http://dx.doi.org/10.3745/KIPSTB.2012.19B.1.031

평가의 시간 순서를 고려한 강화 학습 기반 협력적 여과

이 정 규[†] · 오 병 화[††] · 양 지 훈[†††]

요 약

최근 사용자의 흥미에 맞는 아이템이나 서비스를 추천해 주는 추천 시스템에 대한 관심이 높아지고 있다. 최근 종료된 Netflix 경연대회(Netflix Prize)가 이 분야에 대한 연구자들의 연구 의욕을 고취시켰고, 특히 협력적 여과(Collaborative Filtering) 방법은 아이템의 종류에 상관없이 적용 가능한 범용성 때문에 활발히 연구되고 있다. 본 논문은 강화 학습을 이용해서 추천 시스템의 협력적 여과 문제를 푸는 방법을 제안한다. 강화 학습을 통해, 영화 평점 데이터에서 각 사용자가 평점을 매긴 순서에 따른 평점 간의 연관 관계를 학습하고자 하였다. 이를 위해 협력적 여과문제를 마르코프 결정 과정(Markov Decision Process)로 수학적으로 모델링하였고, 강화 학습의 가장 대표적인 알고리즘인 Q-learning을 사용해서 평가의 순서의 연관 관계를 학습하였다. 그리고 실제로 평가의 순서가 평가에 미치는 영향이 있음을 실험을 통해서 검증하였다.

키워드 : 기계학습, 추천 시스템, 협력적 여과, 강화 학습

A Reinforcement Learning Approach to Collaborative Filtering Considering Time-sequence of Ratings

Jungkyu Lee[†] · Byonghwa Oh[††] · Jihoon Yang[†††]

ABSTRACT

In recent years, there has been increasing interest in recommender systems which provide users with personalized suggestions for products or services. In particular, researches of collaborative filtering analyzing relations between users and items has become more active because of the Netflix Prize competition. This paper presents the reinforcement learning approach for collaborative filtering. By applying reinforcement learning techniques to the movie rating, we discovered the connection between a time sequence of past ratings and current ratings. For this, we first formulated the collaborative filtering problem as a Markov Decision Process. And then we trained the learning model which reflects the connection between the time sequence of past ratings and current ratings using Q-learning. The experimental results indicate that there is a significant effect on current ratings by the time sequence of past ratings.

Keywords : Machine Learning, Recommender Systems, Collaborative Filtering, Reinforcement Learning

1. 서 론

추천 시스템은 많은 양의 이용 가능한 아이템(Item)중에서 사용자(User)의 흥미에 맞는 아이템을 추천해주는 시스템을 말한다. 여러 가지 추천 시스템 기법 중 가장 많이 사용되는 방법은 협력적 여과로서, Amazon.com, CDnow.com 등의 상업적으로 성공한 여러 전자상거래 사이트에서 활용되고 있다[1].

전통적인 협력적 여과 방식에서는 사용자 목록과 아이템 목록을 이용한다[2]. 이를 통해 행렬을 구성하는데, 각 행렬의 항목 값은 사용자의 아이템에 대한 의견을 나타낸다. 사용자의 의견은 평점과 같은 사용자의 평가를 통해 직접적으로 획득하거나, 또는 사용자의 구매 기록이나 이용 패턴, 특정 페이지에서의 시간 기록 등을 분석해서 간접적으로 얻기도 한다.

미국의 DVD 대여 회사인 Netflix는 Netflix Prize 경진대회[3]을 개최하였는데, 이는 자사가 현재 갖추고 있는 협력적 여과 기반의 추천 시스템보다 10퍼센트 이상 향상된 성능의 알고리즘을 제안하는 시스템 개발에 대해서 100만 달러의 상금을 수여하는 대회였다. 이 대회를 통해 협력적 여과 문제를 해결하는 여러 알고리즘이 개발되었다. 대표적인 알고리즘은 특이 값 분해(Singular Value Decomposition, SVD)[4], 제약적 볼츠만 머신(Restricted Boltzmann Machine, RBM)[5], K-최근접 이웃(K-Nearest Neighbor,

† 정 회 원 : (주)사이람 연구원
†† 준 회 원 : 서강대학교 컴퓨터공학과 박사과정
††† 종신회원 : 서강대학교 컴퓨터공학과 교수
논문접수 : 2011년 5월 18일
수 정 일 : 1차 2011년 7월 28일
심사완료 : 2011년 7월 31일

KNN)[6] 알고리즘 등이 있다. 이 대회의 우승팀인 BellKor's Pragmatic Chaos 팀은 위의 세 가지 알고리즘을 기반으로 하여 100여개 이상의 예측기를 학습한 후, 이들을 앙상블(Ensemble)한 예측기로 우승을 하였다.

