분류 속성과 Naive Bayesian을 이용한 사용자와 아이템 기반의 협력적 필터링

User and Item based Collaborative Filtering Using Classification Property Naive Bayesian

김종훈*, 김용집**, 정경용***, 임기욱****, 이정현*****
인하대학교 컴퓨터정보공학과*, ACCESS SEOUL Co, LTD**,
상지대학교 컴퓨터정보공학부***, 선문대학교 컴퓨터정보학부****, 인하대학교 컴퓨터정보공학부*****

Jong-Hun Kim(jhkim@hci.inha.ac.kr)*, Yong-Jip Kim(jobs76@naver.com)**,
Kyung-Yong Chung(kyjung@sangji.ac.kr)***, Kee-Wook Rim(ksw0503@hotmail.com)****,
Jung-Hyun Lee(jhlee@inha.ac.kr)*****

요약

협력적 필터링은 피어슨 상관 계수에 의해 유사도를 구하고, 선호도를 기반으로 이웃 선정 방법을 사용하므로 아이템에 대한 내용을 반영하지 못할 뿐만 아니라 희박성 및 확장성의 문제를 가지고 있다. 이러한 문제점을 개선하기 위하여 아이템 기반 협력적 필터링이 실용화되었으나 아이템의 속성을 반영하지는 못한다. 본 논문에서는 기존 추천 시스템의 문제점을 보완하기 위하여 분류 속성과 Naive Bayesian을 이용한 사용자와 아이템 기반의 협력적 필터링을 제안하였다. 제안한 방법에서는 희박성 문제를 해결하기 위하여 명시적 데이터에 기반한 아이템 유사도와 묵시적 데이터에 기반한 사용자 유사도를 복합적으로 참조한다. 참조 결과에 대해 Naive Bayesian을 적용한다. 또한 속성을 반영하기 위해 아이템 분류속성간의 유사관계 순위를 아이템 유사도 계산에 반영함으로써 정확성을 높일 수 있었다.

■ 중심어 : | 협력적 필터링 | 네이브 베이지안 | 데이터마이닝 |

Abstract

The collaborative filtering has used the nearest neighborhood method based on the preference and the similarity using the Pearson correlation coefficient. Therefore, it does not reflect content of the items and has the problems of the sparsity and scalability as well. the item-based collaborative filtering has been practically used to improve these defects, but it still does not reflect attributes of the item. In this paper, we propose the user and item based collaborative filtering using the classification property and Naive Bayesian to supplement the defects in the existing recommendation system. The proposed method complexity refers to the item similarity based on explicit data and the user similarity based on implicit data for handing the sparse problem. It applies to the Naive Bayesian to the result of reference. Also, it can enhance the accuracy as computation of the item similarity reflects on the correlative rank among the classification property to reflect attributes.

■ keyword : | Collaborative Filtering | Naive Bayesian | Data Mining |

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (IITA-2007-C1090-0701-0020)

접수번호 : #070614-001
접수일자 : 2007년 06월 14일
심사완료일 : 2007년 08월 13일
교신저자 : 정경용, e-mail : kyjung@sangji.ac.kr

1. 서 론

사용자 기반 협력적 필터링에서 고려하기 힘든 부분에 대해서 내용 기반 필터링을 이용함으로써 문제점을 해결한다. 그러나 사용자 기반 협력적 필터링은 대용량 데이터베이스에서는 아이템간의 지지도와 신뢰도에 근거하여 규칙을 발견하는 특징을 갖고 있기 때문에 개인별 사용자의 성향을 반영하지 못하는 단점을 가지고 있다. 보다 좋은 성능을 얻기 위해서는 이러한 필터링 기법을 보완하는 연구가 필요하며 내용 기반 필터링과 사용자 기반 협력적 필터링을 결합하여 더 좋은 예측 결과를 얻고자 하는 연구가 최근에 이루어지고 있다[10]. 최근 아마존 웹사이트에서 도입하여 성능과 효율성 측면에서 우수성이 입증된 아이템 기반 협력적 필터링을 적용한 방법이 있다[1]. 이는 기존의 방법이 가지는 희박성과 확장성 측면의 단점을 개선하여 상업적으로 성공한 기술이라고 평가받고 있다. 그러나 여전히 희박성의 문제가 존재하며 아이템의 속성이 반영되지 않는다는 문제점이 있다. 본 논문에서는 기존의 아이템 기반 협력적 필터링에서의 희박성 문제를 개선하기 위하여 아이템의 유사도와 사용자의 유사도를 모두 참조하며 그 결과를 Naive Bayesian에 적용한다. 그 결과는 선호도에 대한 예측으로 나타나며, 이를 사용하여 추천이 이루어진다. 또한 아이템의 속성을 반영하기 위하여 아이템이 가지는 분류 속성간의 유사도 순위에 의한 가중치가 아이템 유사도 계산 과정에 반영됨으로써 정확도를 높일 수 있다.

2. 협력적 필터링과 Naive Bayesian

2.2 사용자 기반 협력적 필터링

협력적 필터링은 사용자의 암묵적인 데이터를 사용하는지 명시적 데이터를 사용하는지에 따라 구분된다. 명시적인 데이터는 사용자에게 특정 아이템에 대한 선호를 0에서 5까지 정도의 이산형 척도로 입력받는 경우를 말한다. 암묵적 데이터를 사용하는 경우는 사용자의 선호를 대변하는 웹사이트 클릭 패턴이나 구매 패턴 등을 웹 로그나 구매 이력 데이터에서 발견하여 예측하는 것이다. 따라서 협력적 필터링은 암묵적 협력적 필터링과 명시적 협력적 필터링으로 구분한다. 협력적 필터링에서 사용자와 유사한 선호도를 가지는 이웃을 찾아내고 사용자간에 평가한 아이템을 예측하기 위해서 사용되는 유사도 가중치로는 피어슨 상관계수, 스피어만 순위 상관계수, 벡터 유사도, 기본 선호도, 역 사용자 빈도 등이 사용된다[2].

