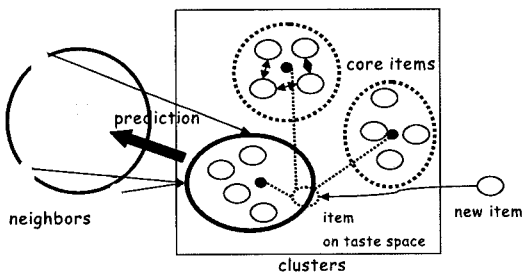

그림 3 새로운 항목에 대한 예측 과정

$$M = U\Sigma V^T$$

여기에서 $U^TU = V^TV = I$ 이고, $1 \le i \le r$, $\sigma_j = 0$, $j \ge r+1$에 대해서 $\Sigma = diag(\sigma_1, \cdots, \sigma_n), \sigma_i > 0$이다. U와 V는 각각 원본 행렬의 좌측 및 우측 고유 벡터(singular vector)이며, Σ는 해당 원소로 고유값(singular value)을 가지는 대각 행렬(diagonal matrix)이다. U는 원본 행렬의 열에 해당하는 고유 벡터들로 이루어진 $m \times c$행렬이고, V는 원본 행렬의 행에 해당하는 고유 벡터들로 이루어진 $n \times r$행렬이다. Σ는 해당 원소를 고유값으로 갖는 $r \times r$대각 행렬이다. 고유값들은 내림차순으로 구성되어지는데 이는 특징들의 중요성을 나타낸다. 행렬 M의 계수(rank)는 영이 아닌 고유값 r의 개수가 되며, 행렬의 차원을 줄이기 위해서는 고유값들 중 값이 큰 k개의 값들만 취하면 된다. Σ의 차원을 줄임으로써 U와 V의 차원도 같이 낮아지는데, 결과적으로 생성되는 행렬은 원본 행렬 M의 계수 k 근사 행렬(rank-k approximation matrix) $M_k = U_k\Sigma_k V_k^T$이 된다. 차원이 감소된 행렬은 본래의 행렬에서 중요한 특징들만 추출하여 구성되어지므로 적절한 k 값만 잘 선택하면, 고차원의 본래 행렬보다 계산이 용이하고 중요하지 않은 성분들을 제거해주는 효과가 있다.

3.5 평가 예측 생성

평가 예측을 위해서는 먼저 사용자-항목 행렬에서 SVD를 사용하여 항목들의 특징을 추출하고 항목들 간의 관계를 얻은 후 이 정보를 기반으로 클러스터링을 수행한다. 협업 필터링에서 사용자와 항목은 취향이라는 매개체를 통해 연관된다. SVD를 이용하면 항목들을 사용자 *취향 공간*(사용자 벡터들이 이루는 공간)에 대한 벡터들로 표현할 수 있는데, 이는 다음과 같다.

$$\hat{i}_k = i^T U_k \Sigma_k^{-1}$$

i는 항목 벡터로써 사용자 평가 점수로 표현되는 특성 벡터이고, U_k는 각각 k개의 원소를 가지는 고유 벡터들로 이루어지고, Σ_k는 고유값들 중 값이 큰 k개의 값들을 원소로 가지는 대각 행렬이다. $\hat{i}_k$는 현재 사용자 벡터(U_k의 열)들이 이루는 공간에 대한 i의 사영(projection)이 된다. 이렇게 해서 얻어진 취향 공간에 대한 항목 벡터들을 서로 유사한 것들끼리 클러스터링 하는데, 클러스터링을 위한 거리 척도로는 두 벡터 간의 cosine 값을 사용한다. 클러스터링을 위한 알고리즘은

잘 알려진 k-means 클러스터링 알고리즘을 사용한다[8]. 각 클러스터는 항목에 대한 사용자 취향을 나타내며 각 클러스터 별로 클러스터의 중심점(centroid)과 가장 근접한 상위 n개의 항목들을 선별하며 이를 핵심 항목(core item)이라고 한다.