본 연구에서는 협력적 여과 문제를 해결하는 새로운 방법으로, 강화 학습 기법(Reinforcement Learning)을 사용하는 방식을 제안한다. 강화 학습은 에이전트가 경험을 통해 환경에 대해 적응해 가는 과정을 모델링한 연구 분야로서, 보행 로봇의 이동, 헬기의 자동 비행, 네트워크 라우팅, 마케팅 전략 선택, 웹 인덱싱 등 많은 분야에서 효율적이라고 평가된다. 그러나 본 연구 이전에, 협력적 여과 문제를 강화 학습 방식으로 해결한 연구는 없었다.

논문의 구성은 다음과 같다. 2장에서 본 연구가 제안하는 알고리즘의 기본 예측기 및 성능 비교에 쓰이는 SVD에 대해 간단히 소개한다. 3장에서는 본 연구가 제안하는 협력적 여과 문제를 강화 학습으로 해결하기 위해 필요한 문제 정의를 한다. 4장에서는 제안 알고리즘에 대해 설명한 후, 5장에서 이 제안 알고리즘에 대한 여러 실험 결과를 제시한다. 6장에는 결론 및 향후 과제에 대해 논의한다.

2. 관련 연구(Related Works): SVD

SVD 알고리즘은 협력적 여과 문제를 해결하는 단일 알고리즘 중 가장 예측 정확도가 높은 알고리즘이다. SVD는 본래 자연어 처리 응용분야에서 효율적이라 알려졌으나[7], Simon Funk(Brandyn Webb의 예명)가 Netflix 대회에서 협력적 여과 문제에 SVD를 사용할 수 있음을 최초로 제안하였다[8]. SVD는 예측의 정확성이 높을 뿐만 아니라 SVD의 결과로 나오는 행렬이 매우 유용하다. 본 연구에서도 SVD의 예측과 결과 행렬을 사용하였기 때문에 SVD 알고리즘에 대해서 간단히 알아보겠다.

영화의 개수를 M, 사용자수를 N이라 하자. SVD는 완전히 관찰된 목표 행렬(Fully Observed Target Matrix) A와의 합-제곱 거리(Sum-Squared Distance)를 최소화하는 저계수(Low-Rank) 행렬 $V = UM'$을 찾는다. 여기서 $U \in R^{N\times K}$, $M \in R^{M\times K}$ 이고, K는 특성(Feature)의 차원이다. 수식으로 표현하면 다음과 같다.

$$f(U,M) = \sum_{(i,j)\in Y} (x_{ij} - \sum_{k=1}^{K} U_{ik}M_{kj})^2 + \lambda \sum_{(i,j)\in Y} (\|U\|^2 + \|M\|^2) \quad (1)$$

$$[U,M] = \arg\min_{U,M} f(U,M)$$

목적 함수 f의 첫 번째 항은 영화 평점 데이터베이스 상에 존재하는 실제 사용자-영화 평점 데이터 쌍의 집합 $(i, j) \in Y$에 대해 사용자 i와 영화 j에 대한 실제 평점과 SVD의 예측 평점 $\sum_{k=1}^{K} U_{ik}M_{kj}$ 사이의 제곱 오차의 합을 나타낸다. 목적 함수 f의 두 번째 항은 과적합(Overfitting)을 막기 위한 조정(Regulation) 작용을 한다. 수식 (1)을 최소화하기 위해 경사 강하법(Gradient Descent)을 사용한다.

협력적 여과 문제에 대한 SVD는 여러 Netflix Prize 참가자에 의해서 다양한 유형으로 발전하였는데 본 연구에서는 Netflix Prize의 1위 팀의 팀원인 Koren이 제안한 SVD++를 사용하였다[9].

3. 문제 정의

협력적 여과(Collaborative Filtering)는 어떤 아이템에 대해 많은 수의 사용자에 의해 주어진 취향 정보를 토대로 사용자의 선호도를 예측하는 방법이다. 어떤 사람이 과거에 어떤 아이템을 좋아한다면, 미래에도 그 아이템을 좋아할 것이라는 것이 협력적 여과의 기본적인 가정이다.

데이터베이스에는 N명의 사용자와 M개의 영화에 대한 선호도가 수치적으로 모아져 있다고 가정한다. 예를 들어, 사용자는 자신이 본 영화에 대해서 1에서 5까지의 평점을 줄 수 있다. 일반적으로 사용자는 데이터베이스 안에 있는 모든 영화에 대해서 점수를 매기지 않는다. 단지 사용자가 본 영화에 대해서만 점수를 매긴다. 그리고 각 사용자의 아이템 소비 형태가 다르기 때문에 어떤 사용자는 많은 영화를 볼 수도 있고, 어떤 사용자는 단지 몇 개의 영화만 볼 수 있다. $X \in R^{N\times M}$는 데이터베이스에 모아진 평점 행렬이다.

