

碩 士 學 位 論 文

SAMC 알고리즘을 이용한
베이지안 개인화 순위 방법 개선

高麗大學校 大學院

應用統計學科

金 民 淨

2021年 2月

全 秀 榮 教 授 指 導

碩 士 學 位 論 文

SAMC 알고리즘을 이용한
베이지안 개인화 순위 방법 개선

이 論 文 을 統 計 學 碩 士 學 位 論 文 으 로
提 出 함 .

2021年 2月

高 麗 大 學 校 大 學 院

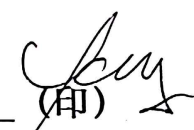
應 用 統 計 學 科


金 民 淨 


金 民 淨의 統計學 碩士學位論文

審査를 完了함.

2021年 2月

委員長 정승오 (印) 

委 員 김경수 (印) 

委 員 진서훈 (印) 

요 약 문

개인화 추천 알고리즘은 개인의 클릭, 검색, 구매 이력을 기반으로 맞춤형 상품을 추천해주는 시스템이다. 과거 기업이 고객의 신상 정보만을 수집하였지만, 최근에는 검색, 구매 이력과 같이 일상 정보까지 수집하면서 취향을 알기 쉬워져 개인화 추천이 점점 더 널리 쓰이고 있다. 평점 등 구체적인 점수를 매긴 명시적 데이터(explicit data)는 편향이 있거나 존재하지 않는 경우가 많아 조회 여부, 구매 여부처럼 명시적으로 점수화하지 못하는 암시적 데이터(implicit data)를 사용하여 추천 알고리즘을 개발하려는 연구가 진행되고 있다.

베이지안 개인화 순위(Bayesian personalized ranking, BPR; Rendle et al., 2009)는 사용자가 과거 행동 자료로부터 선호도가 있다고 해석할 수 있는 자료인 암시적 데이터를 분석하는 추천 알고리즘 통계 기법이다. 기존의 베이지안 개인화 순위 알고리즘에 이용된 확률적 경사 하강법(stochastic gradient descent)은 단순하고 구현이 쉽지만, 방향에 따라 기울기가 달라지는 비등방성(anisotropy) 함수에서 탐색 경로가 비효율적인 문제점이 존재한다.

따라서 기존의 BPR 알고리즘의 문제점을 극복하기 위해 본 연구는 확률적 근사 몬테카를로(stochastic approximation Monte Carlo, SAMC; Liang et al., 2007) 알고리즘을 이용하여 BPR 사후분포의 최적화를 다루고자 한다. SAMC 알고리즘은 표본공간에서 자기조절 능력을 갖추고 있어 국소 트랩의 문제점을 근본적으로 해결된다. 실증 자료 분석을 통해 본 연구에서 제안하는 SAMC 알고리즘을 이용한 BPR 알고리즘은 기존의 추천 알고리즘과 비교하여 우수한 결과를 보여준다.

핵심어: 개인화 추천 알고리즘, Implicit data, Bayesian personalized ranking, Stochastic approximation Monte Carlo

목 차

요 약 문.....	
목 차.....	
표 목 차.....	
제 1 장 서 론.....	1
제 2 장 행렬분해 알고리즘.....	4
2.1 Matrix Factorization (MF).....	4
2.2 Singular Value Decomposition (SVD).....	5
2.3 Non-Negative Matrix Factorization (NMF).....	6
제 3 장 베이저안 개인화 순위.....	7
3.1 Bayesian Personalized Ranking (BPR).....	7
3.2 Weighted Bayesian Personalized Ranking (WBPR).....	12
제 4 장 SAMC 알고리즘을 이용한 BPR 분석.....	13
4.1 SAMC 알고리즘.....	13
4.2 SAMC-BPR 알고리즘.....	15
제 5 장 실증 자료 분석.....	18
5.1 성능평가 방법.....	18
5.2 실증 자료 분석.....	20
제 6 장 결 론.....	31
참 고 문 헌.....	33

표 목차

<표 1> 분석에서 사용되는 실제 데이터에 대한 정보.....	20
<표 2> $k = 10$ 일 때 추천 알고리즘 결과 비교.....	21
<표 3> $k = 10$ 일 때 추천 알고리즘 결과 비교.....	24
<표 4> $k = 20$ 일 때 추천 알고리즘 결과 비교.....	25
<표 5> $k = 30$ 일 때 추천 알고리즘 결과 비교.....	26
<표 6> $k = 20$ 일 때 추천 알고리즘 결과 비교.....	27
<표 7> $k = 20$ 일 때 추천 알고리즘 결과 비교.....	29

그림 목차

<그림 1> MF의 구조.....	4
<그림 2> SVD의 구조.....	6
<그림 3> 암시적 피드백 데이터 형태.....	7
<그림 4> $k=10$ 일 때 추천 알고리즘 결과 비교.....	21
<그림 5> k 에 따른 AUC.....	22
<그림 6> $k=10$ 일 때 추천 상품 개수에 따른 nDCG.....	23
<그림 7> $k=20$ 일 때 추천 상품 개수에 따른 nDCG.....	24
<그림 8> $k=30$ 일 때 추천 상품 개수에 따른 nDCG.....	25
<그림 9> $k=20$ 일 때 추천 알고리즘 결과 비교.....	27
<그림 10> $k=20$ 일 때 추천 알고리즘 결과 비교.....	29

제 1 장 서 론

스마트폰의 대중화로 사용자들이 생성하는 빅데이터 및 이를 처리하는 기술이 빠르게 발전하고 있고, 사람들은 정보의 홍수 속에서 단순히 많은 것보다는 양질의 정보를 원한다. 이에 정보를 제공하는 온라인 비즈니스 업계에서는 ‘개인화(personalization)’, 즉, 개인의 특성과 기호에 맞는 맞춤형 서비스에 주목하고 있다. 실제로 쇼핑몰, 음악 스트리밍 서비스 등 여러 업종에서 개인화 추천 서비스가 구현되고 있다. 전 세계적으로 개인화 추천 서비스로 유명한 사례는 무수히 많다. 대표적으로 유튜브는 매분 500시간 이상의 새로운 동영상의 업로드되며, 매일 3,000만 명이 방문해 10억 시간 이상 시청한다. 유튜브의 핵심은 맞춤형 편성을 가능하게 한 개인화 추천 알고리즘이다. 유튜브에 들어가서 영상 하나를 재생할 때 ‘다음 동영상’으로 표시되는 영상 목록은 모두 알고리즘으로 추천된 콘텐츠이다. 알고리즘은 일종의 규칙 모음이다. 현재 재생 중인 영상과 비슷한 영상이나 함께 재생된 적이 많은 영상을 모두 올려 유튜브 체류 시간을 늘리는 전략이다. 구글은 이용자들이 추천 목록의 하단까지 내려가 클릭한 영상에는 가중치를 높이는 식의 변화를 이어가며 사용자의 관여도와 만족도를 높이고 있다.

현대는 정보 과잉의 시대이다. 정보가 많을수록 무언가를 결정하기가 힘들어진다. 온라인에서 접할 수 있는 콘텐츠가 기하급수적으로 늘어나면서 이를 추천해주는 서비스들도 생겨나고 있다. 이런 서비스들은 모두 추천 알고리즘을 통해 제공한다. 추천 알고리즘은 ‘특정 사용자가 특정 상품에 대한 선호도를 예상하거나 특정 사용자가 관심을 가질 만한 N개의 상품 목록을 알아내기 위해 사용되는 일련의 개인화 정보 필터링 기법’으로 정의된다. 이 기법은 사용자의 구매 혹은 사용기록에 기반을 두어 사용자의 상품 선호도를 함축하는 프로파일을 생성하고 얻어낸 선호도 프로파일을 이용해 해당 사용자가 구매하지 않은 상품에 대한 선호도를 예측, 평가해 최종 추천 상품 목록을 선정하는 과정으로 구성되어 있다. 이 과정에서 어떠한 방식으로 정보를 처리하는지와 어떠한 기반 정보를 이용하는지에 따라 추천 알고리즘을 크게 콘텐츠 기반 필터링(content-based filtering)과 협업 필터링(collaborative filtering)으로 나누어진다.

콘텐츠 기반 필터링은 콘텐츠 정보만을 가지고 다른 콘텐츠를 추천하는 방식으로 콘텐츠가 가지고 있는 고유의 특징을 분석하여 추천한다. 콘텐츠 기반 필터링의 장점은 콘텐츠 자체를 분석하는 것이기 때문에 초기에 사용자의 데이터가 적더라도 추천할 수 있다. 다만 콘텐츠만을 분석하기 때문에 사용자의 세부 성향을 파악하기가 어렵다는 단점이 있다. 협업 필터링은 메모리 기반 협업 필터링(memory-based collaborative filtering)과 모델 기반 협업 필터링(model-based collaborative filtering)으로 나뉜다. 메

모리 기반 협업 필터링은 비슷한 사람을 찾는 사용자 기반 협업 필터링(user-based collaborative filtering)과 비슷한 상품을 찾는 아이템 기반 협업 필터링(item-based collaborative filtering)이 있다. 모델 기반 협업 필터링은 지도 학습 및 비지도 학습으로 모델을 학습하여 기존 데이터가 갖는 규칙성에 따라 예측하는 방법이다. 이 방법은 기계학습을 적용한 필터링 방식이다 보니 메모리 기반 협업 필터링보다 성능이 더 우수하다. 그중에서 적은 데이터에서도 잘 작동하며 사용자에게 더 많은 상품을 추천할 수 있고 쉽게 확장할 수 있는 행렬분해(matrix factorization)가 가장 널리 사용된다.