알고리즘 1은 새로운 항목 n에 대한 사용자 u의 평가 점수 예측 과정을 나타낸다. 새로운 항목에 대한 예측을 구하기 위해서는 먼저 사용자 평가 점수로 구성되는 특성 벡터에 해당 공간에 대한 사영 연산을 수행하고, 이를 통해 얻어진 항목 벡터를 취향을 나타내는 클러스터와 비교한다(줄 4). 줄 5-8은 벡터 $\hat{n}$과 가장 근접한 클러스터를 찾은 후 평가 점수를 예측하는 과정으로 새로운 항목에 대한 평가 점수를 해당 클러스터 내의 핵심 항목들에 대한 평가 점수를 기반으로 예측하게 된다. 줄 9-11에서는 새로운 항목이 핵심 항목으로 유지될지 여부를 판단한 후 교체하는 과정이다.

Algorithm 1 새로운 항목에 대한 평가 예측

1: **INPUT:** C_k : clusters of projected items, R : rating database, NN :Top-n neighbors, u : user, n : new item vector
2: **OUTPUT:** pr : predicted rating
3: **Procedure:** predict-rating($C_k, R, NN\ u, n$)
4: $\hat{n} \leftarrow$ projection of n onto the taste space : $n^T U_k \Sigma_k^{-1}$
5: **for** each cluster C_i in C_k **do**
6: compare $\hat{n}$ with centroid of C_i and find nearest cluster C_n
7: **end do**
8: **for** each core item c_i **in** C_i **do**
9: **for** each neighbor u_i **in** $NN(u_i, U_n)$ **do**
10: $pr \leftarrow$ calculate weighted average of $R(u_i, c_i)$ in C_i
11: **end do**
12: **end do**
13: **if** distance($\hat{n}$, centroid) < distance(Top-n core item, centroid) **then**
14: reconstruct core items with $\hat{n}$
15: **end if**

4. 실험

4.1 실험 데이타 집합 및 환경

본 실험을 위해서 Digital Equipment Research Center에서 1995년부터 1997년에 걸쳐 *EachMovie* 서비스를 통해 수집한 *EachMovie* 데이타를 사용했다[10]. 이는 약 1,600편의 영화에 대해서 대략 60,000명 정도의 사용자가 0부터 5까지의 점수(정수)로 평가한 약 280만 개의 데이타로 추천 시스템의 성능을 평가하는데 널리 사용되는 데이타다. 본 논문의 실험에서는 평가 개수가 적은 영화를 제외하여 20명 이상의 사용자들로부터 평가를 받은 1623편의 영화만을 선별하였다. 그 후 시간에 따라 추가되는 동적 컬렉션을 시뮬레이션하기 위해 데이타를 두 부분으로 나누었는데, 후반기에 새로 추가되어 이전에 평가를 받지 않은 영화들을 후반기 영화, 나머지 전반기부터 계속 지속되는 영화들을 전반기 영화로 각각 구성하였다. 그 결과 전반기의 영화는 대략 80% 정도인 1323편이며 후반기의 영화는 약 20%인 300편이 되었다.

이렇게 얻어진 전반기 영화들에 대한 평가 데이타를 사용하여 1565개의 행(1565 사용자), 1323개의 열(1323 영화)을 가지는 사용자-영화 행렬 M을 구성하였다. 행렬 M의 각 원소 $r_{i,j}$는 i번째 사용자의 j번째 영화에 대한 평가 점수를 의미한다. 그러고 나서 [4]에서 행한 것과 같이 해당 행렬에 대한 정규화를 실시했다. 정규화 작업은 평가되지 않아 비어있는 값에 항목 평균(각 영화에 대한 평가 점수의 평균)을 채우고, 각 값에서 사용자 평균을 빼는 방식을 택했다. 그 후 정규화된 행렬 A에 MATLAB을 이용해 SVD 연산을 수행하여 분해된 세가지 행렬 U, Σ, V을 얻었다. Σ_k는 Σ행렬에서 값이 큰 k 개의 고유값들만 유지해서 얻어진 행렬이며, 최적의 k 값은 실험적인 결과에 의해 구한다. 사용자 벡터공간에 대한 항목 벡터의 사영을 구하기 위해 $U_k\Sigma_k^{-1}$ 행렬을 구한 후 A^T을 곱한다. 결과 행렬은 사용자 취향 공간에 대해 사영되어진 항목들의 벡터로 이루어지며 클러스터링 작업을 위한 거리 척도로 사용된다. 클러스터링은 k-means 알고리즘에 의해 수행되며 평가 점수 예측은 3장에서 설명되어진 알고리즘에 의해 구한다.