보통 영화 데이터베이스와 같이 아이템의 수가 매우 많은 경우 사용자가 점수를 매긴 비율이 전체 아이템의 수에 비해 매우 적으므로, 행렬 X는 매우 희소(Sparse)하다. 즉 대부분의 행렬의 원소는 결여(Missing)되어 있다. X는 협력적 여과 알고리즘의 훈련 데이터(Training Data)로 쓰인다.

협력적 여과 알고리즘의 목표는 데이터베이스 내에 결여치(Missing Value)를 추측해 내는 것이다. $A \in R^{N\times M}$는 결여된 평점 데이터가 실제 평점으로 모두 채워진 행렬이라고 하자. 협력적 여과의 성능은 예측 평점과 실제 평점과의 오차로 측정 할 수 있다. 실제 평점은 행렬 A에 기록되어 있기 때문에 성능평가는 테스트 행렬인 A를 이용해서 할 수 있다. 행렬 A는 오차를 계산하는 프로시저(또는 함수)에는 알려진다. 예를 들어, A는 데이터베이스 안에서 최근에 생성된 데이터의 집합으로 구성하거나 임의로 고를 수도 있다. 행렬 A에 들어간 데이터는 훈련 데이터 V에서는 제외된다. $I \in \{0, 1\}^{N\times M}$는 표시 함수(Indicator Function)인데, $I_{ij} = 1$ 이면 행렬 A에서 사용자 i가 영화 j에 대한 평점을 매겼음을 뜻하고, 매기지 않았으면 0이다. 알고리즘의 성능 평가 방법으로는 Netflix 경연대회에서 사용하는 근 평균 제곱 오차(Root Mean Squared Error, RMSE)를 사용하였다. $P \in R^{n\times m}$이 협력적 여과 알고리즘이 추측한 예측 행렬이라 할 때, RMSE는 다음 수식과 같이 정의된다.

$$RMSE(P,A) = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(A_{ij} - P_{ij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}}} \quad (2)$$

4. 제안 알고리즘

4.1 동기(Motivation)

어떠한 빛의 자극이 눈에 들어왔다가 제거되었을 때 시각 기관에 이에 대한 흥분상태가 계속되어 잠시 시각작용이 남는 경우가 있다. 이러한 현상을 잔상 효과(Persistence of Vision)라고 한다. 잔상 효과와 비슷한 현상이 사용자가 영화, 음악 등을 평가할 때에도 나타날 수 있다. 예를 들어 어떤 사용자가 재미없는 영화 A를 보고 난 후 영화 B를 보았을 때의 평점과 재미있는 영화 C를 보고 난 후 영화 B를 보았을 때, 사용자가 내린 영화 B에 대한 평점은 다를 것이다. 현재 어떤 영화의 평점을 매겨야 한다면 이 전에 본 영화가 지금 내려야 할 결정에 영향을 끼칠 수 있다. 잔상 효과처럼 사용자의 아이템에 대한 평가의 순서가 사용자의 결정에 끼치는 영향을 수학적으로 모델링하기 위해 본 연구팀은 강화 학습 기법을 협력적 여과 문제에 적용하였다.

4.2 문제 정형화(Problem Fomulation)

협력적 여과 문제를 강화 학습 기법으로 접근한 제안 알고리즘을 설명하기 위해서는 일단 협력적 여과 문제를 마르코프 결정 과정(Markov Decision Process, MDP)으로 정형화해야 한다[10]. 마르코프 결정 과정은 순차적 결정 문제에 대한 수학적 틀을 제공한다. 협력적 여과문제는 다음과 같이 MDP의 5개 구성 요소로 정의 될 수 있다.

- 상태 (State) S: 영화의 평점이 상태로 정의된다. 1부터 K까지 평점을 매길 수 있다면 상태의 크기 $|S|$는 K개이다. $s_t^{(i)} \in S$ 는 사용자 i가 t번째로 본 영화의 평점이 된다.
- 행동(Action) A: 앞서 정의했듯이 상태 s_t, s_{t+1}은 각각 t번째와 $t+1$번째로 본 영화의 평점이 된다. 사용자 i가 t번째 영화에 대해 평점 s_t를 내리고 그 다음 본 영화에 대해서 s_{t+1}의 평점을 매김으로써 상태가 전이하는데 이러한 상태 전이를 행동 $a_t^{(i)} \in A$로 정의한다.
- 전이 확률(Transition Probability): 협력적 여과에 대한 MDP는 결정적(Deterministic) MDP라고 가정한다. 즉 현재 상태 s_t에서 현재 행동 a_t를 취했을 때 다음 상태 s_{t+1}은 랜덤이 아니다.
- 할인 계수(Discount Factor) γ: $0 < \gamma < 1$로 설정한다.
- 보상(Reward) r: 사용자가 상태 s_t에서 행동 a_t를 취해서, 다음 상태 s_{t+1}로 전이했을 때, 사용자는 보상 또는 벌칙(Penalty)을 받는다. 만약 상태를 평점으로 정의한다면, 보상 함수 r은 수식 (3)과 같이 정의된다.