강부식 외 2인(2008)에 따르면 추천 알고리즘의 목적은 어떤 사용자 행동이나 상품에 대한 정보로부터 사용자가 선호할 만한 관련 상품을 추천하여 알고리즘의 성능을 향상시키는 것이다. 하지만 추천 알고리즘을 적용하기 어려운 이유는 데이터가 매우 희소하다는 점이다. 이 때문에 데이터가 충분히 쌓이기 전까지 신규 사용자나 새로운 상품에 대한 추천이 어려운 콜드 스타트(cold start) 문제가 종종 발생한다. 이러한 문제를 총칭하여 데이터 희소성(data sparsity)이라고 한다. 그러므로 추천 알고리즘을 구축할 때 사용자가 선호하는 데이터를 얻는 것은 매우 중요한데, 수집하는 방법은 크게 두 가지로 나뉜다. 첫 번째는 사용자에게 직접 선호도나 관심도를 물어 답변을 받는 명시적 데이터를 수집하는 방법이고 다른 한 가지는 사용자가 상품을 구매하거나 상품 정보를 열람하는 등 사용자의 선호가 분명하게 드러나지 않아 상품에 흥미가 있다고 해석할 수 있는 암시적 데이터를 모으는 방법이다. 두 방법의 장단점을 살펴보면 먼저 양적인 면에서는 암시적 데이터가 압도적으로 많다. 데이터양이 많을수록 통계적 방법을 사용하기 유리하며 이를 이용해 구축한 모델도 더 높은 성능을 기대할 수 있다. 반면 데이터의 정확성 관점에서 명시적 데이터는 사용자가 직접 자신의 의사를 표명한 것이므로 정확도가 높지만, 암시적 데이터는 사용자가 잘못 클릭한 경우에도 긍정 평가로 취급되는 방식이라 정확도가 떨어질 수 있다.

대부분의 사용자들은 상품에 대해 명확한 평가를 하지 않고 단지 상품을 클릭하고 본인 기준에 마음에 든다면 장바구니에 넣어뒀다가 이를 구매하는 정도이다. 상품에 대한 리뷰를 작성하거나 별점을 주는 사용자는 극소수에 가깝다. 즉, 명시적 데이터가 없어도 암시적 데이터는 존재할 수 있다. 이러한 암시적 데이터로부터 적절한 개인화 추천 알고리즘을 개발하는 것은 중요하다. 일반적으로 암시적 데이터의 주요 특징 중 하나는 부정적인 데이터가 관측되지 않는다. 따라서 관측되지 않은 데이터에 대해서 실제로 사용자가 관심이 없는 것인지 아니면 관심이 있는데 아직 모르는 것인지 구분되지 않는다. 이 두 가지 경우 모두 값을 0으로 표기함으로써 암시적 데이터에서 관측된 데이터는 1(선호), 관측되지 않은 데이터는 0(비선호)으로 결측값을 부정 평가로 처리하여 모델링한다.

암시적 데이터를 이용하는 개인화 추천 알고리즘 중에서 베이지안 개인화 순위 (Bayesian personalized ranking, BPR; Rendle et al., 2009)는 기존의 추천시스템 기법과 비교해 우수한 성능을 보인다. 그런데 이 알고리즘에 이용된 확률적 경사 하강법 (stochastic gradient descent)은 탐색 경로가 비효율적이다. 본 연구는 이와 같은 문제점을 극복하기 위해 확률적 근사 몬테카를로(stochastic approximation Monte Carlo, SAMC; Liang et al., 2007) 알고리즘을 이용한 개인화 추천 알고리즘을 제시하려 한다. SAMC-BPR 알고리즘의 성능을 평가하기 위해 행렬분해에서 대표적인 알고리즘인 특잇값 분해(singular value decomposition, SVD), 비음수 행렬분해(non-negative matrix factorization, NMF) 그리고 BPR 알고리즘과 더불어 기존 알고리즘에서 표본 추출 방법을 달리한 가중 베이지안 개인화 순위(weighted Bayesian personalized ranking, WBPR) 알고리즘과 비교하고자 한다.

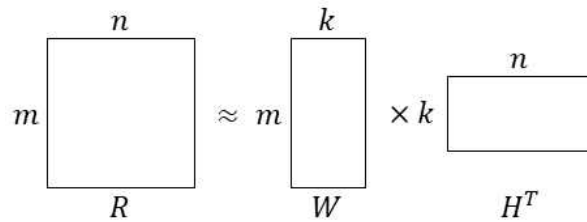
본 논문은 6개의 장으로 구성된다. 2장에서는 행렬분해 알고리즘과 MF, SVD, NMF, 3장에서는 Rendle et al.(2009)에 의해 제안된 베이지안 개인화 순위 알고리즘을 소개한다. 4장에서는 Liang et al.(2007)에 의해 제안된 SAMC 알고리즘과 이를 이용하여 BPR 알고리즘을 개선한 개인화 추천에 대해 알아본다. 5장에서는 실증 자료를 바탕으로 SAMC-BPR 알고리즘 성능을 행렬분해 추천 알고리즘들과 비교한다.

제 2 장 행렬분해 알고리즘

행렬분해(matrix factorization)는 모델 기반 협업 필터링 알고리즘 중 하나로, 데이터 희소성 문제에서 좋은 성능을 보이며 상품 추천의 정확성을 개선한 연구들은 대부분 행렬분해 모델을 기반으로 한다. 이 장에서는 MF 모델과 SVD, NMF 알고리즘을 소개한다.

2.1 Matrix Factorization (MF)

행렬분해(matrix factorization, MF)의 factorization은 인수분해를 의미한다. MF는 행렬을 분해하는 모든 기법의 상위개념이며 사용자-상품 행렬을 2개 또는 3개 이상의 행렬로 분해하는 방법으로 <그림 1>과 같다.



<그림 1> MF의 구조

행렬분해 모델은 식 (1)과 같이 표현할 수 있으며 m 명의 사용자, n 개의 상품을 나타내는 크기 $m \times n$ 인 행렬 R 을 크기가 $m \times k$ 인 행렬 W 와 $k \times n$ 인 행렬 H^T 로 분해하여 모든 사용자와 상품에 대해 오차 함수를 최소화하는 행렬을 찾는 방법이다.

$$R \approx WH^T \quad (1)$$

분해된 행렬 W 와 H^T 는 각각 사용자와 상품의 특성 벡터로 구성되어 있다. 사용자 특성 행렬 W 는 각 행이 모든 사용자 u 의 특성 벡터 w_u 로 이루어져 있고, 상품 특성 행렬 H^T 는 각 열이 모든 상품 i 의 특성 벡터 h_i 로 이루어져 있다. 두 개의 특성 벡터의 내적을 통해 사용자 u 의 상품 i 에 대한 선호도는 식 (2)와 같이 추정된다 (김상화 외 3인, 2011).

$$\hat{r}_{ui} = w_u^T h_i = \sum_{k=1}^K w_{uk} \cdot h_{ik} \quad (2)$$

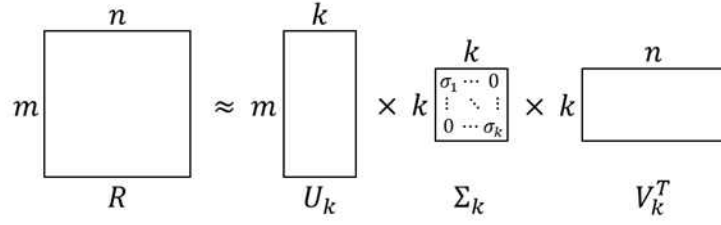
행렬분해의 장점은 추가 정보의 통합을 허용한다. 명시적 피드백을 사용할 수 없는 경우 구매 내역, 검색 기록, 검색 패턴 또는 마우스 이동 등 사용자 행동을 관찰하여 의견을 간접적으로 반영하는 암시적 피드백을 통한 사용자 환경 설정을 유추할 수 있다. 또한, 차원축소가 가능하며 이로 인해 이상치가 나오더라도 축소하면서 완화된다.

2.2 Singular Value Decomposition (SVD)

특잇값 분해(singular value decomposition, SVD) 알고리즘은 MF 모델에서 가장 일반적인 방법 중 하나로 잠재 변수를 추출해서 차원을 축소한다. SVD는 주어진 행렬 R 에 대해서 식 (3)과 같이 세 개의 U , Σ , V^T 행렬로 분해할 수 있다.

$$R = U \Sigma V^T \quad (3)$$

여기에서 R 은 $m \times n$ 크기의 선호도 행렬이며 U 는 역행렬이 대칭행렬인 $m \times m$ 크기의 행렬이다. Σ 는 비대각 성분이 0인 $m \times n$ 크기의 행렬이다. V^T 는 역행렬이 대칭행렬인 $n \times n$ 크기의 행렬이다. 비대각 성분이 0인 행렬 Σ 의 대각 성분을 특잇값이라 하며 이를 활용하여 특잇값 분해를 한다. 다시 <그림 2>와 같이 n 차원의 사용자-상품 행렬 R 을 k 차원으로 차원을 축소하여 SVD를 구성하는 각 행렬을 나타낼 수 있다. U_k 는 행렬 U 에서 대표 특잇값인 k 개의 성분만 남긴 $m \times k$ 크기의 행렬이다. Σ_k 은 대푯값으로 쓰일 k 개의 특잇값 성분만을 남긴 $k \times k$ 크기의 대각행렬이다. V_k^T 는 행렬 V 에서 k 개의 대표 특잇값 성분만 남긴 $k \times n$ 크기의 행렬이다. 이 세 개의 행렬을 곱하면 원래의 선호도 행렬 R 에 근사시킬 수 있다 (정성원, 2019).



<그림 2> SVD의 구조

사용자 u 가 피드백을 나타내지 않은 상품 i 에 대한 예측값은 식 (4)와 같다 (이현중 외 3인, 2011).

$$\hat{r}_{ui} = \bar{r}_u + u_{u,k} \cdot \Sigma_k \cdot v_{i,k}^T \quad (4)$$

여기에서 \bar{r}_u 는 사용자 u 가 피드백을 나타낸 상품에 대한 선호도 평균이다. $u_{u,k}$ 는 k 차원으로 축소된 행렬 U_k 의 u 번째 행, $v_{i,k}^T$ 는 k 차원으로 축소된 행렬 V_k^T 의 i 번째 열을 나타낸다.

2.3 Non-Negative Matrix Factorization (NMF)

비음수 행렬분해(non-negative matrix factorization, NMF) 알고리즘은 SVD 알고리즘과 유사하나 긍정적 선호도만을 반영한다 (정성원, 2019).

NMF 알고리즘은 MF의 구조인 식 (1)과 같으며 $m \times n$ 크기의 행렬 R 을 W 와 H 행렬로 분해한다. W 는 $m \times k$, H 는 $n \times k$ 이고 $k < n$ 인 음수 미포함 행렬이다. 일반적으로 NMF 알고리즘의 과정은 행렬 W 와 H 를 반복적으로 업데이트하여 행렬 R 과의 거리를 최소화하는 방향으로 근사한다.

