

# Matrix Factorization Techniques for Recommender Systems

## 1.Introduction

Matrix Factorization은 협업 필터링(collaborative filtering)의 방식, 그 중에서도 잠재 요인 방법(latent factor method) 중 하나이다. 이는 어떤 특성을 명시하는 방식으로 진행되는 콘텐츠 기반 필터링(contents based filtering)과는 다르게 유저들의 이전 플레이 기록이나 서비스 평가 기록 등 과거 유저의 행동에 초점을 두고 유저-서비스 간의 상관관계를 찾아내는 방식으로 진행이 된다.

이 방법은 기본적으로 유저의 서비스 플레이 패턴으로부터 추론한 요인 벡터를 통해 서비스와 유저 사이의 특성을 잡아내며, 아직 특정 서비스를 사용하지 않은 유저와 서비스 사이에 강한 관련성이 있다는 결과가 도출되면 이 서비스를 추천할 수 있게 된다.

## 2.MF Methods and A Basic MF Model

Matrix Factorization(이하 MF)를 수행하기 위해서는 입력 데이터로 특정 서비스에 대한 명시적 피드백(explicit feedback)이 가장 좋은 데이터가 될 것이지만, 특정 서비스에 대한 유저의 평가 정보는 찾아보기 힘들다. 또한, 존재한다고 하더라도 이를 유저-서비스 행렬로 나타낸다면 희소(sparse) 행렬이 될 수 밖에 없다. 이렇게 명시적 피드백을 이용하기 어려울 때는 암시적 피드백(implicit feedback)을 이용하게 되는데, 여기서 이용 가능한 것이 플레이 타임 등이다.

MF는 유저와 서비스를 f차원의 결합 잠재 요인 공간에 매핑하는데, 유저-서비스의 관계는 이 공간에서 내적을 이용해서 계산되게 된다. 식으로 표현하면 다음과 같다.

$$\hat{r}_{ui} = q_i^T p_u$$

여기서 서비스 i는  $q_i$ 로, 유저 u는  $p_u$ 라는 벡터로 표현이 되었고, 그 둘의 내적인  $r$ 은 둘 사이의 상호작용을 의미하며 이는 곧 서비스에 대한 유저의 전반적인 관심을 표현한다고 볼 수 있다.

이 문제는 SVD(Singular Vector Decomposition)과 유사한데, 현실에서는 유저-서비스 행렬에 결측값이 존재하기 때문에 직접적인 SVD를 이용하는 것은 불가능하고 이를 대체하는 방법 또한 정확하지 않다.

따라서, 기록된 데이터만으로 모델링을 하는 방법이 제시되었으며 실제 상호작용  $r$ 과 예측된  $\hat{r}$ 사이의 거리를 최소화 하는 방식으로 목적함수를 정의하게 된다. 식은 다음과 같다.

$$\min_{q,p} \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

위의 식에서 앞 부분은 실제 오차를 줄이는 과정을 의미하고 뒷 부분은 과적합을 방지하기 위한 규제 항을 의미한다. 결과적으로 이 모델은 결측된 부분을 예측해야 하기 때문에 과적합 방지가 필수이다.

## 3. Learning Algorithms and Adding Biases

위의 목적식을 최소화하기 위해 본 논문에서는 두 가지 방법을 제시하였다.

### 1.Stochastic Gradient Descent

이 알고리즘은 이번 논문에서 처음 제시된 논문이 아니고 딥러닝 모델을 최적화하기 위한 경사하강법 중에서 가장 널리 사용 되는 방법 중 하나이기 때문에 익숙한 방법이다. 로직은 다음과 같다.

각각의 훈련 세트에 유저-서비스의 상호작용인  $r$ 을 예측하고 다음과 같이 예측 오차를 계산한다.

$$e_{ui} = r_{ui} - q_i^T p_u$$

이후  $q$ 와  $p$ 를 다음과 같은 규칙으로 업데이트 하면서 오차를 줄여나가게 된다.

$$\begin{aligned} q_i &:= q_i + \gamma(e_{ui}p_u - \lambda q_i) \\ p_u &:= p_u + \gamma(e_{ui}q_i - \lambda p_u) \end{aligned}$$

## 2.Alternating Least Squares

이 방법은 E-M 알고리즘과 매우 흡사하게 작동하게 된다. 처음에는  $q$ 를 고정시키고 최적화를 진행하고 다음에는  $p$ 를 고정시키고 최적화를 진행하고 이를 지속적으로 반복한다. 이 방법은 다음의 경우에 큰 효과를 발휘할 수 있다.

- 연산 장치가 병렬화를 지원하는 경우
- 입력 데이터가 암시적 데이터에 집중되어 있는 경우

## 3.Adding Biases

위에서 정의한 목적식은 유저와 사용자의 상호관계를 파악하기 위한 것인데, 어떤 경우에는 유저나 서비스의 자체적인 특성이 플레이 결과에 영향을 미치게 된다. 이를 위해서 bias 또는 intercept를 목적식에 추가하는 방식을 이용하기도 한다. 식은 다음과 같다.

$$\begin{aligned} \hat{r}_{ui} &= \mu + b_i + b_u + q_i^T p_u \\ \min_{p,q,b} \sum_{(u,i) \in K} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2) \end{aligned}$$

## 4. Additional Input Sources and Temporal Dynamics

어떠한 서비스도 이용한 경험이 없는 신규 유저나 새롭게 출시한 서비스에 대해서 MF를 적용해야 될 때가 있는데, 이 문제를 cold start 문제라고 한다. 이를 위해서 해당 유저에 대한 모든 정보들을 활용해 암시적 피드백  $N(u)$ 를 만들고 이를 정규화 시킨 정보를 이용한다. 또한, 인구학적 정보와 같은 사용자 속성 정보가 있다면 이를  $y$ 라고 표현하며 이를 모두 포함한 상호관계  $r$ 은 다음과 같다.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T [p_u + |N(u)|^{-0.5} \sum_{i \in N(u)} x_i + \sum_{a \in A(u)} y_a]$$

또한, 시간이 변화하면 유저의 서비스 선호도가 변할 수 있지만 지금까지 정의한 모델은 이를 반영하지 못하는 정적(static)인 모델이다. 이를 동적(dynamic)으로 바꾸어주기 위해서 Temporal Effect를 정의하고 추가할 수 있다.

$b_i(t)$ : 아이템의 인기는 시간에 따라 변한다.

$b_u(t)$ : 사용자의 성향도 시간에 따라 변한다. (baseline rating)

$p_u(t)$ : 시간이 흐름에 따라 아이템에 대한 사용자의 선호는 변화할 수 있다.

정리

이 논문은 2009년에 게재된 논문으로 당시에는 추천 분야에서 가장 혁신적인 방법이었고 이를 이용해서 다양한 서비스들이 큰 성공을 거두었다. 하지만 현재는 추천을 위한 더 정교한 방법론들이 개발 되었으며 다양한 현실의 상황에 맞게 변화했다.

-EOD-