유사도 가중치를 계산한 후에는 선호도를 예측하기 위해 몇 명의 이웃을 사용할 것인지를 결정해야 한다. 유사도 가중치가 구해진 모든 이웃을 사용해서 선호도를 예측할 수 있지만 이는 정확도나 성능 면에서 그리 좋은 방법이 아니다. 반면 유사도가 높은 이웃만을 예측해서 고려할 경우 유사도가 높지 않은 아이템에 대해서 예측할 수 없는 경우가 발생할 수 있다. 그렇지만 정확도나 성능 면에서 유사도가 높은 이웃의 선호도를 사용해서 예측하는 것이 더 나은 결과를 얻을 수 있을 것으로 생각되며 이는 데이터의 분석을 통해서 예측의 정확도가 얼마나 되는지 얼마나 많은 사람과 아이템을 예측할 수 있는지 따라 적절한 이웃의 수를 결정해야 한다. 특정 사용자와 유사한 n명의 이웃을 사용해서 예측하기 위해 임계값과 가장 좋은 이웃 방법을 사용하여 예측에 사용될 이웃의 수를 결정한다. 이들 방법을 조합하여 유사도 가중치가 임계값 이상인 이웃을 대상으로 n명의 가장 좋은 이웃 방법을 적용한다[3]. 특정 사용자와 가까운 이웃이 선택되면 예측을 위해서 이웃의 아이템에 대한 선호도를 같은 분포를 따르는 척도로 변환하여 조합한다. 가능한 모든 이웃의 선호도를 사용하는 것이 기본적인 방법이지만 이웃의 유사도를 가중치로 보고 선호도를 유사도 가중평균값으로 계산하는 방법이 많이 사용된다. 따라서 모든 사용자의 선호도가 근사적으로 같은 분포를 따른다는 가정을 기본적으로 하게 된다. 사용자 선호도 예측의 방법으로는 평균 편차 방법과 z값 가중치 평균 방법이 있다[8].

2.2 아이템 기반 협력적 필터링

아이템 기반 협력적 필터링은 사용자 기반의 방법과 다르지 않지만, 유사도 가중치의 계산과 선호도를 예측

하는 방법이 다를 뿐이다[4]. 이는 추천을 요구한 사용자가 평가한 아이템의 집합을 찾아내고 추천 후보 아이템 i와 얼마나 비슷한지 계산하게 된다. 이 과정에서 k개의 가장 유사한 아이템 집합 $\{i_1, i_2, ..., i_k\}$을 선택하며, 동시에 대응하는 유사도 $\{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$를 계산한다. 유사한 아이템이 발견되면, 가중치의 선호도 합으로써 예측한다[4].

아이템 유사도는 서로 다른 아이템에 대해 평가한 사용자를 분리해내고, 아이템 i와 아이템 j에 대한 유사도 $s_{i,j}$를 계산하는 것이다. 코사인 기반의 유사도는 아이템을 m차원의 공간에 있는 벡터로 구성한다. 구성된 아이템간의 유사도는 벡터 사이의 코사인을 계산하여 측정된다. 상관계수 기반의 유사도는 아이템간의 유사도를 피어슨 상관계수로 계산하는 방법이다. 사용자 u가 아이템 i와 아이템 j를 공통으로 평가한 아이템을 기반으로 유사도를 계산한다. 이는 아이템 기반의 협력적 필터링의 대표적인 방법이다. 개선된 코사인 유사도는 서로 다른 사용자 사이에서 나타나는 평가치의 차이를 취급하지 않는다는 단점을 보완한 방법이며 기존 방법 대비 정확성 측면에서 우수하다는 장점이 있다.

아이템간의 유사도가 계산되어지면 예측을 수행한다. 기존 연구에서는 협력적 필터링에서 예측결과를 생성시키는 것이다. 먼저 유사도를 바탕으로 유사 아이템 집합을 구성하면, 다음 단계는 목표한 사용자의 평가 데이터를 기반으로 추천하는 것이다. 가중치 합 방법은 유사한 아이템에 대해서 평가한 선호도의 가중치 합계를 기반으로 예측하는 방법이다. 이는 아이템 i와 유사한 아이템에 대한 사용자 u의 선호도 합을 계산함으로서 아이템 i에 대한 사용자 u의 선호도를 예측한다.

2.3 Naive Bayesian

Bayesian은 통계적인 분류 방법이며, 주어진 데이터가 특정 클래스에 속할 확률을 예측할 수 있다. 이 방법은 Bayes 이론에 기초하고 있다. Naive Bayesian으로 알려져 있는 Bayesian은 의사 결정 트리나 신경망 알고리즘과 비교해 볼 때 대용량 데이터베이스에서 높은 정확성과 속도를 나타내고 주어진 클래스의 속성 값이 다른 속성에 영향을 주지 않는다.

Bayes 이론은 X를 클래스를 알 수 없는 샘플 데이터라고 하고, H는 X가 C라는 클래스에 포함된다는 가정이라고 정의한다. X라는 샘플 데이터가 발생했을 때 H라는 가정이 발생할 확률은 $P(H|X)$로 결정할 수 있으며, 사후확률이라고 부른다. $P(H|X)$를 계산하기 위해서는 $P(X)$, $P(H)$, $P(H|X)$가 필요하다. 이들은 학습 데이터들을 통해서 측정이 가능하며, 사전확률이라고 한다. $P(H|X)$는 (식 1)로 구할 수 있다.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad \text{(식 1)}$$

Naive Bayesian은 $\{x_1, x_2, ..., x_n\}$라는 n차원의 특징 벡터 X가 존재하며, $\{C_1, C_2, ..., C_m\}$라는 m개의 클래스가 존재한다고 가정한다. 임의의 데이터 X가 가장 높은 사후확률을 가지는 클래스에 속할 것이라는 예측은 Bayes 이론을 사용하여 (식 2)와 같이 계산할 수 있다.

