

An Introduction to Recommender Systems

1.1 Introduction

- Web이 전자상거래의 매개로서의 비중이 커지면서 추천 시스템 기술 발전에 대한 필요가 커졌다.
- 이런 형상의 촉매가 되는 요인은 Web은 users로 부터 긍정 또는 부정의 피드백을 받을 수 있다는 것이다.
(TV, 뉴스 등 단방향 매체가 아니라 Web은 사용자의 행동이 로그로 남기 때문에 양방향 매체가 될 수 있다.)
- 가장 대표적인 피드백 방법은 "ratings"이다. users가 수치적으로 가치를 매기는 평가 시스템이다.
(예, 별점 5점 만점)
- 다른 형태의 feedback은 ratings만큼 explicit하지 않지만 Web-centric paradigm에서는 비교적 쉽게 얻을 수 있는 정보들이다.
예를 들어 item을 사거나 browsing하는 단순한 행동은 해당 item에 대한 관심으로 볼 수 있습니다.
- 추천시스템의 가장 기본적인 아이디어는 고객의 관심을 표현하는 다양한 data 소스를 이용하는 것이다.
- recommendation analysis는 이전에 존재했던 users와 items 간의 관계에 기반한다. 왜냐하면 과거의 관심과 성향이 미래에 할 선택에 대한 좋은 지표가 되기 때문이다.
다른 기법으로는 'knowledge-based recommender systems'가 있다. 이는 users의 과거 행동 기록이 아닌 user-specified requirements에 기반한다.
- 추천 시스템 알고리즘 동작의 기저에 있는 기본적인 원칙은 무엇일까?
바로 user and item centric activity 에는 significant dependencies 존재한다는 것이다.
- 이러한 의존성은 ratings matrix를 data-driven manner에서 '학습'가능하다.
- 비슷한 제품에 비슷한 관심을 가지는 users들의 구매 또는 평점 정보를 모아 'cohorts'를 만들 수 있다.
이 cohorts의 관심과 행동은 cohorts 내의 개개인에 대한 추천을 할 수 있게 한다.
- 위 설명은 가장 단순한 추천 시스템 알고리즘 family인 'neighborhood models'에 기반한다. 이 family는 더 넓은 모델 class인 'collaborative filtering'에 속한다.
여기서 collaborative filtering의 의미는 다수의 users로 부터 얻은 평점 정보를 collaborative 방식으로 이용/가공하여 missing ratings를 예측하는 것이다.
- 추천 시스템은 다양한 부가 정보(wide variety of auxiliary data types)들 때문에 더 복잡하고

data-rich(?)할 수 있다.

- 예를 들어, 'content-based recommender systems'는 content가 주된 역할을 하게된다. 즉 user의 rating과 item의 특징 정보가 함께 사용된다.
이 idea는 유저가 과거에 관심을 가졌던 아이템에 대한 속성(properties, attributes)에 기인하여 모델링된다는 것이다.
- 또 다른 framework는 knowledge-based systems이다. 사용자가 상호작용적으로 관심사를 명시하고 추천을 하기 위해서 도메인 지식과 사용자의 설명이 결합되어 진다.
- 더 advanced models에서는 contextual data(temporal information, external knowledge, location information, social information or network information)들이 사용될 수 있다.

1.2 Goals of Recommender Systems

- 추천 문제는 다양한 방식으로 공식화 할 수 있다. 대표적인 두 가지 모델을 살펴보자.
 1. Prediction version of problem
 - User-item 조합에서 rating value를 예측하는 것이다.
 - m 명의 users와 n 개의 items를 가정하자. 이는 $m \times n$ matrix로 표현된다. 이때 **specified (or observed) values** 는 학습에 사용되고 **missing(or unobserved) values** 는 모델이 예측해야하는 값이 된다.
 2. Ranking version of problem
 - 이 방법은 특정 item에 대한 user의 ratings를 예측할 필요없다..
 - 대신 특정 유저에게 top-k items를 추천해주거나 특정 아이템을 타겟으로 하는 top-k users를 정하는 것이다.
 - Top-k items가 top-k users 보다 더 일반적이다.
 - Top-k recommendation problem이라고 불리기도 합니다. 이는 일종에 추천 문제의 ranking formulation이다.
- 위 설명 중 두번째 방법은 ratings의 예측을 정확한 값으로 추론할 필요는 없다.
- 첫 번째 방법이 일반적이다. 왜냐하면 첫 번째 방법을 해결함으로써 두 번째 방법까지 이용가능하다. 하지만 두 번째 ranking version의 문제가 상대적으로 해결하기 쉽다. Ch 13에서 더 다룬다고 한다.
- 궁극적인 추천 시스템의 목적은 매출의 증가이다. 정교한 추천을 통해 users는 items에 관심을 가지고 이는 판매 규모와 수익의 증가를 가져올 수 있다. 하지만 수익 증가의 결과에는 다양한 요소들이 영향을 미친다.(즉 추천 시스템의 성능 지표로 사용하기엔 어렵단 뜻?)
- 그래서 수익 증가라는 넓은 관점의 비즈니스 중심의 목표를 이루기 위해, 추천 시스템에는 **operational** 과 **technical** goals가 존재한다.

1. **Relevance** : 가장 명확한 operational goal은 관련있는 users에게 직접 아이템을 추천하는 것이다. 유저는 본인이 관심을 가지는 item을 소비하길 원한다. 물론 relevance가 추천 시스템에서 가장 주된 operational goal이라고 해도 독립적으로 고려하기엔 충분치 못 하다. 따라서 아래의 요소들도 부가적인 목표로 한다. 이는 relevance 만큼 중요하진 않지만 그래도 꽤 영향력이 있다.