$$r(s_t, a_t) = s_{t+2} - predictor(i,j) \tag{3}$$

여기서 $predictor(i, j)$는 사용자 i의 영화 j에 대한 어떤 예측기의 예측 평점을 의미한다. 본 논문에서는 이를 기본 예측기(Base Predictor)로 부른다. 예를 들어 영화 j의 평균 평점이나 SVD 예측기의 예측 평점이 될 수 있다. 다음다음 상태 s_{t+2}는 각 사용자가 본 일련의 영화에 대한 평점 데이터를 통해 알 수 있다. 상태를 평점으로 정의할 때 왜 다음 다음 상태 s_{t+2}를 이용하는지에 대해서는 4.4절에서 자세히 설명한다.

- 정책(Policy) π: $S \rightarrow A$는 각 상태 S에서 행동 A로의 사상(Mapping)이다. 시작 상태 s_0와 정책 π가 주어지면, 가치 함수(Value Function) $V^{\pi}(s_0)$는 수식 (4)와 같이 정의된다. 가치 함수 $V^{\pi}(s_0)$는 상태 s에서 시작해서 정책 π를 따라 상태를 전이해 나갈 때의 할인된 보상들의 총합(Sum of Discounted Rewards)이다.

$$\begin{aligned} & V^{\pi}(s_0) \\ & = r(s_0, \pi(s_0)) + \gamma r(s_1, \pi(s_1)) + \gamma^2 r(s_2, \pi(s_2)) + \ldots \\ & = r(s_0, \pi(s_0)) + \gamma V^{\pi}(s_1) \end{aligned} \tag{4}$$

최적 가치 함수(Optimal Value Function) $V^*(s)$는 수식 (5)와 같이 정의된다.

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{5}$$

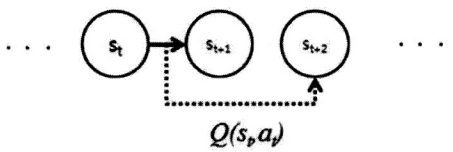
즉 $V^*(s)$는 가장 최적의 정책을 따를 때 얻을 수 있는 보상 총합이다.

4.3 훈련(Training)

4.2절과 같이 정의된 MDP를 기반으로 강화 학습 기법인 Q-learning을 이용해서 $Q(s, a)$를 학습하였다[11]. $Q(s, a)$는 일반적으로 다음 식과 같이 정의된다.

$$\begin{aligned} Q(s,a) &= r(s,a) + \gamma V^*(s') \\ &= r(s,a) + \gamma \max_{a'} Q(s',a') \end{aligned} \tag{6}$$

$Q(s, a)$값은 상태 s에서 행동 a를 취했을 때 얻을 수 있는 추정된 총 보상의 합(Estimated Total Sum of Rewards)이다. 본 연구에서 정형화한 MDP에서는 $Q(s_t, a_t)$값은 이전 평점 s_t에서 행동 a_t을 취해 다음 평점을 s_{t+1}로 매겼을 때, 기본이 되는 예측기가 상태 s_{t+2}에 내린 예측 평점보다 얼마만큼 높게 평점을 매겼느냐 또는 낮게 평가를 내렸느냐를 나타낸다. 이를 (그림 1)에 나타내었다.



(그림 1) $Q(s_t, a_t)$값의 정의

Q-learning에서 $Q(s_t, a_t)$는 s_t에서 s_{t+1}로의 전이가 s_{t+2}의 결정에 미치는 영향을 학습한다. Q-learning의 업데이트 규칙은 다음과 같다.

$$\Delta Q(s,a) = \alpha [r(s,a) + \max_{a'} Q(s',a') - Q(s,a)] \tag{7}$$

$Q(s_t, a_t)$는 예전의 총 보상의 합의 추정 값이고, $r(s,a) + \max_{a'} Q(s',a')$는 새로운 총 보상의 합의 추정

값을 의미한다. 상태 s에서 취할 수 있는 행동의 수는 $|S|$이므로 $Q(s_t, a_t)$는 $|S| \times |S|$ 크기의 테이블이 된다. 학습을 위해 각 사용자의 평점 데이터가 사용되는데 이를 위해 평점 행렬 $V \in R^{N\times M}$를 다음과 같이 구성한다.