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)},\ 1 \le i \le m \quad \text{(식 2)}$$

$P(X)$는 모든 클래스에 대해 일정한 값을 가지므로, 오직 $P(X|C_i)P(C_i)$만을 최대화하도록 고려한다. 만약 클래스의 사전확률을 알 수 없다면 $P(X|C_i)$만을 고려할 수 있다. $P(X|C_i)$는 Naive Bayesian의 독립가정에 의해 (식 3)과 같이 계산하며, 그 결과 X는 가장 큰 사후 확률을 가지는 클래스로 분류할 수 있다.

$$\begin{aligned} P(X|C_i) &= P(x_1, x_2, \cdots, x_n|C_i)P(C_i) \\ &= P(x_1|C_i)P(x_2|C_i)\cdots P(x_n|C_i)P(C_i) \\ &= P(C_i)\prod_{k=1}^{n} P(x_k|C_i) \end{aligned} \quad \text{(식 3)}$$

3. 분류속성 가중치와 Naive Bayesian을 이용한 사용자와 아이템 기반의 협력적 필터링

기존의 추천 시스템에서 [12]의 방법은 협력적 필터링의 초기 평가 문제를 해결하였으나 희박성 문제를 해결하지 못하였다. LSI를 이용한 방법과 SVD를 이용한

방법은 데이터 입력 차원수를 줄임으로써 희박성 문제를 해결할 수 있었으나 초기 평가 문제는 해결하지 못하였다. [8]의 방법에서는 희박성 문제와 초기 평가 문제를 동시에 해결하려는 시도를 하였다. 본 논문에서는 기존의 아이템 기반 협력적 필터링을 이용한 추천 시스템의 문제점을 보완하기 위해서 아이템 유사도 측정에 아이템 분류 속성의 유사 관계도를 고려하였다. 사용자와 아이템의 유사도에 근거하여 협력적 필터링을 구현하였으며, 최종적으로 Naive Bayesian을 적용한 추천 시스템을 제안한다.

[그림 1]은 제안하는 사용자와 아이템 기반의 협력적 필터링의 구성도이다. 첫 번째 단계로 훈련 데이터 집합으로부터 사용자가 각 아이템에 대하여 평가한 자료를 바탕으로 각 아이템간의 유사도를 계산한다. 아이템간의 유사도에는 각 아이템의 분류속성 관계에 따른 가중치를 곱하며 이 계산을 근거로 아이템 유사도 검색 테이블(Item Similarity Lookup Table)을 생성한다. 동시에 동일한 훈련 데이터 집합으로부터 각 사용자가 아이템에 접근한 기록을 이용하여 Interest-Behavior Matrix(I-B Matrix)를 생성한다. I-B Matrix는 사용자간의 유사도를 측정하는 척도가 되며, 이를 기반으로 사용자 유사도 검색 테이블(User Similarity Lookup Table)을 생성한다.

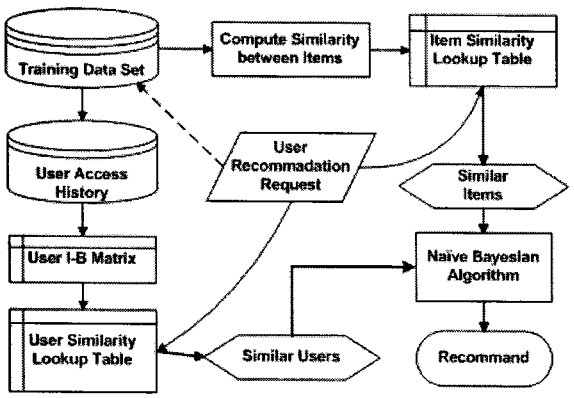


그림 1. 제안한 방법의 시스템 구성도

사용자의 추천 요구가 발생하였을 때, 새로운 선호도가 있거나 아이템 이용 내역이 추가되었다면 이것은 훈련 데이터 집합에 추가되어 사용자와 아이템 유사도 검색 테이블을 갱신하도록 한다. 두 번째 단계에서는 추천 요구에 대한 대응으로 아이템과 사용자 유사도 검색 테이블에서 가능성 있는 추천 후보 아이템과 유사 사용자 집단을 추출한다. 마지막 단계에서 추천 후보 아이템과 유사한 사용자에 대해 Naive Bayesian을 적용하여 추천 후보 아이템에 대한 예측을 한다.

3.1 아이템 기반 유사도 측정

협력적 필터링에서 아이템의 유사도 $sim(i,j)$는 높은 정확성을 요구하기 위해 개선된 코사인 유사도로 계산된다. 두 번째 단계로 아이템간의 유사도에 속성을 반영하기 위하여 아이템이 속하고 있는 분류개념들 사이의 유사관계 순위를 가중치로 나타내어 첫 번째 단계에서 계산된 $sim(i,j)$에 곱하게 되며 이 가중치를 분류 속성 가중치라고 한다. 아이템이 속한 속성 사이의 유사관계 순위를 정하고 각 등급에 따라 가중치를 차등하여 정의한다. [그림 2]는 아이템 유사도 검색 테이블을 생성하기 위한 아이템간의 유사도 측정의 개념도이다.

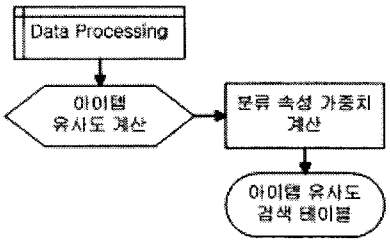


그림 2. 아이템간의 유사도 측정 개념도

예를 들어 아이템 A와 아이템 B가 속해있는 속성이 같다면 A와 B의 속성의 유사도가 높다고 가정할 수 있으며, 반대로 A와 B의 속성이 전혀 관련성이 없다면 두 아이템 사이의 속성 유사도는 상대적으로 낮다고 가정할 수 있다. 이를 기반으로 아이템 속성간의 유사관계에 관한 설문 조사하고, 그 결과를 분석한 후 속성간의 유사관계 순위를 결정한다. 이 순위에 따라 각 속성간의 분류속성 가중치를 결정하며 별도로 저장하게 된다. 아이템 i와 j가 각각 속한 속성 사이의 분류속성 가중치를 w_{ij}라고 하며, 이 가중치를 고려한 유사도 s_{ij}는 (식 4)로 정의한다. [알고리즘 1]은 아이템 기반 유사도 계산하는 방법이다.

$$s_{i,j} = w_{i,j} \frac{\sum_{u \in U}(R_{u,i} - \overline{R_i})(R_{u,j} - \overline{R_j})}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R_i})^2} \sqrt{\sum_{u \in U}(R_{u,j} - \overline{R_u})^2}}$$

$$0 < w_{i,j} \leq 1 \qquad \text{(식 4)}$$

알고리즘 1. 아이템 기반 유사도 계산

```
Begin
 For (All item i in Item-list)
 {
   For (All item j in Item-list)
    {
     Select user list who co-rate for item i and j
     compute sim(i,j)
     si,j = sim(i,j) × wi,j
    }
 }
End
```

마지막 단계에서는 계산된 아이템간의 유사도를 기반으로 하여, 각 아이템과 유사한 아이템 집합과 유사도를 나타내는 아이템 유사도 검색 테이블을 생성한다. 각 아이템에 대응하는 유사 아이템 집합의 크기는 해당 유사도를 내림차순으로 정렬하여 k개의 원소를 갖도록 한다. 아이템 유사도 검색 테이블의 구성은 [표 1]과 같다.