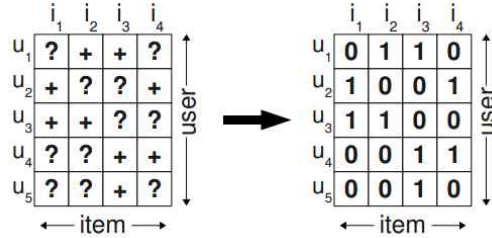
NMF 알고리즘은 비음수적으로 분해함으로써 독립적인 사용자들일지라도 이들 사이에 의미 관계를 발견할 수 있다. 그러나 이 알고리즘은 불안정한 값을 대상으로 결측치를 예측하는 방법이기 때문에 계산과정이 복잡하고 시간이 많이 소요되어 실시간에서의 동적인 추천이 어려운 단점이 있다.

제 3 장 베이지안 개인화 순위

이 장에서는 Rendle et al.(2009)에 의해 제안된 베이지안 개인화 순위(Bayesian personalized ranking, BPR)를 소개한다.

3.1 Bayesian Personalized Ranking (BPR)

개인화 순위는 순위가 등록된 상품 목록을 사용자에게 제공하는 것으로, 상품 추천이라고도 불린다. BPR 알고리즘은 개인별 상품 간의 선호도를 확률 모델링한 것이다. 이 방법은 상품 클릭, 구매 등과 같은 사용자 u 의 상품 i 에 관한 암시적 피드백인 개인화 점수 x_{ui} 를 고려한다. 모든 사용자의 집합을 U , 모든 상품의 집합을 I 라고 가정했을 때, $x_{ui} > 0$ 인 사용자-상품 쌍의 집합 $S := \{(u, i) \in U \times I : x_{ui} > 0\}$ 를 <그림 3>과 같이 표현할 수 있다.



<그림 3> 암시적 피드백 데이터 형태

행렬에서 사용자 u 가 암시적 피드백을 나타낸 긍정적 상품들(positive items)의 집합인 $I_u^+ := \{i \in I : x_{ui} > 0\}$ 는 '1'로, 사용자 u 가 암시적 피드백을 나타내지 않은 부정적 상품들(negative items)의 집합인 $I_u^- := I - I_u^+$ 는 '0'으로 나타내고 훈련 데이터 D_S 는 식 (5)와 같이 정의한다 (김동우, 이은령, 2017).

$$D_S := \{(u, i, j) \in U \times I \times I : i \in I_u^+, j \in I_u^-\} \quad (5)$$

D_S 를 바탕으로 모든 상품 $i \in I$ 에 대해 개인 맞춤형 순위를 찾는 베이지안 공식은 식 (6)과 같이 사후 확률을 최대화하는 파라미터 θ 를 찾는 것이다. 이를 최대 사후 확률 추정(maximum a posteriori estimation)이라고 한다.

$$p(\theta | >_u) \propto p(>_u | \theta) p(\theta) \quad (6)$$

θ 는 개인화 순위를 결정하는 모델 파라미터 벡터이다. 사후 확률이란 어떤 정보가 고려된 파라미터에 대한 확률이다. 여기에서는 사용자의 선호 정보($>_u$)이다. 이에 대비되는 개념이 사전 확률이다. 이는 위 식의 $p(\theta)$ 로, 파라미터에 대한 사전 정보이다. 사전 확률은 이미 주어진 정보이므로 사후 확률과 달리 오직 파라미터에만 의존한다. 따라서 목표는 $>_u$ 가 주어졌을 때, 이를 최대한 잘 나타낼 수 있는 파라미터를 추정하는 것이다. 사후 확률은 베이즈 정리에 의해 우도 함수와 사전 확률의 곱으로 나타낼 수 있다.

우도 함수란 $>_u$ 에 대한 확률분포이다. 사용자는 각 상품과 모든 쌍에 대해 고유의 선호도가 있다고 가정한다. 즉, 사용자 u 가 상품 j 보다 상품 i 를 더 선호하는 경우 $i >_u j$ 라고 표기한다. $>_u$ 를 $(u, i, j) \in D_S$ 로 정의한다면 $i >_u j$ 와 $j >_u i$ 두 가지 경우 밖에 존재하지 않으므로 베르누이(Bernoulli) 분포를 따른다고 볼 수 있다. 이때 사용자들은 서로 독립적으로 행동하고 특정 사용자에게 대한 모든 상품 쌍 (i, j) 순서가 다른 모든 쌍 순서와 독립적이라고 가정한다. δ 는 지시 함수이며 사용자별 우도 함수는 식 (7)과 같이 나타낸다.

$$\begin{aligned} \prod_{u \in U} p(>_u | \theta) &= \prod_{(u, i, j) \in U \times I \times I} p(i >_u j | \theta)^{\delta((u, i, j) \in D_S)} \cdot (1 - p(i >_u j | \theta))^{\delta((u, i, j) \notin D_S)} \\ &= \prod_{(u, i, j) \in D_S} p(i >_u j | \theta) \end{aligned} \quad (7)$$

BPR 알고리즘은 상품의 쌍을 살펴본다. 사용자 u 가 상품 i 를 구매했고 상품 j 는 사지 않았다면 u 는 j 보다는 i 에 더 관심이 있을 가능성이 높다. 따라서 모델을 훈련시킬 때 (u, i) 에 대해 생성한 점수 \hat{x}_{ui} 는 (u, j) 에 대한 점수 \hat{x}_{uj} 보다 높아야 한다. 다시 말해, 원하는 점수 모델은 $\hat{x}_{ui} - \hat{x}_{uj} > 0$ 이다. 이 알고리즘을 훈련시키려면 세 요소로 이루어진 쌍인 (사용자, 긍정적 상품, 부정적 상품)이 필요하다. 이 (u, i, j) 에 대해 쌍별 점수의 차이는 $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$ 이다.

$$p(i >_u j | \Theta) := \sigma(\hat{x}_{uij}(\Theta)) = \sigma(\hat{x}_{ui} - \hat{x}_{uj}) \quad (8)$$

식 (8)에서 $\sigma(x) = 1/(1 + e^{-x})$ 는 시그모이드(sigmoid) 함수이고, $\hat{x}_{uij}(\Theta)$ 는 사용자 u , 상품 i, j 사이의 관계를 나타내는 모델 파라미터 벡터 Θ 임의의 실제 값 함수를 의미한다.

사전 확률은 파라미터 Θ 에 대한 확률분포를 가정하는 것이다. 특별한 사전 정보가 없어 식 (9)와 같이 평균이 0이고 분산-공분산 행렬이 Σ_Θ 인 정규분포를 사용한다.

$$p(\Theta) \sim N(0, \Sigma_\Theta) \quad (9)$$

다음으로, 알 수 없는 하이퍼 파라미터의 개수를 줄이기 위해 $\Sigma_\Theta = \lambda_\Theta I$ 라고 설정하고 λ_Θ 는 모델별 정규화 파라미터를 의미한다.

BPR 알고리즘을 MF에 적용하면 개인화 점수 \hat{x}_{ui} 을 예측하는 문제는 행렬 $X: U \times I$ 를 추정하는 문제로 볼 수 있다. 타겟 행렬 X 는 식 (10)과 같이 두 개의 낮은 랭크 행렬 $W: |U| \times k, H: |I| \times k$ 의 내적으로 근사된다.

$$\hat{X} := WH^T \quad (10)$$

k 는 차원/랭크의 근사값, W 의 각 행 w_u 는 사용자 u 를 설명하는 특성 벡터, H 의 각 행 h_i 는 상품 i 를 설명하는 특성 벡터이다.

각 상품 i 에 관한 사용자 u 의 개인화 점수 \hat{x}_{ui} 의 예측 공식은 식 (11)과 같다.

$$\hat{x}_{ui} = \langle w_u, h_i \rangle = \sum_{f=1}^k w_{uf} \cdot h_{if} \quad (11)$$

사용자 u 가 상품 i 에 대한 점수를 구하는 방식은 사용자의 상품들에 대한 숨겨진 선호도 w_u 와 그에 대응하는 상품들의 숨겨진 특성 h_i 에 의해 결정된다. MF는 사용자와 상품의 특성 벡터를 예측하는 것이므로 w_u 와 h_i 가 파라미터가 된다.

따라서 행렬분해 알고리즘에 적용하여 파라미터 $\Theta = (W, H)$ 를 추출하기 위한 로그 사후 확률분포는 식 (12)와 같다.

$$\begin{aligned}
BPR-OPT &:= \ln p(\Theta | >_u) \\
&\propto \ln \prod_{u \in U} p(>_u | \Theta) p(\Theta) \\
&= \ln \prod_{(u, i, j) \in D_S} \sigma(\hat{x}_{uij}) p(\Theta) \\
&= \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{uij}) + \ln p(\Theta) \\
&= \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2 \\
&= \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{ui} - \hat{x}_{uj}) - \lambda_\Theta \|w_u\|^2 - \lambda_\Theta \|h_i\|^2 - \lambda_\Theta \|h_j\|^2 \\
&= \sum_{(u, i, j) \in D_S} \ln \sigma(w_u h_i^T - w_u h_j^T) - \lambda_\Theta \|w_u\|^2 - \lambda_\Theta \|h_i\|^2 - \lambda_\Theta \|h_j\|^2 \\
&= \sum_{(u, i, j) \in D_S} \ln \frac{1}{1 + e^{-(w_u h_i^T - w_u h_j^T)}} - \lambda_\Theta \|w_u\|^2 - \lambda_\Theta \|h_i\|^2 - \lambda_\Theta \|h_j\|^2
\end{aligned} \tag{12}$$

로그 사후분포(BPR-OPT)에서의 최대사후확률(maximum a posteriori, MAP) 문제는 곡선아래면적(area under the curve, AUC)을 구하는 것과 유사하다. 사용자별 AUC는 식 (13)과 같이 정의된다.

$$\begin{aligned}
AUC(u) &:= \frac{1}{|I_u^+| |I - I_u^+|} \sum_{i \in I_u^+} \sum_{j \in |I - I_u^+|} \delta(\hat{x}_{uij} > 0) \\
&= \frac{1}{|U|} \sum_{u \in U} AUC(u) = \sum_{(u, i, j) \in D_S} z_u \delta(\hat{x}_{uij} > 0)
\end{aligned} \tag{13}$$

그러나 AUC 수식과 BPR-OPT는 정규화 상수 z_u 와 손실함수가 다르다. AUC는 미분할 수 없는 손실함수 $\delta(x > 0)$ 를 사용하지만, BPR-OPT는 미분 가능한 손실함수 $\ln \sigma(x)$ 를 사용한다.

개인 맞춤형 순위에 대한 최적화 기준을 최대화하기 위해 Rendle et al.(2009)은 부트스트랩(bootstrap)을 이용하여 확률적 경사 하강법(stochastic gradient descent, SGD)을 기반으로 한 학습 알고리즘을 제시하였다.