$$s_1^{(1)} \xrightarrow{a_1^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}} s_3^{(1)} \xrightarrow{a_3^{(1)}} \cdots \xrightarrow{a_{T_1-1}^{(1)}} s_{T_1}^{(1)}$$
$$s_1^{(2)} \xrightarrow{a_1^{(2)}} s_2^{(1)} \xrightarrow{a_2^{(2)}} s_3^{(1)} \xrightarrow{a_3^{(2)}} \cdots \xrightarrow{a_{T_2-1}^{(2)}} s_{T_2}^{(2)}$$
$$\cdots$$
$$s_1^{(m)} \xrightarrow{a_1^{(m)}} s_2^{(1)} \xrightarrow{a_2^{(m)}} s_3^{(1)} \xrightarrow{a_3^{(m)}} \cdots \xrightarrow{a_{T_m-1}^{(m)}} s_{T_m}^{(m)}$$

(그림 2) 사용자 평점 시퀀스 구성

$s_j^{(i)}$는 사용자 i가 j번째 본 영화에 대해 내린 평점이다. $a_j^{(i)}$는 사용자 i가 j번째 영화를 볼 때의 행동으로 정의된다. 그리고 T_i는 사용자 i가 평점을 매긴 영화의 개수이다. 위의 훈련 데이터는, 평점 행렬 $X \in R^{N\times M}$에 추가적으로 영화에 대한 평점이 등록된 시간 데이터가 있으면 쉽게 구성할 수 있다.

각 사용자의 일련의 평점 시퀀스는 하나의 에피소드(Episode)로 볼 수 있다. 이 일련이 에피소드들이 Q-learning의 훈련 데이터가 된다. <표 1>과 같은 (사용자, 영화, 평점, 시간 순서) 항목 데이터에 대해 구성할 수 있는 에피소드들은 (그림 3)과 같다.

〈표 1〉 (사용자, 영화, 평점, 시간 순서) 데이터의 예

	movie 1	movie 2	movie 3	movie 4	movie 5
user 1	(2,1st)	missing	(5,2nd)	missing	(2,3rd)
user 2	(1,1st)	(1,2nd)	missing	(3,3rd)	missing
user 3	(5,1st)	missing	missing	(3,2nd)	(5,3rd)
user 4	(1,1st)	missing	missing	missing	(4,2nd)
user 5	(2,1st)	(4,2nd)	(5,3rd)	(5,4th)	missing

2→5→2
1→1→3
5→3→5
1→4
2→4→5→5

(그림 3) 〈표 1〉에서 구성한 에피소드 데이터

이렇게 훈련 데이터 집합이 완성되면 Q-learning 알고리즘을 통해 $Q(s, a)$를 학습할 수 있다. 전체적인 알고리즘을 아래와 같이 기술하였다. Train 알고리즘에서 정해주어야 두 개의 매개변수가 있는데, γ는 MDP의 5가지 구성 요소 중에 하나인 할인 계수이고 a는 학습률이다. 실험을 통해 적당한 값을 정해주어야 한다. 5.2절에서 이에 대한 실험을 다룬다.

Algorithm Train(α,γ)
Output: $Q(s,a)$

1: Initialize $Q(s,a)$
2: Convert $V \in R^{N\times M}$ to training episodes set
3: For each user i = 1 : N
4: For each movie j = 1 : T_i seen by i
5: $r(s_j^{(i)},a_j^{(i)}) = s_{j+2}^{(i)} - predict(i,j)$
6: $Q(s,a) = Q(s,a) + \Delta Q(s,a)$

4.4 예측(Prediction)

$Q(s, a)$의 학습이 끝난 후 테스트 데이터에 대해서 예측을 할 수 있다. 만약 상태를 평점으로 정의한다면, 사용자 i가 j번째 본 영화에 대한 예측 평점 $p_j^{(i)}$은 수식 (8)과 같이 계산된다.

$$p_j^{(i)} = predictor(i,j) + Q(s_{j-2}^{(i)}, a_{j-2}^{(i)}) \quad (8)$$

4.3절에서 언급한 것처럼 상태를 영화의 평점으로 정의한다면, $r(s_j^{(i)},a_j^{(i)})$를 계산하는데 다음 상태 $s_{j+1}^{(i)}$가 아니라 다음다음 상태 $s_{j+2}^{(i)}$를 이용하는 이유는 수식 (8)과 관련이 깊다. 만약 $s_{j+1}^{(i)}$을 사용한다면 예측 평점 $p_j^{(i)}$은 수식 (9)와 같이 된다.

$$p_j^{(i)} = predictor(i,j) + Q(s_{j-1}^{(i)}, a_{j-1}^{(i)}) \quad (9)$$

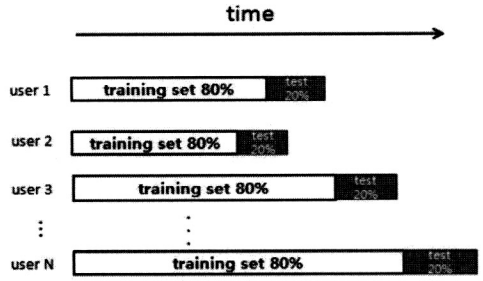
하지만 수식 (9)에서 $Q(s_{j-1}^{(i)}, a_{j-1}^{(i)})$는 예측하고자 하는 시점에서는 알 수 없다. 왜냐하면 $a_{j-1}^{(i)}$를 알기 위해서 현재 상태 $s_j^{(i)}$를 알아야 하는데, 상태를 영화의 평점으로 정의하였기 때문에 $s_j^{(i)}$는 바로 영화의 평점 즉 우리가 추측하고자 하는 답이 되어 이용할 수 없다.