표 1. 아이템 유사도 검색 테이블

item	Similar Item Set
i_1	$(i_{17} \mid s_{1,\ 17}),\ (i_8 \mid s_{1,\ 8}),\ \cdots\ \cdots,\ (i_{kth} \mid s_{1,\ kth})$
i_2	$(i_{209} \mid s_{2,\ 209}),\ (i_4 \mid s_{1,\ 4}),\ \cdots\ \cdots,\ (i_{kth} \mid s_{2,\ kth})$
...	... ...
i_N	$(i_{92} \mid s_{N,\ 92}),\ (i_{20} \mid s_{N,\ 20}),\ \cdots\ \cdots,\ (i_{kth} \mid s_{N,\ kth})$

3.2 사용자 기반 유사도 측정

명시적 데이터를 사용한 유사도 측정에서 기인하는 희박성 문제를 보완할 수 있는 방법으로써 묵시적 데이터, 즉 사용자가 아이템에 접근한 기록을 이용하여 유사한 사용자를 검색한 후 추천하는데 사용될 수 있다. [그림 3]은 사용자 기반 유사도 측정의 개념도이다.

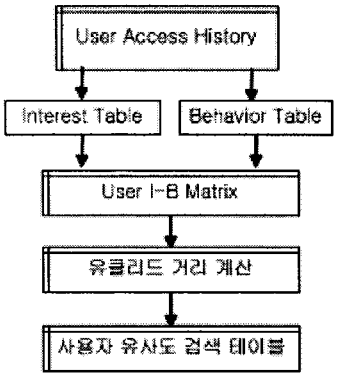


그림 3. 사용자 기반 유사도 측정 개념도

각각의 사용자는 [표 2]와 같은 형태의 접근 기록을 가지게 된다. 이 접근 기록은 트랜잭션 번호와 각 트랜잭션에 있는 아이템 그룹으로 구성된다. 본 논문에서 다루고 있는 아이템 그룹은 MovieLens[6]에서 각 아이템에 관한 10가지 속성을 기준으로 정한다. 이 접근 기록을 바탕으로 해당 사용자의 Interest Table을 작성하는데, 그 형태는 [표 3]과 같다. 이 테이블은 사용자가 각 아이템 그룹에 대해 접근한 시작 트랜잭션(FT)과 마지막 트랜잭션(LT), 횟수($Count$), 지지도($Support$)를 기록한다. 지지도는 (식 5)로 계산된다[7]. 여기서 T_c와 FT는 현재 트랜잭션의 번호와 시작 트랜잭션 번호를 의미한다. 일반화하기 위해 T_c라고 표기하였으나, 실제 계산에 사용된 데이터는 이미 수집이 완료된 데이터이므로 마지막 트랜잭션(LT)이 T_c로써 사용되었다. 구체적인 예로 T1의 트랜잭션에 아이템 그룹은 {A, C, E}이 있다. 여기서 사용자가 5개의 트랜잭션에서 아이템 그룹에 속해있는 A를 2번 접근하였다면, (식 5)에 의해서 Support는 0.4가 된다.

$$Support = \frac{Count}{T_c - FT + 1} \qquad \text{(식 5)}$$

표 2. 사용자의 접근 기록

Transaction	Item Groups in Transaction
T1	A, C, E
T2	B, C, E, F
...	...
Tx	A, G

표 3. Interest Table

Item Group	Count	FT	LT	Support
A	2	T1	T5	0.4
B	2	T2	T4	0.5
...	...	...	...	...
G	1	T5	T5	1

표 4. Behavior Table

Item Group Pair	Count	FT	LT	Support
AC	1	T1	T1	0.2
AE	1	T1	T1	0.2
...	...	...	...	...
EF	2	T2	T3	0.5

표 5. I-B Matrix

	A	B	C	D	E	F	G
A	0	0	0	0	0	0	1
B		0	0	0	0	0	0
C			1	1	0	1	0
D				0	0	1	0
E					0	0	0
F						1	0
G							1

동일한 방법과 식을 사용하여 2개의 아이템에 대한 지지도를 나타내는 사용자의 Behavior Table을 [표 4]와 같은 형태로 생성한다. 이 테이블의 구성은 Interest Table과 동일하다. 이렇게 하여 얻은 두 개의 테이블에는 모두 지지도가 기록되어 있다. 최소 지지도를 만족하는 아이템에 대하여 [표 5]와 같은 형태의 I-B Matrix를 구성할 수 있다. 사용자는 고유의 I-B Matrix를 가지게 되며 (식 6)의 유클리드 거리 계산식을 사용하여 사용자 i와 j사이의 유사도 d_{ij}를 측정할 수 있다 [7].

$$d_{i,j} = \| i - j \| = \sqrt{\sum_{l=1}^{n} (i_l - j_l)^2} \qquad \text{(식 6)}$$

사용자간의 유사도가 계산되면 아이템 유사도 검색 테이블을 생성한 것과 같은 방법으로 [표 6]과 같은 사용자 유사도 검색 테이블을 생성할 수 있다. 전체 사용자의 수를 M이라고 할 때, 사용자간의 유사도를 계산하는데 O(M(M-1)/2)의 비용이 필요하게 된다. 아이템 기반 유사도 측정에서 다루고 있는 유사도는 그 값의 크기에 비례하지만, 사용자의 유사도를 측정하는 척도로써 거리를 사용하였기 때문에 유사도와 거리는 반비례한다는 점이 다를 뿐이다. 사용자 기반 유사도 측정의 방법은 아이템 기반의 방법에서 간과하고 있는 사용자의 기호와 행동양식에 대한 패턴을 참조 대상으로 삼고 있으며, 최근의 방법에 비해 정확성 측면에서 부족한 유클리드 거리 계산을 사용하지만 비교적 간단한 계산식을 사용함으로써 추가적인 오버헤드의 부담 없이 사용자 유사도 검색 테이블을 생성할 수 있다.