먼저, 모델 파라미터 관점에서 BPR-OPT의 기울기는 식 (14)와 같다.

$$\begin{aligned} \frac{\partial BPR-OPT}{\partial \Theta} &= \sum_{(u,i,j) \in D_S} \frac{\partial}{\partial \Theta} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} \frac{\partial}{\partial \Theta} \|\Theta\|^2 \\ &\propto \sum_{(u,i,j) \in D_S} \frac{-e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} - \lambda_{\Theta} \Theta \end{aligned} \quad (14)$$

이 경우, $(u, i, j) \in D_S$ 의 업데이트가 식 (15)와 같이 수행된다.

$$\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_{\Theta} \Theta \right) \quad (15)$$

모델 파라미터의 업데이트에는 학습률 α 와 정규화 상수 λ_{Θ} 를 사용한다. 훈련 데이터는 사용자-상품 쌍으로 이루어져 쌍의 순서가 중요하다. 따라서 연속적인 업데이트 단계에서 동일한 사용자-상품 쌍을 선택하는 확률을 낮추기 위해 부트스트랩 표본 추출 기법으로 u, i, j 를 선택하는 SGD 알고리즘 사용을 제안한다.

상품 추천시스템이란 특정 상품 목록에 대해 개인화된 순위를 매기는 문제를 의미한다. 기존의 추천 알고리즘 기법들에 비해 우수한 성능을 보여주는 것으로 알려져 있는 BPR 알고리즘은 베이지안 분석 기법에 의거하여 AUC 영역을 최대화하면서 개인화 순위를 최적화시키는 개인화 추천의 목표를 다루며 확률적 경사 하강법을 이용한 행렬 분해 알고리즘을 선보였다. BPR 알고리즘에 이용된 확률적 경사 하강법은 단순하고 구현도 쉽지만, 방향에 따라 기울기가 달라지는 함수인 비등방성(anisotropy) 함수에서 탐색 경로가 비효율적이다. 이를 극복하기 위해 본 연구는 확률적 근사 몬테카를로(stochastic approximation Monte Carlo, SAMC) 알고리즘을 이용하여 BPR 사후분포의 최적화를 다루고자 한다.

3.2 Weighted Bayesian Personalized Ranking (WBPR)

BPR 알고리즘은 사용자 u 가 상품 j 보다 상품 i 를 선호할 확률을 시그모이드 함수로 정의한다. 이 확률을 이용하여 사후 확률을 최대화시키고자 한다. 이 알고리즘은 확률적 경사 하강법을 사용하는데 훈련 데이터 D_S 는 사용자 u 가 암시적 피드백을 나타낸 상품 i 와 나타내지 않은 상품 j 의 삼중항(triplet)이므로 고려해야 할 쌍의 개수가 너무 많아 데이터를 균일하게 표본 추출하기에는 결과가 좋지 않을 수 있다. 따라서 식 (16)은 (u, i, j) 에서 j 를 달리 고르기 위해 식 (12)을 조정한 표본 추출 방식을 제안하고 있다 (Gautner et al., 2012).

$$\max \sum_{(u, i, j) \in D_S} w_u w_i w_j \ln \sigma(\hat{x}_{uij}) - \lambda \|\Theta\|^2 \quad (16)$$

이때 $w_u = 1/|I_u^+|$, $w_i = 1$ 로 정의한다. 즉, 관측한 상품이 더 많은 사용자는 적게 추출하고 모든 긍정적 상품은 균일하게 추출한다. 그리고 $w_j = \frac{1}{|U||I|} \sum_u I_{j \in I_u^+}$ 으로 정의하여 많은 사용자들이 관측하고 좋아하는 상품 위주로 표본을 추출한다.

제 4 장 SAMC 알고리즘을 이용한 BPR 분석

본 장에서는 Liang et al.(2007)에 의해 제안된 확률적 근사 몬테카를로(stochastic approximation Monte Carlo, SAMC) 알고리즘에 대한 소개와 이를 이용한 BPR 알고리즘에서의 추정을 설명한다.

4.1 SAMC 알고리즘

SAMC 알고리즘은 표본공간을 분할하여 국소 트랩(local trap) 문제를 해결한다. 먼저 식 (17)과 같은 형태의 분포로부터 표본을 추출하는 데에 관심이 있다고 하자.

$$f(x) = \frac{1}{Z} \psi(x), \quad x \in \chi \quad (17)$$

여기에서 Z 는 정규화 상수, χ 는 표본공간, $\psi(x)$ 는 비음(non-negative) 함수이다. 표본공간은 미리 정해진 상수인 u_1, \dots, u_{m-1} 에 따라 함수 $U(x)$ 가 m 개의 분리된 하위영역으로 분할된다. 즉, $\chi = \cup_{i=1}^m E_i$; $E_1 = \{x : U(x) \leq u_1\}$, $E_2 = \{x : u_1 < U(x) \leq u_2\}$, \dots , $E_{m-1} = \{x : u_{m-2} < U(x) \leq u_{m-1}\}$, $E_m = \{x : U(x) > u_{m-1}\}$ 과 같이 나타낼 수 있다. SAMC는 미리 정해진 빈도를 이용하여 각 영역에서 표본을 추출한다.

$i = 1, \dots, m$ 에 대해 $w_i = \int_{E_i} \psi(x) dx > 0$ 이다. $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ 를 $0 < \pi_i < 1$ 와

$\sum_{i=1}^m \pi_i = 1$ 을 만족하는 m 벡터인 각 부분 영역에서의 원래의 표본 함수라고 정의한다.

$i = 1, \dots, m$ 에 대해 $\theta_i = \log\left(\int_{E_i} \psi(x) dx / \pi_i\right) = \log(w_i / \pi_i)$, $\theta = (\theta_1, \theta_2, \dots, \theta_m)$, Θ 는 θ 의

공간이라 정의한다. $\psi(x)$ 의 일반적인 선택을 위해 $\Theta = R^m$ 이라고 한다.

$\theta^{(t)} = (\theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_m^{(t)})$ 는 t 번째 반복에서 얻어진 θ 의 추정치이다. 따라서 식 (17)을 식 (18)과 같이 다시 표현할 수 있다.

$$f_{\theta^{(t)}}(x) \propto \sum_{i=1}^m \frac{\psi(x)}{e^{\theta_i^{(t)}}} I(x \in E_i) \quad (18)$$

이득 계수(gain factor) $\{\gamma_t\}$ 는 어떤 $\tau \in (1, 2)$ 에 대해

$$(a) \sum_{t=1}^{\infty} \gamma_t = \infty, \quad (b) \sum_{t=1}^{\infty} \gamma_t^{\tau} < \infty \quad (19)$$

를 만족하는 양수이고 단조 증가하는 수열이라고 정의한다. 본 연구에서는 $t_0 > 1$ 인 어떤 정해진 값에 대해 $\gamma_t = \frac{t_0}{\max(t_0, t)}$, $t = 1, 2, \dots$ 을 사용한다. $\mathcal{J}(x)$ 는 표본 x 가 속해 있는 하위영역의 인덱스이다. 이와 같은 가정을 이용한 SAMC 알고리즘을 다음과 같이 정리한다.

SAMC 알고리즘:

[단계 1] (표본 추출) 목표(target)분포를 얻기 위해 한 번의

Metropolis-Hastings(MH) 업데이트에 의해 표본 $x^{(t+1)}$ 을 추출한다.

- (1) 표본 제안분포 $q(x^{(t)}, y)$ 에 따라 y 를 생성시킨다.
- (2) 다음의 비율을 계산한다.

$$r = e^{\theta_{\mathcal{H}(x^{(t)})}^{(t)} - \theta_{\mathcal{H}(y)}^{(t)}} \frac{\psi(y)q(y, x^{(t)})}{\psi(x^{(t)})q(x^{(t)}, y)}$$

- (3) 확률 $\min(1, r)$ 을 가지고 생성된 y 를 받아들인다.

만약 받아들이면 $x^{(t+1)} = y$ 이고, 그렇지 않다면 $x^{(t+1)} = x^{(t)}$ 이다.

[단계 2] (θ 업데이트) $\theta^* = \theta_t + \gamma_{t+1}(e_{t+1} - \pi)$ 라고 정의한다.

여기에서 $e_{t+1} = (e_{t+1,1}, \dots, e_{t+1,m})$ 이고, 만약 $x^{(t)} \in E_i$ 이면 $e_{t+1,i} = 1$, 그렇지 않으면 $e_{t+1,i} = 0$ 이다. 만약 $\theta^* \in \Theta$ 이면 $\theta_{t+1} = \theta^*$, 그렇지 않으면 $\theta_{t+1} = \theta^* + c^*$ 이다. 즉, y 가 추출된 영역에 낮은 가중치를 줘서 다른 영역에서 y 가 추출될 수 있도록 θ 를 업데이트한다. 여기에서 $c^* = (c^*, \dots, c^*)$ 이고 $\theta^* + c^* \in \Theta$ 를 만족하는 임의의 벡터이다.

4.2 SAMC-BPR 알고리즘

본 절에서는 Rendle et al.(2009)에 의해 제안된 베이지안 개인화 순위 방법에서의 추정 방법을 설명한 후, 김민정 외 3인(2020)의 방법을 개선한 SAMC-BPR 알고리즘을 제안한다.