5. 실 험

5.1 데이터 집합

데이터 집합으로는 MovieLens 데이터를 사용하였다[12]. Netflix 데이터 대신 MovieLens 데이터를 사용한 이유는 MovieLens 데이터는 평점의 시간 데이터가 초단위로 기록되어 있기 때문이다. Netflix 데이터에는 사용자가 평점을 매긴 시간이 일 단위로 되어 있기 때문에 하루에 다수의 영화의 평점을 매겼을 경우 그 순서를 알 수 없다. 평점을 매긴 순서는 본 연구에서 제안한 알고리즘의 중요한 조건 중에 하나이기 때문에 MovieLens 데이터를 사용하였다. MovieLens 데이터는 71,567명의 사용자와 10,681개의 영화에 대해서 10,000,054개의 평점들로 이루어져 있는 데이터를 사용하였다.

(그림 4)와 같이 각 사용자의 영화 평점 데이터를 시간 순으로 정렬한다. 이때 각 사용자마다 평점을 매긴 영화의 수는 사용자마다 다르다. 먼저 본 영화에 대한 평점 80%를 훈련 데이터 집합으로 사용하고 나머지 20%를 테스트 데이터 집합으로 사용하였다. MovieLens 데이터는 0.5점부터 0.5 간격으로 5점까지의 평점으로 이루어져 있다.



(그림 4) 데이터의 구성

5.2 실험 및 분석

평점을 상태로 본다면, 0.5에서 5점까지의 평점에 대해 총 10가지의 상태가 있다. 그러므로 제안 알고리즘으로 학습해야 할 $Q(s, a)$는 10 × 10 크기의 테이블이 된다. (그림 5)에 제안 알고리즘으로 학습된 $Q(s, a)$를 나타내었다. x, y, z는 각각 상태, 행동, Q-값을 의미한다. 왼쪽 상단 그래프는 수식 (3)의 *predictor*를 각 영화의 평균으로 했을 때의 $Q(s, a)$이고, 오른쪽 상단 그래프는 *predictor*를 SVD++로 했을 때의 Q(s, a)의 모습이다. 그리고 아래쪽 그림들은 각각 위의 그래프를 옆에서 본 모습이다.

(그림 5)를 보면, *predictor*를 영화 평균으로 했을 때, predictor를 SVD++로 했을 때보다 각 상태 변이 시 Q-값의 변화가 더 큼을 알 수 있다. $Q(s, a)$의 이 실험 결과는 t - 2번째 본 영화와 t - 1번째 영화의 평점이 t번째 평점을 매길 영화에 대해서 미치는 영향을 정량적으로 보여준다. 예를 들어, (그림 5)의 왼쪽 그래프에서 보듯이 $Q(s, a)$는 사용자가 t - 2번째 본 영화에 대해서 평점 5점을 매기고 다시 t - 1번째 본 영화에 대해서 평점 5점을 매겼을 때, t번째 본 영화에 대해서 그 영화에 대한 평균 평점보다 약 0.6점 가량 더 높은 평점을 매기는 경향이 있다는 것을 알려 준다.

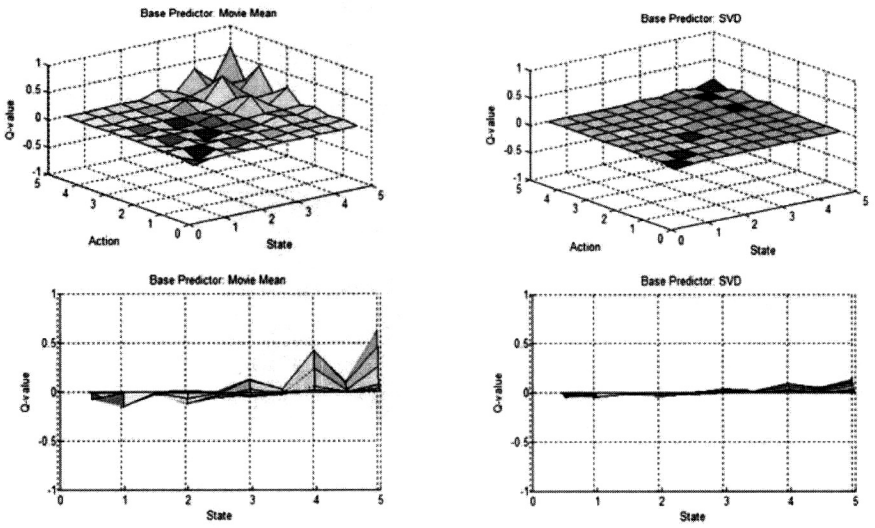
다음 실험으로 각 기본 예측기에 대해 제안 알고리즘을 적용하였을 때 성능 향상을 비교하여 보았다. 5.1절의 설명에 따라 분리한 테스트 데이터 집합들에 대해 기본 예측기(영화평균, SVD++)와 기본 예측기 + 제안 알고리즘의 RMSE 성능을 비교해보았다. SVD++ 알고리즘 특성의 차원은 10으로 설정하였다. 실험에 사용된 모든 알고리즘은 C로 구현하였고 모든 실험은 인텔 쿼드코어 2.4Ghz CPU, RAM 8GB PC에서 실험하였다.