표 6. 사용자 유사도 검색 테이블

user	similar user set
u_1	(u_{17}|d_1, 17), (u_8|d_1, 8), ···, (u_{kth}|d_1, kth)
u_2	(u_{209}|d_2, 209), (u_4|d_1, 4), ···, (u_{kth}|d_2, kth)
...	...
u_M	(u_{92}|d_M, 92), (u_{20}|d_M, 20), ···, (u_{kth}|d_M, kth)

3.3 Naive Bayesian을 사용한 추천

아이템 유사도 검색 테이블과 사용자 유사도 검색 테이블을 생성하였다. 사용자 u_c에 의한 추천 요구가 발생하였을 때, 우선 아이템 유사도 검색 테이블에서 사용자 u_c의 최근 트랜잭션에서 접근이 있었던 아이템과 유사한 k개의 아이템 집합 {ic_1, ic_2, ···, ic_k}을 탐색한다. 여기서 사용자 u_c가 접근했던 아이템과 중복되는 유사 아이템은 제거된다. 동시에 사용자 유사도 검색 테이블에서 추천을 요구한 사용자와 유사한 m명의 사용자를 선택하며, 사용자 u_c와 유사한 사용자 {u_1, u_2, ···, u_m}의 최근 트랜잭션에서 접근이 있었던 아이템의 합집합 {i_1, i_2, ···, i_n}을 구한다. 사용자 u_c와 유사한 사용자의 트랜잭션으로부터 추출할 수 있는 아이템의 개수는 사용자 유사도 거리에 따른 가중치에 따라 (식 7)로 구할 수 있다.

$$A = \sum_{i=1}^{m} d_{c,i} \qquad B_i = \frac{A - d_{c,i}}{A} \qquad C = \sum_{i=1}^{m} B_i$$

$$n_i = n^* \frac{B_i}{C}\left(n = \sum_{i=1}^{m} n_i\right) \qquad \text{(식 7)}$$
$$1 \le i \le m, \quad n \ge m$$

이상 3개의 집합은 [표 7]의 형태로 구성된다. [표 7]의 아이템 ic_i의 클래스는 아이템 유사도 검색 테이블에서 사용자 u_c의 최근 트랜젝션에서 접근이 있었던 아이템과 유사한 k개의 아이템 집합 $\{ic_1, ic_2, \cdots, ic_k\}$을 의미한다. [표 7]의 값들은 사전 계산을 통해 추출된 유사 아이템에 대한 유사 사용자의 선호도를 의미한다.

표 7. Naive Bayesian을 적용하기 위한 테이블

	i_1	i_2	⋯	i_n	ic_1	ic_2	⋯	ic_k
u_1	0.8	0	⋯	0	0	0.2	⋯	1
u_2	0.6	1	⋯	0.4	1	0	⋯	0.8
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
u_m	0.2	0	⋯	1	0.2	0.6	⋯	0.6
u_c	0	0.6	⋯	1	?	?	⋯	?

아이템 i_k에 대한 사용자 u_c의 예측은 (식 8)의 Naive Bayesian을 사용하여 사후확률 $P(C_{ik}|X)$를 최대화함으로써 판단하게 된다.

$$P(C_{ik}|X) = \frac{P(X|C_{ik})P(C_{ik})}{P(X)} \qquad \text{(식 8)}$$

X는 사용자 u_c가 가지는 조건, 즉 $\{i_1, i_2, \cdots, i_n\}$에 대한 평가를 나타내며, C_{ik}는 아이템 i_k의 클래스(0, 0.2, 0.4, 0.6, 0.8, 1)를 나타낸다. $P(C_{ik}|X)$는 조건 X일 경우 C_{ik}라는 클래스의 확률을 말한다. 사전확률 $P(X)$는 일정하기 때문에 $P(X|C_{ik})P(C_{ik})$에 대해서만 고려한다면, 이것은 (식 9)로 표현한다.

$$P(X|C_{ik})P(C_{ik}) = P(C_{ik})\prod_{j=1}^{n} P(x_j|C_{ik}) \qquad \text{(식 9)}$$

추천 아이템 $\{ic_1, ic_2, \cdots, ic_k\}$의 각 클래스에 대한 확률이 계산된다면 가장 높은 확률을 가지는 클래스를 선택함으로써 각 아이템의 평가치를 예측할 수 있으며, 예측 값의 크기에 따라 추천 순서를 결정할 수 있다. 실제 계산에서 조건 확률의 분모가 0이 되는 것을 방지하기 위해 라플라시안 정밀도를 사용한다[8].

4. 실험 방법

4.1 실험 방법 및 결과

본 논문에서 제안한 방법은 MS Visual C++ 6.0으로 구현되었으며, 실험 환경은 Pentium-4 1.6 Ghz, 512MB RAM 환경에서 수행되었다. 실험 방법은 본 논문에서 제안한 방식을 3가지로 구분하였다. 첫 번째 방법(ICF+WCA)은 기존의 아이템 기반 협력적 필터링에 아이템의 분류속성 가중치를 적용한 방법이고, 두 번째 방법(IUCF+NB)은 아이템과 사용자를 기반으로 하는 협력적 필터링에 Naive Bayesian을 적용한 방법이다[11]. 마지막 방법(IUCF+WCA+NB)은 두 번째 방법에 아이템 분류속성을 적용한 방법이다.

MovieLens[6]를 전처리하여 30,861명의 사용자와 1,612 종류의 영화, 1,574,431개의 평가 데이터에 대해서 실험을 진행하였다. 먼저 사용자와 아이템에 대한 유사도 검색 테이블 생성을 위한 훈련 데이터 집합을 만들기 위해 평가 데이터에서 2개월 동안 기록된 데이터를 추출하였으며, 추출된 평가 데이터에 포함되지 않은 사용자의 데이터를 제거하였다. 그리고 사용자 데이터 중 20개 미만의 평가 데이터를 가지고 있는 사용자 데이터를 삭제하여 그 결과를 평가 데이터에 반영하였다. 결국 1,000명의 사용자 데이터와 67,963개의 선호도 데이터, 1,612편의 아이템 데이터를 훈련 데이터로써 사용하였다. 성능 측정을 위한 실험 데이터로써 1개월여 동안 기록된 평가 데이터 중 훈련 데이터에 포함된 사용자의 평가 데이터를 사용하였다.