행렬분해 알고리즘에 대한 모델 파라미터는 $\Theta = (W, H)$ 이다. 이 파라미터는 관찰되지 않은 사용자의 취향과 상품 속성들을 모델링하는 잠재 변수로 볼 수 있다. 사용자가 어떤 하나의 상품보다 다른 상품을 선호하는지를 추정하는 문제의 경우, 제시한 학습 알고리즘의 BPR-OPT 기준에 따라 모델 파라미터 Θ 를 업데이트 하기 위해 \hat{x}_{uij} 의 기울기를 사용한다. \hat{x}_{uij} 의 기울기는 식 (20)과 같다.

$$\frac{\partial}{\partial \theta} \hat{x}_{uij} = \begin{cases} (h_{if} - h_{jf}) & \text{if } \theta = w_{uf}, \\ w_{uf} & \text{if } \theta = h_{if}, \\ -w_{uf} & \text{if } \theta = h_{jf}, \\ 0 & \text{else} \end{cases} \quad (20)$$

세 개의 파라미터 w_u, h_i, h_j 에 대한 기울기를 개별적으로 유도하면 식 (21)과 같다.

$$\begin{aligned} \frac{\partial}{\partial w_u} \hat{x}_{uij} &= \frac{1}{1 + e^{(w_u h_i^T - w_u h_j^T)}} \cdot (h_i - h_j) - 2\lambda w_u \\ \frac{\partial}{\partial h_i} \hat{x}_{uij} &= \frac{1}{1 + e^{(w_u h_i^T - w_u h_j^T)}} \cdot (w_u) - 2\lambda h_i \\ \frac{\partial}{\partial h_j} \hat{x}_{uij} &= \frac{1}{1 + e^{(w_u h_i^T - w_u h_j^T)}} \cdot (-w_u) - 2\lambda h_j \end{aligned} \quad (21)$$

본 논문에서 개선된 모델 파라미터 θ 의 추정을 위한 표본공간 함수 $U(x)$ 는 곡선 아래 면적인 AUC를 이용하여 $U(x) = -AUC(x)$ 로 정의한다. 로그 사후분포인 BPR-OPT 추정을 위한 SAMC-BPR 알고리즘은 다음과 같다.

SAMC-BPR 알고리즘(김민정 외 3인, 2020):

[단계 1] (표본 추출) 한 번의 MH 업데이트에 의해 표본 $\theta^{(t+1)}$ 을 추출한다.

- (1) 식 (20) BPR 학습 알고리즘에 따른 표본 제안분포 $q(\theta^{(t)}, y)$ 에 따라 y 를 생성시킨다.
- (2) 다음의 비율을 계산한다.

$$r = e^{\Phi_{J(\theta^{(t)})}^{(y)} - \Phi_{J(y)}^{(t)}} \frac{\psi(y) q(y, \theta^{(t)})}{\psi(\theta^{(t)}) q(\theta^{(t)}, y)}$$

- (3) 확률 $\min(1, r)$ 을 가지고 생성된 y 를 받아들인다.

만약 받아들이면 $\theta^{(t+1)} = y$ 이고, 그렇지 않다면 $\theta^{(t+1)} = \theta^{(t)}$ 이다.

[단계 2] (Φ 업데이트) $\Phi^* = \Phi_t + \gamma_{t+1}(e_{t+1} - \pi)$ 라고 정의한다.

본 연구에서는 효율적인 SAMC 알고리즘 이용을 위해 다음과 같이 김민정 외 3인 (2020)의 방법을 수정한 방법을 제안한다.

수정된 SAMC-BPR 알고리즘:

표본공간의 부분공간인 E_1, \dots, E_m 은 AUC 값 $U(x)$ 에 따라 순서화되어 있다. 이때 $w(u)$ 를 $U(x)=u$ 를 가지는 표본 x 가 속하는 부분공간이라고 하자. 예로 $x \in E_i$ 일 때 $w(U(x))=i$ 가 된다. $\chi^{(t)}$ 를 t 번째 반복시행에서의 표본공간이라고 하자.

[단계 1] 표본공간을 $\chi^{(1)} = \cup_{i=1}^m E_i$ 로 시작한다. 즉, 모든 부분공간이 사용된다.

(1) (표본 추출) 한 번의 MH 업데이트에 의해 표본 $\theta^{(t+1)}$ 을 추출한다.

(2) (Φ 업데이트) $\Phi^* = \Phi_t + \gamma_{t+1}(e_{t+1} - \pi)$ 라고 정의한다.

[단계 2] 다음과 같이 표본공간을 반복적으로 줄인다.

$$\chi^{(t)} = \cup_{i=1}^{\omega(U_{\min})} E_i$$

여기에서 U_{\min} 은 시행할 때마다 그때까지 얻어진 $U(x)$ 의 최솟값이다. 따라서 시행할 때마다 U_{\min} 은 달라질 수 있다.

본 논문의 모든 실증 자료 분석에서 $m=5$, 그리고 반복의 총 횟수를 100,000번으로 지정하였다. 기존의 전 영역에서 표본 추출하는 것보다 훨씬 효율적이며 표본 추출 과정에서 발생하는 국소 트랩 문제를 해결하여 더 정확한 모수 추정과 성능을 기대한다.

제 5 장 실증 자료 분석

5.1 성능평가 방법

기존에 소셜 커머스(social commerce) 분야의 순위 기반 추천시스템에서는 ‘사용자가 실제로 살 것 같은 하나의 상품’을 잘 추천해주는 것이 중요했기 때문에 순위 기준의 평가 방법을 사용하지 않았다. 하지만 대부분의 검색과 추천시스템에서 다수의 사용자들은 1, 2페이지 또는 상위의 목록을 참조할 것이기 때문에 노출되는 상품의 순위를 잘 제공하는 것이 중요하다. 이처럼 본 연구에서는 추천 알고리즘의 성능평가를 위해 BPR 알고리즘에서 사용된 AUC와 테스트 데이터에 대한 추천 결과를 평가하는 방법 중에 가장 널리 알려진 순위 평가 방법인 nDCG에 대해 소개한다.

5.1.1 AUC (Area Under the Curve)

AUC는 ROC curve에서 곡선의 아랫부분 면적을 의미하며 추천 결과들이 랜덤으로 생성된 추천 결과에 비해 얼마나 잘 추천을 했는가 판단하는 데에 이용된다.

본 실증자료분석에서는 Rendle et al.(2009)에서와 같이 하나남기기 교차검증 (leave-one-out cross validation, LOOCV) 방법으로 평가한다. 즉, 각 사용자 u 마다 랜덤으로 I_u^+ 에서 하나의 상품을 추출하여 테스트 데이터를 만든다. 결과적으로 분리된 훈련 데이터 S_{train} 와 테스트 데이터 S_{test} 가 생성된다. 그런 다음 모델은 S_{train} 에서 학습하고 테스트 데이터 S_{test} 의 예측된 개인화 순위는 식 (22)와 같이 AUC에 의해 평가된다.

$$AUC = \frac{1}{|U|} \sum_u \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(\hat{x}_{ui} > \hat{x}_{uj}) \quad (22)$$

여기에서 사용자 u 당 평가 쌍은 $E(u) := \{(i,j) | (u,i) \in S_{test} \wedge (u,j) \notin (S_{test} \cup S_{train})\}$ 이며 AUC는 1에 가까울수록 성능이 더 좋다.

5.1.2 nDCG (normalized Discounted Cumulative Gain)

nDCG(normalized discounted cumulative gain, nDCG)는 순위 기반 추천 알고리즘에 주로 쓰이는 평가지표이다. 정확성 측정을 위한 척도로 추천된 상품들의 순서를 고려하기 위해서 nDCG를 사용한다. 검색엔진, 영상추천, 음악추천 등의 다양한 추천 알고리즘에서 평가지표로 활용되고 있다. 이 지표는 사용자별로 훈련 데이터에 포함되지 않은 상품 중에 예측된 순위 결과가 가장 높은 N 개 상품을 추천 상품으로 선택하여 테스트 데이터와 겹침 여부로 정확도를 평가한다. 사용자 u 에게 N 개의 상품을 추천할 때 암시적 피드백의 경우 $nDCG_u$ 는 식 (23)과 같이 정의한다 (김한준 외 2인, 2012; Gao et al., 2016).

$$nDCG_u@N = \frac{1}{Z_N} \sum_{i=1}^N \frac{I(rank_u^{-1}(i) \in I_u^+)}{\log_2(1+i)} \quad (23)$$

여기에서 Z_N 은 정규화 상수이며 가장 완벽한 순위일 경우 nDCG가 1이 될 수 있도록 조정하고, $rank_u^{-1}(i)$ 는 사용자 u 에게 i 번째로 추천되는 상품을 말한다. 종합적인 nDCG는 $nDCG_u$ 의 평균을 뜻하며 $nDCG@N = (1/|U|) \sum_{u \in U} nDCG_u@N$ 으로 정의한다 (김동우, 이은령, 2017). nDCG는 0.0에서 1.0까지의 범위이며 값이 클수록 우수한 검색 결과를 나타낸다 (이재웅, 이종욱, 2017).

5.2 실증 자료 분석

본 장에서는 실제 데이터에 SAMC-BPR 추천 알고리즘을 적용하기 위해 추천에 많이 이용되는 분야인 도서, 음악 등 네 가지 데이터를 이용하였고, 김민정 외 3인(2020)에서 분석한 실증 자료를 포함하고 있다. Python 패키지 ‘cornac’을 이용하여 행렬분해 알고리즘인 SVD, NMF, 베이지안 개인화 순위를 의미하는 BPR 알고리즘, 이를 조정 한 WBPR 알고리즘과 비교한다.

모든 알고리즘들은 각각 10번 실행하여 행렬의 차원인 인자의 수 k 와 추천 상품 개수 N 에 따른 평균 AUC와 nDCG@N을 계산하여 비교하였다. 실증 자료 분석을 위한 프로그램으로는 Python(3.7)과 R(4.0.2), C(4.9.2)를 사용하였다.

<표 1> 분석에서 사용되는 실제 데이터에 대한 정보

	OTT service	Travel review	Book crossing	Song list
Number of users	414	464	99	677
Number of items	10	24	85	203
Number of feedbacks	7,552	65,630	38,790	765,518
Sparsity	0.00%	3.06%	90.36%	95.33%
Train data	4,476	59,440	30,873	631,985
Test data	3,076	6,190	7,917	133,533

<표 1>은 분석에서 사용되는 실제 데이터들에 대한 정보이다. 콜드 스타트 문제를 피하고자 적어도 2개 이상의 상품에 대한 피드백이 있는 사용자들로 데이터를 구성하였다. 그리고 몬테카를로 방법인 SAMC-BPR 알고리즘과의 비교를 위해 행렬의 크기를 일부 축소하였다. OTT service, Travel review 데이터는 전체 공간에 비해 데이터가 있는 공간이 뾰뚱하게 차 있는 데이터를 의미하는 밀집 데이터(dense data)이다. 반면 Book crossing, Song list 데이터는 데이터가 90% 이상 비어있는 희소 데이터(sparse data)이다. 이처럼 다양한 경우의 데이터를 고려하여 밀집 데이터뿐만 아니라 결측값이 많아 적용하기 어려운 희소 데이터일 때 추천 알고리즘들의 성능을 비교하고자 한다. 밀집 데이터에서 희소 데이터 순으로 그리고 상품의 수가 적은 데이터에서 많아지는 데이터 순으로 분석 결과를 살펴본다.