제안 알고리즘의 경우 10초 이내에 학습, 예측, 평가가 완료되었다. 제안 알고리즘은 정해주어야 할 두 가지 매개 변수가 있다. Q-learning의 학습률 a와 할인 계수 γ이다. 최적의 매개 변수를 찾기 위해, 수식 (3)의 *predictor*(i, j)를 영화 j의 평균 평점으로 정하고 a와 γ를 변화시켜 가면서 RMSE를 측정하였다. 기본 예측기를 영화평균으로 할 경우 a = 0.5, γ = 0.000003에서 가장 좋은 성능을 보여주었고, 기본 예측기를 SVD++으로 할 경우 a = 0.65, γ = 0.000006에서 가장 좋은 성능을 보였다. 이 매개 변수를 가지고 성능 비교를 한 결과를 <표 2>에 나타내었다.

〈표 2〉 제안 알고리즘의 RMSE 향상

Algorithm	Base	Proposed	Improve
Movie Means	0.9381	0.9109	0.0272
SVD++	0.8000	0.7954	0.0046

첫 번째 실험인 기본 예측기를 영화 평균으로 할 경우 0.0272의 RMSE 성능 향상이 있었고, 두 번째 실험인 본 예측기를 SVD++으로 할 경우 0.0046의 RMSE 성능 향상이 있었다. 첫 번째 실험의 경우, 영화 평균에 비해 큰 폭의 성능 향상을 보였기 때문에 제안 알고리즘이 아이템에 대한 평가의 순서가 사용자의 결정에 끼치는 영향을 의미 있게 학습해 냈다고 볼 수 있다. 또 두 번째 실험의 경우 현재 협력적 여과 분야에서 단일 알고리즘 중 가장 효율적이라고 알려져 있는 SVD++의 성능을 더욱 올려놓았다. 이는 제안 알고리즘이 SVD++이 학습할 수 없는 평점 데이터의 순차적인 연관 관계를 추가적으로 학습할 수 있게 해준다는 것을 의미한다. Netflix 대회에서 가장 각광받는 예측기의 앙상블 기법에서도, 성능이 좋은 단일 알고리즘을 앙상블 예측기에 하나씩 추가할 때 약 0.0010 정도의 향상도 힘들기 때문에, 0.0037 정도의 성능 향상은 의미가 있다고 할 수 있다.



(그림 5) 실험 결과

6. 결론 및 향후과제

지금까지 영화 데이터의 협력적 여과 문제를 강화 학습으로 접근하여 해결하는 방식을 제시하였다. 이를 위해 협력적 여과문제를 마르코프 결정 과정으로 사용자의 아이템에 대한 평가의 순서가 사용자의 결정에 끼치는 영향을 수학적으로 모델링하였고, 강화 학습의 가장 대표적인 알고리즘인 Q-learning을 사용해서 평가의 순서의 연관 관계를 학습하였다. 실험을 통해서 실제로 평가의 순서가 평가에 미치는 영향이 있다는 것을 증명하였다.

아직 본 연구에서 진행시키지 못한 몇 가지 일이 남아 있다. 협력적 여과를 마르코프 결정 과정으로 모델링할 때, 상태를 평점뿐만 아니라 다른 것들로 정의할 수 있다. 예를 들어, 평점을 매기고자 하는 영화의 장르나, 영화를 감상한 계절, 영화에 달린 태그들 등이다. 상태를 평점으로 정의했을 경우 4.4절에서 언급한 문제로 인해 s_t를 예측할 때 s_{t-2}에서 s_{t-1}로의 상태 전이를 이용하지만, 위의 예시된 정보를 상태로 정의할 경우 s_t를 예측할 때 바로 s_{t-1}에서 s_t로의 상태 전이의 정보를 이용할 수 있다. 또한 위의 정보를 상태로 정의했을 때, 액션 영화를 본 후 멜로 영화를 봤을 때의 예와 같은 경우 평가에 미치는 영향 등을 학습해 낼 수 있게 된다.