아이템 기반 협력적 필터링을 위한 첫 번째 단계에서는 개선된 코사인 유사도를 사용하여 402,988개의 유사도를 가지는 아이템 유사도 검색 테이블을 생성하였다. [그림 4]는 아이템 유사도 검색 테이블의 일부분이다. 두 번째 단계에서는 아이템 속성에 따른 분류속성 가중치를 정의하기 위해 일반인 30명을 대상으로 아이템 속성간의 유사관계에 관한 설문조사를 [그림 5]와 같이

하였다. 설문조사를 할 때 가장 유사도가 높은 관계에 대해서는 1을, 가장 유사도가 낮은 관계에 대해서는 5를 주도록 요구하였다. 설문조사 결과를 바탕으로 아이템 속성간의 유사관계에 관한 5등급의 순위를 결정하고 각 등급에 따라 1, 0.98, 0.95, 0.92, 0.9의 분류속성 가중치를 부여하였다. 각 유사도 순위에 따라 이와 같은 가중치를 부여한 이유는 순위에 따른 임의적 값이 실험에 미치는 영향을 보고자 하였다. 속성간의 분류속성 가중치는 [표 8]과 같이 정의하였다. 이를 바탕으로 아이템 유사도 검색 테이블을 재구성하였다. [그림 6]은 재구성된 아이템 유사도 검색 테이블의 일부분이다.

Item_i	Item_j	Sim
230	160	0.255979899297
230	157	0.033159515354
230	156	0.281018168653
230	159	0.619463106505
230	154	-0.12505425414
230	155	0.385753011489
230	158	-0.38287757748
230	117	-0.85583737341
230	116	-0.18817195924
230	121	-0.28828417517
230	122	-0.54589319096
230	113	0.137423298042
230	119	-0.51289485201
230	114	-1
230	123	-0.70491100267
230	120	1
230	118	0.165747052301
230	130	1

그림 4. 아이템 유사도 검색 테이블

	Action	Animation	Art Foreign	Classic	Comedy	Drama	Family	Horror	Romance	Thriller
Action	1									
Animation		1								
Art Foreign			1							
Classic				1						
Comedy					1					
Drama						1				
Family							1			
Horror								1		
Romance									1	
Thriller										1

그림 5. 아이템 속성간의 유사관계 순위

표 8. 분류속성 가중치

Genre_i	Genre_j	Weight rank	Weight
Action	Action	1	1
Family	Romance	2	0.98
Horror	Thriller	2	0.98
Comedy	Drama	3	0.95
Drama	Romance	3	0.95
Animation	Family	3	0.95
Art Foreign	Family	4	0.92
Classic	Comedy	4	0.92
Action	Drama	4	0.92
Animation	Thriller	5	0.9
Action	Family	5	0.9
Action	Romance	5	0.9

Item_I	Item_J	Genre_I	Genre_J	Sim	WCA	Sim*
230	160	thriller	horror	0.2559798993	0.98	0.25086030364
230	157	thriller	comedy	0.0331595154	0.92	0.03050675429
230	156	thriller	comedy	0.2810181687	0.92	0.25853672624
230	159	thriller	drama	0.6194631065	0.95	0.58848994970
230	154	thriller	art	-0.1250542541	0.92	-0.1150499135
230	155	thriller	drama	0.3857530115	0.95	0.36646535993
230	158	thriller	family	-0.3828775775	0.95	-0.3637337089
230	117	thriller	horror	-0.8558373734	0.98	-0.8387206197
230	116	thriller	art	-0.1881719592	0.92	-0.1731182039
230	121	thriller	drama	-0.2882841752	0.95	-0.2738699615
230	122	thriller	romance	-0.545893191	0.98	-0.5349753499
230	113	thriller	drama	0.137423298	0.95	0.13055212796
230	119	thriller	comedy	-0.512894852	0.92	-0.4718632698
230	114	thriller	art	-1	0.92	-0.9200000167
230	123	thriller	art	-0.7049110027	0.92	-0.6485181451
230	120	thriller	comedy	1	0.92	0.92000001669
230	118	thriller	romance	0.1657470523	0.98	0.16243210435
230	130	thriller	drama	1	0.95	0.94999998808
230	130	thriller	drama	1	0.95	0.94999998808

그림 6. 재구성된 아이템 유사도 검색 테이블

사용자 유사도 검색 테이블의 생성을 위한 첫 번째 단계로 각 아이템에 대한 사용자의 평가 데이터로부터 일간 트랜잭션에 근거한 사용자의 접근 기록을 생성하였다. 두 번째 단계에서 이 기록에 의해 Interest Table과 Behavior Table을 생성한다. 세 번째 단계에서는 두 개의 테이블에 대해 Minimum Support=50%를 적용하여 [그림 7]의 IB-Matrix와 [그림 8]의 사용자 유사도 검색테이블을 생성하였다. 훈련 데이터 집합으로 생성한 아이템과 사용자 유사도 검색 테이블로부터 실험 데이터에서 임의로 추출한 사용자 u_c와 관련있는 유사 아이템과 유사 사용자를 검색하여 Naive Bayesian을 [표 9]와 같이 적용하였다.

User_ID	IB_Matrix
875	0100001110001010000010010000000000010000000000000010110000100000000100000
1918	0100001110001000000010010000100000000000000000000010000110000000000001100010
2198	0101101010001101010111110011011101001000010101000001000110100101001110000
2487	0100111100001010001010000000000000000000000000000000000011000010000000100000
3361	1111111111001101000101111001111111000101010101000100011110010100000011100000
3472	1110111110001010001011000000000000000000000000000100011000101100000010000000
3500	0101111110011010001010111010010010001010100010100010001101110101010001101000
4806	0110001100001010000010000000000000000000000010100110000100001000000000000100010
5688	0100001110001000001010100000000000010101010100000000000000010000000000010
6230	0010000000011010000011000000100010000000000011000100001100000101000011110000
7437	0000001010001000010000000000000000000000000000000000000000000000000000000000
9117	0100001110001010001010100000000000000010100110000010001100000110000000000000
9503	0000001100001010000000100000000000000010100000100000100011000001000000000100010
11205	1111011110011010101110011100100010001010111010010110001110101110101011100011
11504	0101111010011010001001100010011000001010000000000100110000011000001111100000
11963	1110111010011010000010011111111111111010101000000000111000010000010111000 1
12664	1001101000001011001000010000000010000000000000000000100010000001000000000000
12876	0101111111001101000110001000100110001000000000000000110011010001000011110010
12937	0111001010001000001000000000010000000000000000000000000110000011010000000000
13275	0101000010001010000000010000000000000000000100010000000001000000111010000000000
13712	0101001110000000001000000000000000000000000000000000000011010011100000000010
13837	0111111110001010100001011111000000010001010100000000111010001000000000001
13856	0100101110001010001010100000000000000010101000100000000011000001110000011010 10
14055	0101011100001000001010100000000000000000000000000000000111011000001100000011000 00
14228	0100101100001010000010000000000000000000000010010010000010001100000111000001000000
16250	0100001010001010001000000000000000000000000010000000000000110000010000000000000