4.2 평가 척도 및 평가 시스템

개별 항목에 대한 예측 성능을 평가하기 위한 척도로는 Mean Absolute Error (MAE)를 사용했다. MAE는 예측 엔진에서 예측되어 나온 점수와 실제 사용자가 평가한 점수를 비교하여 시스템의 예측 정확도를 평가하는 통계적인 척도이다. 해당 척도는 가장 널리 사용되어지며 직관적이다[4].

본 논문의 예측 기법의 성능을 평가하기 위한 비교 시스템으로는 첫 번째로 [1]에서 설명된 일반적인 CF 엔진과 두 번째로 [3]에서 제안된 항목에 대한 계층적 클러스터링 시스템(H.Clustering)을 구현 사용하였다. CF 엔진은 피어슨 상관계수를 이용한 이웃 기반 알고리즘을 사용하고 있으며 점수 예측을 위해 이웃 편차의 가중 평균에 의해 구해진다.

4.3 실험 과정

본 실험은 동적 컬렉션에 대한 환경을 시뮬레이션하기 위해 새로운 항목들이 지속적으로 추가되는 상황을 가정한다. 부언하면 새로 등장하는 항목에 대해 시스템

의 예측 점수와 실제 사용자의 평가 점수를 비교하는 것이다. 실험 데이타의 후반기 영화는 앞서 설명된 바와 같이 이전에 평가된 적이 없는 영화들로 구성된다. 이러한 후반기 영화들의 평가 점수들을 일단 제거한 후 순차적으로 초기 평가 수를 늘려나가면서 예측의 정확도가 어떻게 변화하는지 검사한다. 즉, 후반기 영화들에 대한 평가 점수 집합을 여러 부분 집합으로 나누었는데, 예를 들어, 평가 개수가 없는 집합, 항목 당 평가수가 10개인 집합, 모든 평가 점수를 가지는 집합 등으로 나누어서 실험에 사용하였다. 새로운 항목들에 대한 예측 정확도를 측정하기 위해 후반기 영화들에 대한 평가 점수들을 무작위로 요청하여 실제 평가 점수와 비교를 하였다.

4.4 실험 결과

최종적인 타 시스템과의 비교 실험 이전에 먼저 본 기법에 필요한 환경 인자들을 정하는 작업이 필요하다. 이러한 환경 인자들에는 차원 감소를 위해 사용되어지는 k 값과 클러스터의 개수, 각 클러스터 내의 핵심 항목 개수 등이 있다.

첫 번째, SVD 연산에서 차원 감소를 위해 사용되는 k 값을 구하기 위해 값을 2부터 하나씩 증가시켜가면서 MAE의 변화를 조사했으며, 그림 4가 이를 나타낸다. 그림을 보면 k 값이 증가에 따라 점차 성능이 향상되다가 15에서 가장 좋은 MAE를 보임을 알 수 있다.