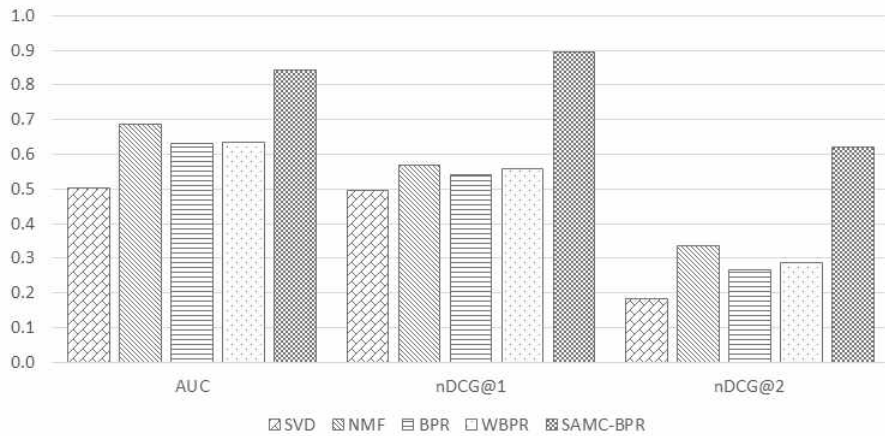
5.2.1 OTT service

OTT service 데이터 세트는 2019년 조사된 한국 미디어 패널 자료이며 개인용 설문지에서 온라인 디지털 콘텐츠 서비스에 대한 항목을 수집하였다. 사용자들이 이용한 서비스 중에 대중적으로 알려진 티빙(Tving), 폭(Pooq), oksusu, olleh TV 모바일, U+모바일TV, 에브리온TV, 네이버 시리즈, 곰TV, 넷플릭스, 카카오TV, 유튜브, 아마존 프라임 비디오 총 12개의 변수를 선택하였다. 그런데 2019년 9월, 기존의 폭(Pooq)과 oksusu가 wavve라는 이름으로 통합되었고, 에브리온TV가 서비스를 종료하였다. 그래서 폭(Pooq)과 oksusu는 wavve로 통합하고, 에브리온TV는 삭제하여 총 10개의 변수를 분석에 사용하였다. 이 변수들은 해당 콘텐츠 서비스를 이용하면 ‘예(1)’, 그렇지 않으면 ‘아니오(2)’로 표기되어있다. LOOCV 방법을 통하여 성능을 파악하기 위해 한 사람당 두 가지 이상의 서비스를 이용한 사용자는 총 414명이었으며, 사용자마다 임의로 1개씩 데이터를 추출하고 남은 데이터를 훈련 데이터로 사용하였다.

OTT service는 414×10 크기의 데이터이므로 행렬의 차원 k 를 10으로 지정하여 계산하였다. 학습률 α 와 정규화 상수 λ_θ 모두 0.001로 설정하였다. 개인화 중심의 미디어인 OTT 서비스는 대부분 비슷한 기능을 담고 있다. 보통 사용자들은 적은 수의 서비스를 이용하기 때문에 추천 목록을 2개까지 생성하여 N 이 1과 2인 경우만 고려하였다. <표 2>는 OTT service 데이터에 대한 결과이며 제안하는 방법과 다른 행렬분해 알고리즘들의 성능을 나타낸다. <그림 4>는 추천 알고리즘들의 AUC와 추천 상품 개수에 따른 nDCG의 결과이다.

<표 2> $k = 10$ 일 때 추천 알고리즘 결과 비교
(OTT service data)

Methods	AUC	nDCG@1	nDCG@2	time(sec)
SVD	0.5036	0.4949	0.1816	0.2
NMF	0.6877	0.5682	0.3345	0.4
BPR	0.6321	0.5415	0.2652	0.2
WBPR	0.6339	0.5587	0.2891	0.2
SAMC-BPR	0.8443	0.8952	0.6204	852.8



<그림 4> $k=10$ 일 때 추천 알고리즘 결과 비교 (OTT service data)

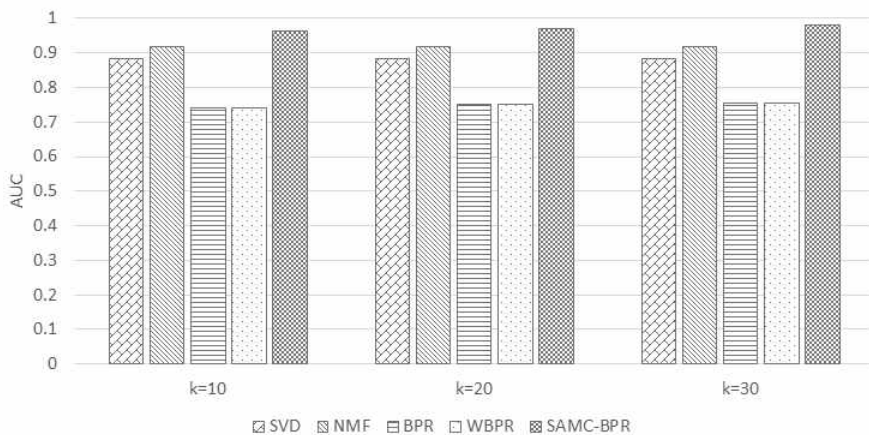
OTT service 데이터에서 AUC는 SAMC-BPR, NMF, WBPR, BPR, SVD 순으로 성능이 좋으며, SAMC-BPR이 0.844로 가장 월등한 결과를 내었다. 사용자에게 한 개의 추천 상품을 제공할 때의 nDCG@1은 SAMC-BPR, NMF, WBPR, BPR, SVD 알고리즘 순으로 높았다. $N=2$ 일 때도 마찬가지로 같은 결과였으며 SAMC-BPR은 다른 알고리즘과의 확연한 결과 차이를 보였다.

5.2.2 Travel review

Travel review 데이터 세트는 UCI Machine Learning Repository (University of California at Irvine)에서 제공되는 데이터를 이용하였다. 이 데이터는 Google 리뷰에서 수집되었으며 유럽 전역의 24개 카테고리의 명소에 대한 사용자 등급 리뷰이다. 24개 카테고리는 [1: churches, 2: resorts, 3: beaches, 4: parks, 5: theatres, 6: museums, 7: malls, 8: zoo, 9: restaurants, 10: pubs/bars, 11: local services, 12: burger/pizza shops, 13: hotels/other lodgings, 14: juice bars, 15: art galleries, 16: dance clubs, 17: swimming pools, 18: gyms, 19: bakeries, 20: beauty&spas, 21: cafes, 22: view points, 23: monuments, 24: gardens]로 구성되어 있다. 등급의 범위는 1~5이며 암시적 데이터로 취급하기 위해 3 이상을 '1', 나머지는 '0'으로 변환하였다. 10개 이상의 리뷰를 작성한 사용자들만 선택하여 총 464명의 사용자가 분석에 이용되었다.

Travel review 데이터 분석에서는 본 연구에서 제안하는 SAMC-BPR 알고리즘의 성능을 보다 구체적으로 보이기 위해 행렬의 차원과 추천 상품 개수에 따른 다양한 경우를 고려하였다.

학습률 α 와 정규화 상수 λ_θ 모두 0.01로 설정하고 행렬의 차원 $k=10, 20, 30$ 과 상위 추천 상품 개수 $N=1, 3, 5, 10$ 에 따라 각 알고리즘의 결과를 구하였다. <그림 5>는 k 에 따른 알고리즘들의 AUC를 비교한다. <표 3, 4, 5>와 <그림 6, 7, 8>은 k 와 N 에 따라 제안하는 방법과 다른 행렬분해 알고리즘들의 추천 성능을 나타낸다.

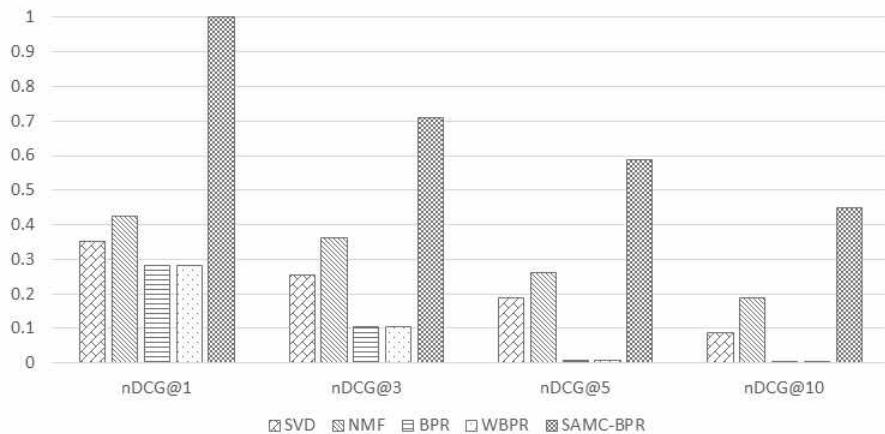


<그림 5> k 에 따른 AUC (Travel review data)

k 에 상관없이 SAMC-BPR의 AUC는 다른 알고리즘들에 비해 모두 0.96 이상으로 탁월한 성능을 보이고 있다. 그 뒤로는 NMF, SVD, BPR, WBPR 순이며 k 가 증가하더라도 알고리즘별로 성능의 차이는 거의 없었다.