그림 7. IB-Matrix

User_i	User_j	Distance
67730	22771	3.16227766016838
67730	22885	3.46410161513775
67730	22972	3.46410161513775
67730	23061	3.3166247903554
67730	23135	4.47213595499958
67730	23337	3.46410161513775
67730	23362	3.74165738677394
67730	23372	6.2449979983984
67730	23373	2.82842712474619
67730	23846	3.60555127546399
67730	23912	5.3851648071345
67730	24022	3
67730	24042	3.3166247903554
67730	24143	3.46410161513775
67730	24832	3
67730	25364	3
67730	25373	3.87298334620742
67730	25436	3.3166247903554
67730	25548	4.24264068711928
67730	25593	3.16227766016838
67730	25724	3.16227766016838
67730	25967	3.16227766016838
67730	26140	3.3166247903554
67730	26348	3.3166247903554
67730	26366	3.60555127546399
67730	26411	3.3166247903554
67730	26543	4.69041575982343

그림 8. 사용자 유사도 검색 테이블

표 9. Naive Bayesian 적용 예

	180	1607	1544	94	1623	780	1580	1591	1393	619	594
38245	1	0.8	0	0	0	1	0.6	0	0	0	0
52020	0	0	0.6	0.4	0	0.6	0	0	0	0	0
44650	0	0	0.4	0	0.8	0.4	0	0	0.4	0	1
48372	0	0	0	0	0	1	1	0.8	1	0	0
31992	0	0	0	0	0	1	0	0	0.8	0.6	0.8
1	0	0	0.4	0	0	0.6	0.8	0	0.6	0	0.6

예측 알고리즘을 평가하는 방법 중에서 정확성 측면에서 성능을 평가하기 위해 MAE[9,10]을 사용하였다. IUCF+NB 방법과 IUCF+WCA+NB 방법에서는 Naive Bayesian을 적용한 경우에 10명의 유사 사용자를 참조하였으며, 각 방법의 모델 사이즈 n에 따라 실험하였다. 기존의 아이템 기반 협력적 필터링 방법인 ICF 방법과 ICF+WCA 방법에서의 n이란 추천을 요구한 사용자에게 추천될 유사 아이템 개수를 의미하며, IUCF+NB 방법과 IUCF+WCA+NB 방법에서의 n이란 유사 사용자의 트랜잭션을 참고하여 조건 확률을 계산하기 위해 사용된 아이템의 개수를 의미한다. 각 방법에 따른 실험은 10명의 사용자가 각각 10개의 아이템에 대한 추천을 요구했을 때의 경우를 가정하여 이루어졌으며, 동일한 사용자 리스트가 모든 방법에 적용되었다. [그림 9]는 MAE에 의한 성능 평가를 나타낸다.

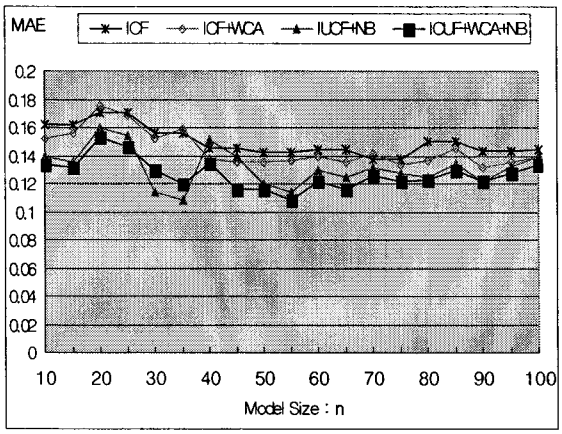


그림 9. MAE에 의한 성능 평가

[그림 9]에서 기존의 아이템 기반 협력적 필터링 방법과 비교해 볼 때 정확도가 향상된 것을 볼 수 있다. ICF+WCA 방법은 기존 대비 3.42%의 MAE 감소를 보였으며, IUCF+NB 방법과 IUCF+WCA+NB 방법은 각각 11.94%와 15.28%의 MAE 감소를 나타내고 있다. MAE의 감소는 정확성 측면에서의 향상을 의미한다. IUCF+NB 방법과 IUCF+WCA+NB 방법의 경우 모델 사이즈 n이 55에서 가장 좋은 정확성 향상을 보여주었으며, 그 이상으로 증가할수록 기존의 ICF 방법의 정확성에 근접함을 알 수 있다. 분류속성 가중치를 사용한 IUCF+WCA+NB 방법의 전체적인 정확도가 IUCF+NB 방법보다 뛰어나긴 하지만, 구간에 따라서는 IUCF+NB 방법보다 정확성이 감소하는 것을 볼 수 있다. 본 논문에서 제안한 방법이 정확성의 향상이라는 측면에서는 긍정적인 결과를 보여주고 있지만, Naive Bayesian에 의한 계산이 기존의 협력적 필터링에 의한 계산보다 많은 데이터를 필요로 하며, 그에 따라 저장 공간에 대한 액세스 시간이 증가하는 현상을 보여주었다. 정확하고 객관적인 속도 측정 실험을 통한 성능 분석이 필요하며, 앞으로 이 문제를 보완해야 할 것으로 보인다. 또한 트랜잭션 데이터를 포함하고 있지 않은 MovieLens 실험에 사용한 관계로 묵시적 데이터에 근거한 사용자 유사도의 참조에 의한 성능 향상이 미흡하다고 판단되며, 정확한 실험을 위해 사용자 로그 프로파일을 포함한 데이터에 대한 구현이 필요하다.