두 번째 클러스터의 개수에 따른 MAE 변화를 살펴보았는데, 그 결과를 그림 5가 보여준다. 그림을 보면 클러스터의 개수가 2부터 증가하여 15에서 가장 좋은 성능을 보이며 그 이상의 클러스터 개수에서는 점차 나쁜 성능을 보인다. 첫 번째 k 값과 클러스터의 개수는 서로 상관 관계를 가지는 변수이므로 여러 차례 실험을 통해 각 값이 최적일 때에 다른 값의 변화를 측정하는 방식으로 결정된다. 즉, 그림 4는 클러스터 개수가 15인 경우이고, 그림 5는 k가 15인 경우이다. 다른 시도로 k 값에 따른 평가 척도로 MAE 값이 아닌 클러스터링의 품질[1]을 계산하여 k와 클러스터 간의 상관관계를 측정하려 했으나 좋은 결과를 얻지 못했다.

다음 실험은 각 클러스터 내의 핵심 항목들에 개수에 따른 MAE의 변화로 그림 6이 이를 나타난다. 그림 6에서 보면 핵심 항목의 증가에 따라 점차 나은 성능을 보이다가 50개에서 가장 좋은 성능을 보였으며, 그 이후로는 점차 나빠졌다. 이는 협업 필터링의 이웃 기반 알고리즘에서 이웃의 개수에 따른 MAE 변화와 유사하다.

마지막으로는 이상의 실험에서 얻어진 환경 변수들을 바탕으로 평가 예측 성능을 알아보았다. 이는 초기 평가

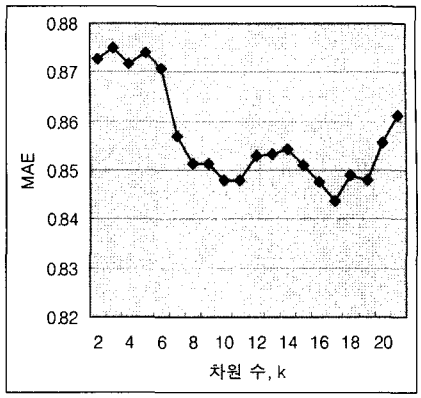

그림 4 차원에 따른 MAE 변화

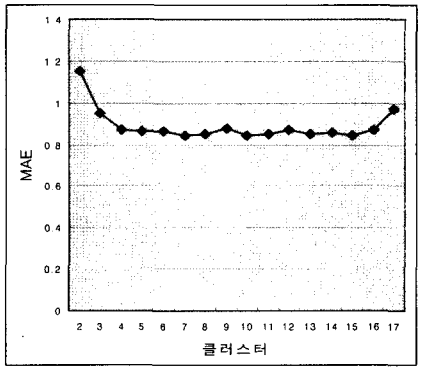

그림 5 클러스터 개수에 따른 MAE 변화

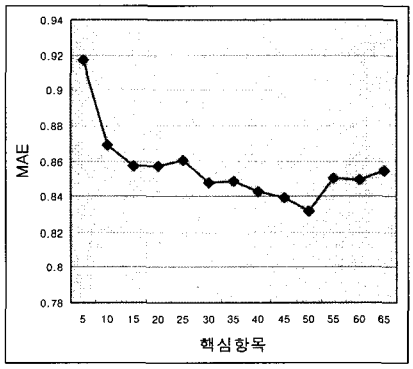

그림 6 핵심항목 변화에 따른 MAE

수를 증가시켜가면서 시스템이 얼마나 빨리 안정화되어 항목에 대한 평가 예측 작업을 수행할 수 있는지 평가하는 것이다. 일반적으로 한 항목이 사용자의 평가수를 얻기 위해서는 많은 시간이 소요되고 평가수의 편차가 항목 별로 크므로 적은 수의 평가수로 시스템을 안정화된 상태로 끌어올리는 것은 추천 시스템의 중요 요인이다. 그림 7은 초기 평가수를 증가시키면서 CF 엔진, H. Clustering, 본 논문의 제안한 기법(SVD Clustering)