또한 여러 예측기를 앙상블하여 더 좋은 예측기를 찾고자 할 때 제안 알고리즘을 후보 예측기로 사용할 수 있다. 앙상블 예측기가 그것을 구성하는 각각의 단일 예측기보다 정확도가 더 높기 위해서는 각 예측기가 임의 추측보다 정확하며 입력 공간의 다른 부분에서 에러를 가져야 한다[13]. 다시 말하면 각각의 예측기가 정확하며, 또한 각 예측기마다 볼 수 있는 데이터의 양상이 다양하여야 한다. 제안 알고리즘은 기존 여타의 단일 알고리즘이 고려하지 않는 평점의 순서와 최종 평점과의 관계를 학습해 내기 때문에 다른 예측기와의 차별성이 있으며, 따라서 앙상블의 후보 예측기로 제안 알고리즘을 사용할 경우 앙상블 예측기의 정확도를 더욱 높일 수 있을 것으로 기대된다.

본 연구에서 추천 시스템의 협력적 여과 분야에서 처음으로 강화 학습 기법을 적용하였다. 비록 협력적 여과가 아닌 다른 추천 시스템에 대한 강화 학습의 적용 예[14]가 있지만 이는 본 연구와 아주 간접적인 연관만 있을 따름이다. 본 연구가 협력적 여과 및 여타 추천 시스템의 연구에 많은 영감을 주길 바란다.

참 고 문 헌

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

[2] B. M. Sarwar and G. Karypis, J. A. Konstan, and J.T. Riedl, "Item-based collaborative filtering recommendation algorithms," Proceedings of the 10th international conference on World Wide Web, pp.285-295, 2001.

[3] Netflix Prize, http://www.netflixprize.com

[4] A. Paterek, "Improving regularized singular value decomposition for collaborative Filtering", KDD-Cup and Workshop, ACM press, 2007.

[5] R, Salakhutdinov, A. Mnih and G. Hinton, "Restricted Boltzmann machines for collaborative Filtering", Proceedings of the 24th International Conference on Machine Learning, 2007.

[6] R. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights", IEEE International Conference on Data Mining, IEEE, 2007.

[7] G. Gorrell and B. Webb, "Generalized hebbian algorithm for incremental latent semantic analysis", Proceedings of Interspeech, 2006.

[8] B. Webb, "Netflix update: Try this at home", http://sifter.org/simon/journal/20061211.html, 2006.

[9] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model", Proceedings of the 14th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp.426-434, 2008.

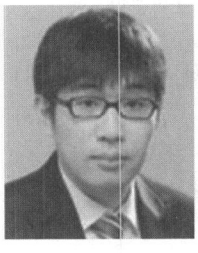
[10] R. Bellman, "A Markovian Decision Process", Journal of Mathematics and Mechanics 6, 1957.

[11] C. Watkins, "Learning from Delayed Rewards", PhD thesis, Cambridge University, Cambridge, England, 1989.

[12] MovieLens, http://www.movielens.umn.edu

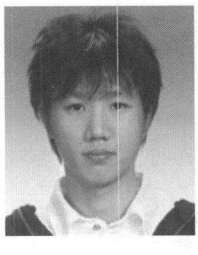
[13] L. Hansen and P. Salamon, "Neural Network Ensembles", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.12, pp.993-1001, 1990.

[14] G. Shani, D. Heckerman and R. Brafman, "An MDP-based recommender system", Journal of Machine Learning Research, Vol.6, No.2, pp.1265-1295, 2006.



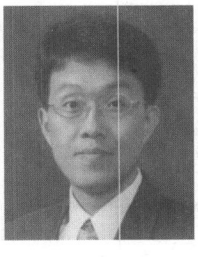
이 정 규

e-mail : sweaterr@gmail.com
2008년 서강대학교 수학과(학사)
2010년 서강대학교 컴퓨터공학과(석사)
2011년~현 재 (주)사이람 연구원
관심분야 : 기계학습, 추천 시스템, 네트워크 과학, 복잡계 이론



오 병 화

e-mail : byonghwaoh@gmail.com
2007년 서강대학교 컴퓨터학과(학사)
2009년 서강대학교 컴퓨터공학과(석사)
2009년~현 재 서강대학교 컴퓨터공학과 재학(박사)
관심분야 : 기계학습, 추천 시스템, 진화 알고리즘, 네트워크 과학, 상황 인식



양 지 훈

e-mail : yangjh@sogang.ac.kr
1987년 서강대학교 전자계산학과(학사)
1989년 ISU Department of Computer Science(석사)
1999년 ISU Department of Computer Science(박사)
1999년~2000년 HRL Laboratories, LLC, Malibu, CA 연구원
2000년~2002년 SRA International, Inco, Fairfax, VA 연구원
2002년~현 재 서강대학교 컴퓨터공학과 교수
관심분야 : 기계학습, 바이오인포메틱스 등