<표 3> $k = 10$ 일 때 추천 알고리즘 결과 비교 (Travel review data)

Methods	AUC	nDCG@1	nDCG@3	nDCG@5	nDCG@10	time(sec)
SVD	0.8838	0.3522	0.2551	0.1875	0.0894	123.0
NMF	0.9178	0.4261	0.3630	0.2602	0.1870	143.8
BPR	0.7423	0.2809	0.1062	0.0085	0.0006	41.6
WBPR	0.7417	0.2807	0.1055	0.0089	0.0007	42.1
SAMC-BPR	0.9644	1.0000	0.7103	0.5894	0.4497	456.5

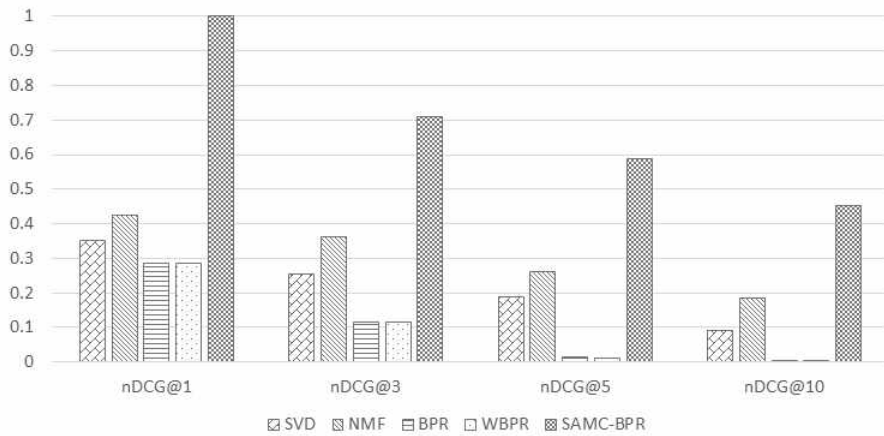


<그림 6> $k = 10$ 일 때 추천 상품 개수에 따른 nDCG
(Travel review data)

$k = 10$ 일 때, 상위 추천 상품 개수 N 에 상관없이 SAMC-BPR의 nDCG가 압도적으로 높았다. 나머지는 NMF, SVD, BPR, WBPR 알고리즘 순으로 높았으며 BPR과 WBPR의 차이는 거의 없었다. SAMC-BPR 알고리즘은 각각 nDCG@1에서는 1.0, nDCG@3에서는 0.7103, nDCG@5에서는 0.5894, nDCG@10에서는 0.4497의 결과를 나타내었으며 특히 $N = 1$ 일 때 1.0으로 가장 좋은 성능을 보이고 있다.

<표 4> $k = 20$ 일 때 추천 알고리즘 결과 비교 (Travel review data)

Methods	AUC	nDCG@1	nDCG@3	nDCG@5	nDCG@10	time(sec)
SVD	0.8840	0.3533	0.2550	0.1887	0.0900	126.5
NMF	0.9182	0.4260	0.3630	0.2603	0.1865	227.5
BPR	0.7505	0.2846	0.1173	0.0132	0.0012	42.4
WBPR	0.7503	0.2844	0.1163	0.0130	0.0013	41.1
SAMC-BPR	0.9687	1.0000	0.7103	0.5897	0.4511	767.8

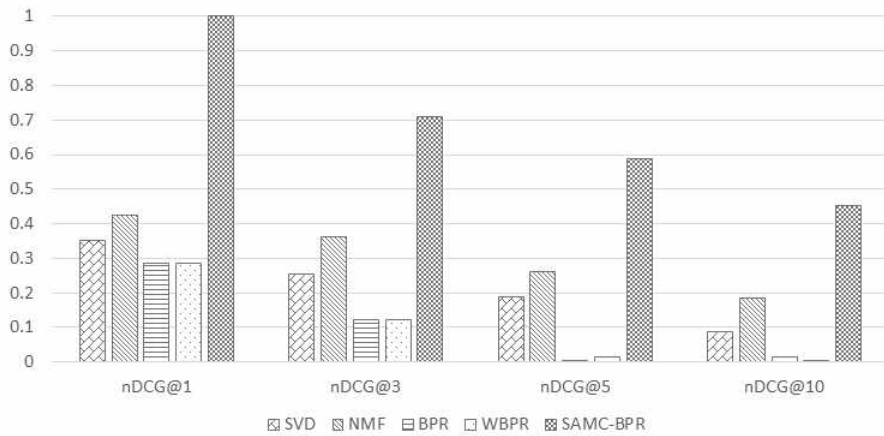


<그림 7> $k = 20$ 일 때 추천 상품 개수에 따른 nDCG
(Travel review data)

$k = 20$ 일 때 역시 SAMC-BPR의 nDCG가 가장 높았고 그 다음은 NMF, SVD, BPR, WBPR 알고리즘 순이었으며 BPR과 WBPR의 차이는 거의 없었다. SAMC-BPR 알고리즘은 각각 nDCG@1에서는 1.0, nDCG@3에서는 0.7103, nDCG@5에서는 0.5897, nDCG@10에서는 0.4511의 결과를 보이며, $k = 10$ 일 때보다 nDCG@5와 nDCG@10에서의 성능이 약간 향상된 것을 확인할 수 있다. 특히 SAMC-BPR 알고리즘은 $N = 1$ 일 때 1.0으로 우수한 추천 결과를 보이고 있다.

<표 5> $k = 30$ 일 때 추천 알고리즘 결과 비교 (Travel review data)

Methods	AUC	nDCG@1	nDCG@3	nDCG@5	nDCG@10	time(sec)
SVD	0.8840	0.3518	0.2537	0.1878	0.0883	131.3
NMF	0.9183	0.4261	0.3631	0.2604	0.1866	323.3
BPR	0.7541	0.2860	0.1223	0.0142	0.0012	133.4
WBPR	0.7540	0.2860	0.1224	0.0141	0.0013	128.4
SAMC-BPR	0.9813	1.0000	0.7103	0.5897	0.4513	1760.0



<그림 8> $k = 30$ 일 때 추천 상품 개수에 따른 nDCG
(Travel review data)

$k = 30$ 일 때도 마찬가지로 SAMC-BPR의 nDCG가 가장 높았고 그 다음은 NMF, SVD, BPR, WBPR 알고리즘 순이었다. SAMC-BPR 알고리즘은 각각 nDCG@1에서는 1.0, nDCG@3에서는 0.7103, nDCG@5에서는 0.5897, nDCG@10에서는 0.4513의 결과를 보이며 k 가 10과 20일 때보다 성능이 다소 향상된 것을 확인할 수 있다. 특히 SAMC-BPR 알고리즘은 $N = 1$ 일 때 1.0으로 우수한 추천 결과를 보이고 있다. Travel review 데이터는 상품의 개수가 24개로, 생성되는 훈련 데이터 쌍들이 적어 좋은 결과를 나타낸 것으로 판단된다. k 를 작게 설정한 이전의 결과보다 행렬의 차원이 증가할수록 약간의 성능 향상을 보이지만 물리적 시간을 고려하였을 때는 $k = 20$ 일 때의 결과가 적당해 보인다.

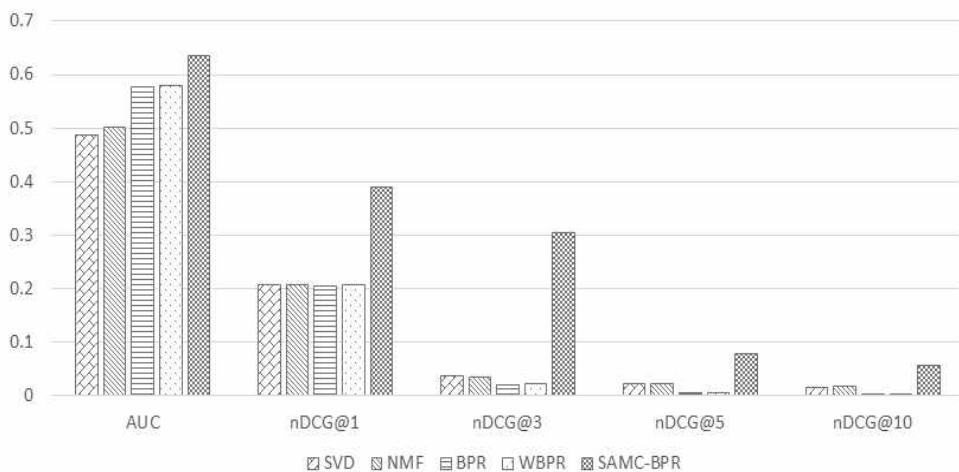
5.2.3 Book crossing

Book crossing 데이터 세트는 2004년 8~9월에 Book-Crossing 커뮤니티에서 4주 동안 크롤링하여 수집되었으며 추천시스템을 위한 공개된 데이터이다. 아래의 Github (<https://github.com/caserec/Datasets-for-Recommender-Systems>)에서 다운받을 수 있다. 등급의 범위는 1~10이며 등급이 1~5일 때 '0', 6~10일 때 '1'로 변환하였고, 책마다 3명 이상이 등급을 매겼거나 책 20권 이상의 피드백이 있는 사용자들을 데이터로 구성하였다.

앞서 Travel review 데이터로 보았을 때 k 에 상관없이 결과가 비슷하였다. 그러나 Book crossing 데이터는 Travel review 데이터보다 사용자, 상품 수가 늘어났으므로 물리적 시간을 반영하여 행렬의 차원을 20으로 지정하고 상위 추천 상품 개수 $N=1, 3, 5, 10$ 에 따른 결과를 비교하려 한다. 이때 학습률 α 와 정규화 상수 λ_θ 모두 0.01로 설정한다. <표 6>은 Book crossing 데이터에 대한 결과이며 제안하는 방법과 다른 행렬분해 알고리즘들의 성능을 나타낸다. <그림 9>는 알고리즘들의 AUC와 추천 상품 개수에 따른 nDCG에 대한 결과이다.

<표 6> $k=20$ 일 때 추천 알고리즘 결과 비교 (Book crossing data)

Methods	AUC	nDCG@1	nDCG@3	nDCG@5	nDCG@10	time(sec)
SVD	0.4869	0.2063	0.0364	0.0219	0.0143	6.3
NMF	0.5022	0.2083	0.0342	0.0216	0.0176	44.6
BPR	0.5774	0.2057	0.0210	0.0049	0.0010	12.6
WBPR	0.5805	0.2065	0.0230	0.0048	0.0012	12.2
SAMC-BPR	0.6355	0.3902	0.3055	0.0774	0.0569	717.9



<그림 9> $k = 20$ 일 때 추천 알고리즘 결과 비교 (Book crossing data)

Book crossing 데이터에서 AUC는 SAMC-BPR, WBPR, BPR, NMF, SVD 순으로 성능이 좋으며, SAMC-BPR이 0.6355로 다른 알고리즘들에 비해 우수한 성능을 보였다. 행렬의 차원 k 가 20일 때, 모든 알고리즘들은 상위 한 개의 추천 목록을 제공한 경우의 nDCG가 제일 높았다. 그 중에서 SAMC-BPR 알고리즘이 0.3902로 가장 높았고 뒤이어 NMF, WBPR, SVD, BPR 순이었으며 이들은 근소한 차이였다. nDCG는 추천 상품의 개수가 많아질수록 제안하는 방법과 다른 알고리즘 간의 격차가 많이 나는 것을 확인할 수 있다.