2. **Novelty** : 추천 시스템은 user가 과거에 본적 없는 item을 제안할 할때 매우 유용하다. 특히 유명한 items만 계속 추천할때 제품의 다양성이 떨어지는 문제도 고려되기 때문이다.

3. **Serendipity** : (조건에 맞는?) 명확한 추천을 하는 것과 반대로 예상치 못한 추천을 하는 것이다. 단순히 몰랐던 것이 아니라 완전히 놀라운 추천을 한다는 점에서 Novelty와 다르다. 이는 특정 item만 소비해오던 users가 다른 타입의 items에 잠재적 관심을 발견하는 경우이다.

(예를 들어, 동네에 인도 식당이 새로 오픈해서 이 인도 식당을 추천하는건 novelty지만, 갑자기 이디오피언 음식을 추천해주는건 serendipity이다.)

전혀 뜬금없는 추천을 한다는 disadvantages도 있지만 이는 장기적으로 봤을때 serendipity가 가져올 장점에 비하면 큰 문제는 아니다.

4. **Increasing recommendation diversity** : top-k items를 추천할때 k개의 items들이 다 비슷하면 users가 하나도 마음에 안 들어 할 위험이 있다. 반대로 추천 후보군이 다양하면 users는 자신의 취향에 맞는 item을 찾을 가능성이 커진다. 즉 추천에 있어서 다양성은 users가 매번 비슷한 items만 추천 받는 지루한 상황을 피할 수 있게 하는 이점이 있다.

- 위 와 같이 구체적인 goals 뿐만 아니라 users나 판매자 입장에서 다양한 soft goals들이 존재한다.
- users관점에서 추천 시스템은 Web site에 대한 전반적인 users의 만족도가 올라가게 한다. 예를 들어 아마존에서 마음에 드는 추천을 계속 받을 경우 users는 계속해서 이 웹사이트를 방문할 것이다. 이는 이후 이 site에서 소비가 발생하고 판매가 증가할 수 있는 상황으로 이어질 수 있다.
- 판매자의 관점에서 users의 니즈를 파악하는 insight를 제공하고 users의 경험을 customize하는데 도움을 준다.
- 마지막으로 user에게 왜 이 특정 item을 추천했는지 설명할때 유용하다. 이후에 이 책에서 explanations에 적합한 알고리즘들을 소개한다.

GroupLens Recommender System

- Pioneering recommender system - Usenet news의 추천을 위한 연구 프로토타입으로 만들어 졌다.
- Usenet news의 임의의 뉴스에 대한 구독자들의 평점 정보를 모아서 아직 그 뉴스를 보지 못 한 유저들의 평점을 예측한다.

- 초기에 몇몇 자동화된 collaborative filtering 알고리즘들이 GroupLens 세팅에서 개발되었다.
 - 이 그룹에서 발전된 general ideas는 books(BookLens)나 movies(MovieLens)로 확장되었다.
 - 초기 collaborative filtering research에 대한 연구뿐만 아니라 데이터들이 벤치마크가 쉽게 되지 못 했던 시기에 몇몇 데이터셋을 공개하였다.
- MovieLens로 부터 세 개의 데이터셋이 가장 두드러지는 examples이다.

Amazon.com Recommender System

- Amazon.com의 추천 시스템에는 평점, 구매 행동, browsing 행동이 제공된다. 평점은 별점 5점으로 평가되어집니다. 특정 고객의 구매, browsing 데이터는 users가 로그인했을때 수집된다.
- users가 로그인 할때마다 메인 페이지에 추천이 이루어진다.
- 대부분의 경우에 추천에 대한 설명이 제공된다. 예를 들어, 이전에 구매한 제품과 추천할 상품 사이의 연관성이 추천 시스템 인터페이스에 포함된다.
- 구매나, browsing 행동은 implicit rating으로 볼 수 있다.
- 많은 모델들이 명시적인 피드백과 암묵적인 피드백을 동시에 고려할 수 있도록 설계되어 있다.

Netflix Movie Recommender System

- 초기 넷플릭스는 메일로 주문을 하는 DVD 대여 서비스 회사였다. 그러다 스트리밍으로 확장하였다. 현재는 구독 위주의 영화나 티비 시리즈 스트리밍 회사이다.
- 넷플릭스는 유저들이 영화나 티비 쇼에 대하여 5-point 평가를 할 수 있다. 거기에 유저의 시청 기록이 저장되어 있다.
- 넷플릭스는 유저에게 추천 items에 대한 설명을 잘 해낸다. 유저가 시청한 items에 기반하여 추천을 한다. 이러한 정보는 유저가 특정 영화를 볼지 안 볼지에 대한 추가적인 정보도 제공해준다.(?)
- 추천에 대한 의미있는 설명을 제공하는 것은 user가 왜 이 특정 items에 흥미를 가지는 지 이해 시킬 수 있어 중요하다. 이는 고객의 충성도를 높이고 서비스에 오래 머물도록 할 수 있다.
- Netflix Prize contest를 열어 연구 커뮤니티에 제공하였다. 다양한 collaborative filtering algorithms로 경쟁하는 포럼을 제공하였다. Netflix movie ratings 데이터가 공개되고 특정 user-item combinations에서 평점을 예측하는 것이 문제이다.

Google News Personalization System

Facebook Friend Recommendations

1.2.1 The Spectrum of Recommendation Applications

1.3 Basic Models of Recommender Systems

1.4 Domain-Specific Challenges in Recommender Systems

1.5 Advanced Topics and Applications

1.6 Summary

Exercises