1) 각 클러스터에 존재하는 점들이 얼마나 서로 가까운지 거리 계산

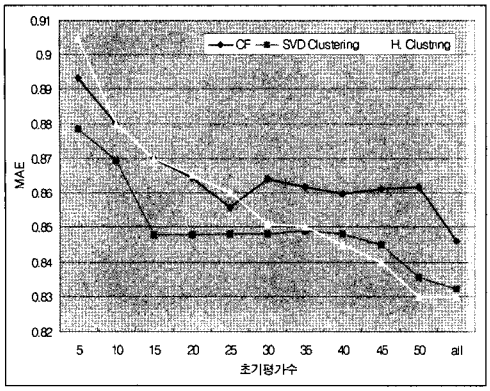

그림 7 평가 예측 성능 비교

간의 MAE 변화를 그린 것이다. 평가수가 극히 적은 경우에는 모두 나쁜 성능을 보였고 평가수가 점차 늘어남에 따라 본 논문의 SVD Clustering 방식이 다른 기법보다 안정 상태로 빨리 접근하는 것을 볼 수 있다.

5. 결론 및 향후 연구

협업 필터링은 추천 시스템에서 널리 사용되는 기법이며, 전자 상거래 시장이 급격하게 발전하면서 추천 시스템은 향상된 고객 서비스를 제공하기 위해 중요한 요소로 자리 잡고 있다. 또한 고객들의 요구를 만족시키기 위해 인터넷 서비스가 점차 다양해지면서 그 응용 분야를 넓혀가고 있다.

본 연구에서는 협업 필터링 알고리즘이 동적으로 추가되는 대상 분야에 적용될 때의 문제점을 인식하고 이에 초점을 두어 협업 필터링의 적용 방안을 제시하였다. 제안한 방법에서는 SVD 기법을 이용해서 항목들의 공통 특징들을 추출하여 취향 공간을 생성하였다. 새로운 항목에 대한 평가 예측이 요구될 때, 예측 엔진은 취향 공간에서 새로운 항목과 과거의 항목들 간의 관련성을 조사하여 과거 항목들에 대한 평가 정보를 공유한다. 제안한 방법을 평가하기 위해 항목들이 지속적으로 추가되는 상황을 시뮬레이션하여 예측 결과의 질적 변화를 분석하였고, 제안한 방법이 동적 컬렉션에 대해서도 빨리 적응하여 예측 작업을 수행할 수 있음을 보였다.

향후 연구로는 사용자와 항목 간의 관계에 대한 보다 면밀한 분석 작업을 통한 향상된 취향 공간 생성을 생각해 볼 수 있다. 이를 통해 일반적인 사용자뿐만 아니라 협업 필터링의 단점인 소수의 사용자에 대한 평가 예측 성능도 향상시킬 수 있으리라 기대된다. 이를 위해 정보 검색 분야에서 연구된 용어, 문서와의 관계 분석을 위한 기법들을 추가로 적용해 볼 수 있다. 또한 보다 정확한 과거와 새로운 항목들 간의 연관성을 구하기 위해 내용기반 추천 기법을 결합하는 방법도 생각해 볼 수 있다.

참 고 문 헌

[1] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work.

[2] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999.

[3] Arnd Kohrs and Bernard Merialdo. Clustering for collaborative filtering applications. In Computational Intelligence for Modelling, Control & Automation. IOS Press, 1999.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems-a case study. In ACM WebKDD Workshop, 2000.

[5] John S. Breese, David Heckerman and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, 43-52, July 1998.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 230-237, 1999.

[7] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. Journal of the Society for Information Science, 41(6):391-407, 1990.

[8] R. Duda and P. Hart. Pattern classification and Scene Analysis. John Wiley and Sons, Inc. 1973.

[9] Berry, M. W., Dumais, S. T., and O'Brien, G. W. Using linear algebra for intelligent information retrieval. SIAM Review 37(4):573-595. 1995.

[10] Eachmovie collaborative filtering data set, 1997. www.research.digital.com/SRC/eachmovie.

이 형 동

1997년 홍익대 컴퓨터공학과(학사). 1999년 서울대 컴퓨터공학과(석사). 2006년 서울대 컴퓨터공학과(박사). 관심분야는 데이타베이스, 정보검색

김 형 주

한국정보과학회논문지 : 데이타베이스 제 34 권 제 2 호 참조