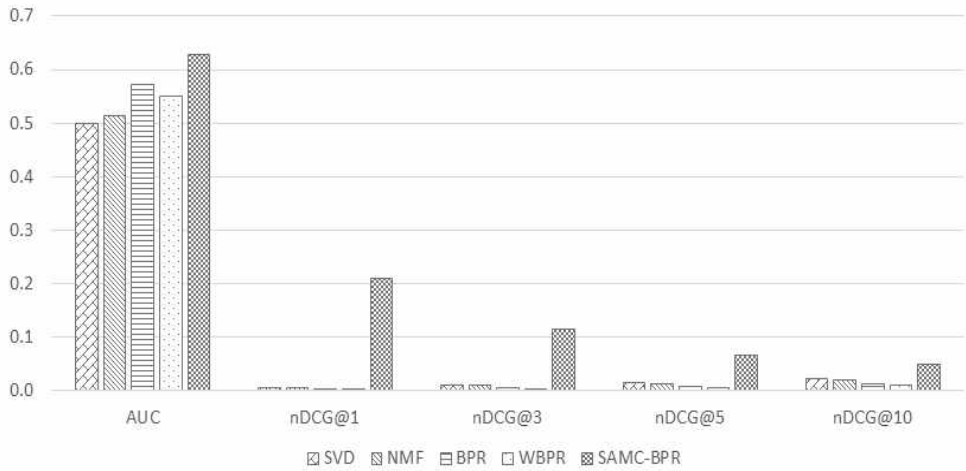
5.2.4 Song list

Song list 데이터 세트는 사용자들이 각자 소비한 노래에 대해 피드백을 남긴 데이터이다. 등급은 1~5로 구성되어 있으며 3 이상을 '1', 나머지는 '0'으로 변환하여 암시적 데이터로 취급하였다. 이 중에서 사용자와 노래 간의 10번 이상의 상호작용이 존재하는 경우만 선택하여 총 677명, 203개 노래에 대한 데이터를 생성하였다. 이 데이터는 분석하는 실증 자료 중에서 가장 크기가 클 뿐만 아니라 행렬에서 95.33%가 아무런 값을 가지지 않아 매우 희소하다. 즉, 한 명의 사용자가 피드백하는 상품 수는 전체와 비교했을 때 극도로 제한되어 있다.

이때 행렬의 차원을 $k = 20$ 이라 지정하고, 학습률 α 와 정규화 상수 λ_θ 모두 0.01로 설정한다. <표 7>은 Song list 데이터에 대한 결과이며 제안하는 방법과 다른 행렬분해 알고리즘들의 성능을 나타낸다. <그림 10>은 알고리즘들의 AUC와 추천 상품 개수에 따른 nDCG에 대한 결과이다.

<표 7> $k = 20$ 일 때 추천 알고리즘 결과 비교 (Song list data)

Methods	AUC	nDCG@1	nDCG@3	nDCG@5	nDCG@10	time(sec)
SVD	0.4993	0.0047	0.0106	0.0147	0.0224	10.6
NMF	0.5129	0.0047	0.0101	0.0138	0.0212	1017.3
BPR	0.5730	0.0022	0.0049	0.0074	0.0135	157.9
WBPR	0.5509	0.0018	0.0038	0.0056	0.0106	154.7
SAMC-BPR	0.6271	0.2106	0.1141	0.0662	0.0493	2811.3



<그림 10> $k = 20$ 일 때 추천 알고리즘 결과 비교 (Song list data)

Song list 데이터에서 AUC는 SAMC-BPR, BPR, WBPR, NMF, SVD 순으로 성능이 좋으며, SAMC-BPR이 0.6271로 다른 알고리즘들에 비해 우수한 성능을 보였다. 행렬의 차원 k 가 20일 때, SAMC-BPR 알고리즘만 상위 한 개의 추천 목록을 제공한 경우의 nDCG@1이 제일 높고 나머지 알고리즘들은 제공하는 상품의 개수가 많아질수록 성능이 좋아져 nDCG@10가 가장 높았다. 그래서 SAMC-BPR 알고리즘과 다른 행렬분해 알고리즘들은 nDCG@1에서 차이가 가장 많이 났고 N 이 커질수록 격차가 좁혀졌다.

제 6 장 결 론

개인화 추천이란 인공지능이 사용자의 취향에 맞는 재화나 서비스를 추천해주는 것을 의미한다. 과거에는 사용자의 신상 정보만을 수집하는 데 그쳤지만, 최근에는 검색 이력이나 구매 이력과 같은 일상 정보까지 수집하면서 취향을 알기 쉬워졌다. 우리는 일상생활에서 개인화 추천을 자주 경험하고 있다. 유튜브나 넷플릭스는 이용자의 취향을 분석하여 영상을 추천해주는 대표적인 개인화 추천 서비스이다. 온라인 쇼핑몰에서 둘러본 상품이 소셜네트워크서비스(SNS) 피드에 광고로 노출되는 것도 개인화 추천의 예라고 할 수 있다. 이처럼 더 나은 개인화 추천을 제공하기 위해 독자적인 분석방법과 추천 알고리즘을 끊임없이 개발하고 있다.

명시적인 사용자 데이터는 쉽게 얻지 못하므로 광범위하게 적용 가능한 암시적 데이터를 사용하여 적절한 추천 알고리즘을 개발하는 것은 중요하다. 암시적 데이터를 기반으로 하는 BPR 알고리즘은 기존의 추천 알고리즘들보다 우수한 성능을 보여주는 것으로 알려져 있다. 그러나 BPR 알고리즘에 이용된 확률적 경사 하강법은 단순하고 구현도 쉽지만, 방향에 따라 기울기가 달라지는 함수인 비등방성(anisotropy) 함수에서 탐색 경로가 비효율적이다. 이를 극복하기 위해 SAMC 알고리즘을 이용한 개인화 추천 알고리즘을 제시하고 기존의 BPR 알고리즘과 이를 변형한 WBPR 알고리즘, 그리고 행렬분해의 대표적인 알고리즘인 SVD, NMF와의 성능을 비교하였다.

실증 자료 분석 데이터는 모두 명시적 데이터로 일부 평점을 '1', 나머지를 '0'으로 변환하여 암시적 데이터의 형태로 만들었다. 그리고 콜드 스타트 문제를 피하고자 상품마다 2개 이상의 피드백이 있는 사용자들의 데이터만 고려하였고, 몬테카를로 방법인 SAMC-BPR 알고리즘과 비교를 위해 사용자와 상품을 임의로 선택하여 행렬의 크기를 축소하였다.

Travel review 데이터에서 SAMC-BPR 알고리즘은 행렬의 차원 k 와 상위 추천 개수 N 에 상관없이 AUC, nDCG 평가에서 탁월한 성능을 보였고, k 가 증가하더라도 알고리즘별로 성능의 차이는 거의 없었다. 또한, 밀집 데이터인 OTT service 데이터에서 k 를 10으로 지정했을 때와 희소 데이터인 Book crossing, Song list 데이터에서 20으로 했을 때 모두 제안하는 SAMC-BPR 알고리즘이 다른 알고리즘보다 높은 성능을 보였다. 특히 상위 하나의 상품을 추천했을 때 정확도가 압도적으로 높았다. 따라서 밀집 데이터뿐만 아니라 희소 데이터에서도 SAMC-BPR 알고리즘의 성능이 우수하며 추천 개수에 상관없이 정확한 순위 결과를 예측하여 순위 기반 추천에 효과적이라고 말할 수 있다.

모든 사람의 취향을 저장할 수 있는 완벽한 추천 알고리즘은 없다. 하지만 SAMC-BPR 추천 알고리즘을 이용해 저마다 다른 개성과 취향을 가진 소비자에게 만족도를 높일 수 있도록 개인화 맞춤 추천 목록을 제안하면 서비스 제공자 입장에서도 서비스 구매 전환율을 높이거나 활성 사용자 수를 늘리는 기회가 될 것이다.

본 연구에서 제안하는 알고리즘은 몬테카를로 방법이기 때문에 표본의 크기가 커질수록 컴퓨팅 시간이 증가하는 한계점이 있다. 따라서 대량 데이터일 경우에 이를 해결하는 방법을 보완한다면 더욱 다양한 분석이 이루어질 수 있을 것으로 기대한다.

참 고 문 헌

- [1] 강부식, 유성진, 홍한국. (2008). 협업 여과기법의 추천 성능 개선을 위한 유사도 척도 비교, *Journal of the Korean Data Analysis Society*, 10(5), 2893-2906.
- [2] 김동우, 이은령. (2017). Modified Bayesian personalized ranking for non-binary implicit feedback, *The Korean Journal of Applied Statistics*, 30(6), 1015-1025.
- [3] 김민정, 장성봉, 서영수, 전수영. (2020). SAMC 알고리즘을 이용한 미디어 패널 개인화 추천 알고리즘, *Journal of the Korean Data Analysis Society*, 22(5), 1835-1847.
- [4] 김상화, 오병화, 김문중, 양지훈. (2011). 협력적 필터링과 콘텐츠 정보를 결합한 영화 추천 알고리즘, *The Korean Institute of Information Scientists and Engineers*, 38(2B), 275-278.
- [5] 김한준, 노준호, 장재영. (2012). 웹 검색 개인화를 위한 개념네트워크 프로파일 기반 순위 재조정 기법, *The Journal of the Institute of Internet, Broadcasting and Communication*, 12(2), 69-76.
- [6] 이재웅, 이종욱. (2017). 상위 N개 항목의 추천 정확도 향상을 위한 효과적인 선호도 표현방법, *The Korean Institute of Information Scientists and Engineers*, 44(6), 621-627.
- [7] 이현종, 정호경, 김광섭, 윤도현. (2011). 개인화된 추천 시스템을 위한 평가 방안 및 SVD를 통한 개인화된 추천의 성능 향상, *The Korean Institute of Information Scientists and Engineers*, 38(2B), 283-286.

- [8] 정성원. (2019). 벡터공간모델을 활용한 상품추천 알고리즘에 관한 실증연구, 박사학위논문, 숭실대학교 대학원.
- [9] Gautner, Z., Drumond, L., Freudenthaler, C., Schmidt-Thieme, L. (2012). Personalized ranking for non-uniformly sampled items, *Journal of Machine Learning Research*, 18, 231-247.
- [10] Guo, C., Wu, S., Wang, L., Tan, T. (2016). Personalized ranking for non-uniformly sampled items, *Neurocomputing*, 214, 191-200.
- [11] Liang, F., Liu, C., Carroll, R. (2007). Stochastic approximation in Monte Carlo computation, *Journal of the American Statistical Association*, 102(447), 305-320.
- [12] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback, *In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452-461.