5. 결 론

사용자 기반 협력적 필터링의 문제점인 희박성과 확장성에 대해서 아이템 기반 협력적 필터링이 주목할 만한 성과를 거두었다. 특히 확장성에 대한 성능 향상은 실시간 추천을 요구하는 웹 기반 추천 시스템에 많은 기여를 하였다. 그러나 기본적으로 명시적 데이터에 기반한 접근 방법이기 때문에 희박성 문제가 남아 있으며, 아이템간의 속성을 고려하지 않는다는 단점을 지적받아왔다. 본 논문에서는 기존의 아이템 기반 협력적 필터링의 희박성 문제를 보완하기 위하여 기존의 아이템 유사도를 이용하여 가중치 합을 예측 방법으로 사용하는 대신에 아이템과 사용자의 유사도를 복합 참조하여 Naive Bayesian을 적용하였다. 묵시적 데이터인 사용자의 트랜잭션 데이터를 바탕으로 하여 사용자 유사도 검색 테이블을 아이템 유사도 검색 테이블과 함께 복합 참조하는 부분이 제안하고자 하는 주개념이다. 그리고 아이템의 속성을 반영하지 못했던 문제점을 해결하기 위해 아이템을 분류하는 속성간의 유사 관계도를 설정하고, 그에 따라 분류속성 가중치를 정의하였다. 정의된 가중치는 기존의 방법에 의해 계산된 아이템 유사도에 적용되었다. 제안된 방법의 성능을 평가하기 위하여 아이템 기반 협력적 필터링과 비교한 결과 예측의 정확도가 향상되었기 때문에 제안한 방식이 정확도면에서 효과적임을 알 수 있었다.

참 고 문 헌

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-Item Collaborative Filtering," Internet Computing, IEEE, Vol.7, No.1, pp.76-80, 2003.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study," In ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.

[3] G. Karypis, "Evaluation of Item-Based Top-N Recommendation Algorithm," Technical Report CS-TR-00-46, Computer Science Dept., University of Minnesota, 2000.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based Collaborative Filtering Recommendation Algorithms," Proc. of the 10th International Conference on WWW, pp.285-295, 2001.

[5] J. Han and M. Kamber, *Data Mining: Concept and Techniques*, Morgan Kaufmann, Chapter 7, pp.296-298, 2001.

[6] http://www.cs.umn.edu/Research/GroupLens/

[7] H. C. Chen and A. L. P. Chen, "Collaborative Filtering and Algorithms : A Music Recommendation System based on Music Data Grouping and User Interests," Proc. of the International Conference on Information and Knowledge Management, pp.231-238, 2001.

[8] K. Miyahara and M. Pazzani, "Collaborative Filtering with the Simple Bayesian Classifier," Proc. of the International Conference on Artificial Intelligence, pp.679-689, 2000.

[9] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-Boosted Collaborative Filtering for Improved Recommendations," Proc. of the National Conference on Artificial Intelligence, pp.187-192, 2002.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," ACM Transactions on Information Systems, Vol.22, No.1, pp.5-53, 2004.

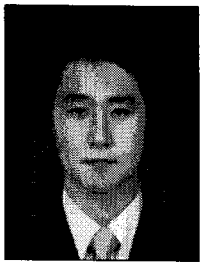
[11] 김용집, 정경용, 이정현, "사용자와 아이템의 혼합 협력적 필터링에서 Naive Bayesian 알고리즘을 이용한 추천 방법", 제30회 한국정보과학회 추계학술발표 논문집(I), pp.184-186, 2003.

[12] N. Good, J. B. Schafer, J. A. Konstan, A.

Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations," Proc. of the Conference on Artificial Intelligence, pp.439-446, 1999.

저 자 소 개

김 종 훈(Jong-Hun Kim) 정회원



- 2001년 2월 : 인천대학교 물리학과 (학사)
- 2003년 2월 : 인하대학교 전자계산공학과 (공학석사)
- 2004년 8월 ~ 현재 : 인하대학교 컴퓨터정보공학과 박사과정

<관심분야> : 임베디드/유비쿼터스 시스템, 데이터마이닝, 인공지능

김 용 집(Yong-Jip Kim) 정회원

- 2002년 2월 : 인하대학교 컴퓨터공학과(공학사)
- 2004년 2월 : 인하대학교 컴퓨터공학과(공학석사)
- 2004년 3월 ~ 2006년 8월 : MEDIA CHORUS Co.,Ltd. 연구원
- 2006년 9월 ~ 현재 : ACCESS SEOUL Co.,Ltd. 연구원

<관심분야> : 데이터마이닝, 모바일서비스

정 경 용(Kyung-Yong Chung) 정회원

- 2000년 2월 : 인하대학교 전자계산공학과(학사)
- 2002년 2월 : 인하대학교 컴퓨터정보공학과(공학석사)
- 2005년 8월 : 인하대학교 컴퓨터정보공학과(공학박사)
- 2005년 8월 ~ 2006년 2월 : 한세대학교 IT학부 교수
- 2006년 3월 ~ 현재 : 상지대학교 컴퓨터정보공학과 교수

<관심분야> : 데이터마이닝, 지능시스템, 인공지능

임 기 욱(Kee-Wook Rim) 정회원

- 1977년 : 인하대학교 전자공학과(공학사)
- 1987년 : 한양대학교 전자계산학(공학석사)
- 1994년 8월 : 인하대학교 전자계산학(공학박사)
- 1977년 ~ 1983년 : 한국전자기술연구소 선임연구원
- 1983년 ~ 1988년 : 한국전자통신연구소 시스템소프트웨어 연구실장
- 1989년 ~ 1996년 : 한국전자통신연구원 시스템연구부장, 주전산기(타이컴)III,IV 개발사업 책임자
- 1997년 ~ 1999년 : 정보통신연구진흥원 정보기술전문위원
- 2000년 ~ 현재 : 선문대학교 컴퓨터정보학부 교수

<관심분야> : 실시간데이터베이스시스템, 운영체제, 시스템구조

이 정 현(Jung-Hyun Lee) 정회원

- 1977년 : 인하대학교 전자공학과(공학사)
- 1980년 : 인하대학교 대학원 전자 공학과(공학석사)
- 1988년 : 인하대학교 대학원 전자 공학과(공학박사)
- 1979년 ~ 1981년 : 한국전자기술 연구소 시스템 연구원
- 1984년 ~ 1989년 : 경기대학교 전자계산학과 교수
- 1989년 ~ 현재 : 인하대학교 컴퓨터공학부 교수

<관심분야> : 자연어처리, HCI, 정보검색, 컴퓨터구조