

Chapter 5. Knowledge-Based Recommender Systems

기존 추천시스템의 알고리즘들의 문제를 해결하기 위해, Knowledge-based recommender system 활용.

- Collaborative Filtering systems: 많은 ratings가 필요 → 그렇지 않으면 cold-start user/item 문제 발생.
- Content-based: cold-start item 문제는 해결, but cold-start user 문제는 여전히 남아있음.

CF, CB

: customized한 아이템에 잘 맞지 않는 추천기법

Knowledge-based recommender system

: 부동산, 자동차, 관광, financial services, 럭셔리 제품들은 판매량이 많지 않고, 충분한 ratings 정보가 없음. 또한 해당 item들에 대한 도메인 영역은 매우 복잡함.

ex) 집을 사고자 할 때, 평수, 방 개수, 화장실 개수 등의 것들을 명확하게 원함. 즉, 부동산 아이템은 설명하는 데에 있어서 복잡성이 매우 커서, user가 비슷한 아이템에 대해 과거에 남긴 ratings를 반영하여 합리적인 ratings set을 구하는 것 자체가 어렵다.

그렇다면, 어떻게 이런 customization, paucity(ratings 부족)한 item을 다룰 것인가?

→ knowledge-based recommender system을 활용.

1) explicitly soliciting user requirements for such items

: 유저의 조건(requirements)를 직접적으로 명시적으로 요청한다.

(-) 유저가 그들의 조건 전체를 말하지 않음

(-) 유저들이 해당 조건에 매칭되는 product가 있는지 알지 못한다.

2) 따라서, **interactive feedback**(상호작용 피드백)을 사용하며, 유저가 이 피드백을 통해

본질적으로 복잡한 아이템 도메인을 탐색 및 다양한 옵션 사이에서 사용할 수 있는 절충안을 배우게 한다.

3) 적합한 item: 주로 정기적으로 팔리지 않는 아이템들

영화를 볼 때는 어떤 내가 직접 요구사항으로 적어넣은 것 없이 추천된 영화를 봐도 무리가 없지만, 만약 내가 집을 구하는데 세부적으로 원하는 요구사항 정보를 넣지 않고 추천된 집은 절대 보지 않을 것임.

- customer가 그들의 요구사항을 직접 입력하고자하며, interactive가 중요한 요소인 경우.
- product complexity 때문에 item에 대한 ratings 정보를 얻기 힘든 경우.

- 시간에 민감한 경우.(차, 컴퓨터의 경우에는 시간이 지나면서 발전하고, 예전 제품들은 더이상 찾지않게 됨)

4) Knowledge-based 추천 시스템의 두 가지 types

- case-based: case들을 유사도행렬과 연계하여 검색을 안내
해당 타겟 케이스에 대한 비슷한 아이템을 검색하기 위해서, 아이템속성에 대한 유사도행렬을 정의. (유사도 행렬 정의는 도메인 영역마다 다를 수 있음.)
interactive하게 수정되어 나온 결과는 새로운 타겟 케이스로 사용됨.
- constraint-based: 구체적인 rule/criteria들을 활용하여 검색을 안내
user가 직접 아이템 속성에 대한 요구나 제약조건을 명시하고, 계속해서 수정 가능.
어떠한 조건에 대해 검색해본 후, 더 넓은/좁은 범위에 대해 다시 검색하는 등
interactively repeat를 반복하여 원하는 결과를 얻어냄.
- 공통점: user에게 요구를 수정할 기회가 주어진다는 공통점

5) interaction types

목표: user에게 복잡한 product space를 잘 안내하기 위함.

- conversational systems
feedback loop에서 결정됨
- search-based systems
사전에 정해진 질문 순서에 따름
(ex."Do you prefer a house in a suburban area or within the city?")
→ constraint-based에 잘 맞음
- navigation-based recommendation (=critiquing recommender system)
user가 현재 추천된 itemset에서 변경을 요청하여 검색을 다시해나감.
+ 해당 상황에서 사용자가 더 원하는 것에 가까워지도록 suggest해주는 방법
→case-based에 잘 맞음

5.1 Introduction

1) Content-based 및 CF 모델은 상당한 양의 과거 구매 기록 혹은 평점 기록을 요구함.

-> 데이터가 부족할 경우, 성능이 떨어지거나 전체 user-item 조합의 전체 스펙트럼에 대해 전부 커버하지 못함.

2) 이러한 문제를 cold-start 문제라고 함. 시스템마다 이러한 문제에 대한 민감성이 다름.

(CF 시스템은 가장 민감. 신규 유저와 아이템 모두에 잘 다루지 못함. CB 시스템은 새로운 아이템은 다소 잘 다루지만, 새로운 유저에 대해 잘 대응하지 못함.)

3) 또한 이러한 시스템(CF, CB)들은 상품이 고도로 맞춤화(customized)된 도메인의 경우 대체로 적합하지 않음.(ex. 부동산, 자동차, 관광 상품, 재무상품, 사치품 등.)

-> 이러한 상품은 자주 구매되지 않으므로 평점이나 구매기록이 부족할 수밖에 없고, 제품의 도메인이 복잡할 수 있으며, 특정 제품 성질을 갖는 비슷한 상품이 적음. 더욱이 과거 평점이 현재의 기호를 반영한다고 보기도 힘들.

4) Knowledge-Based recsys는 상품에 대해 고객의 요구사항을 표면적으로 요구하는 방식에 의존한다.

5) 그러나 고객에게 조차 요구사항을 말하거나, 어떤 상품이 그 요구사항에 가능할지 이해하기도 어려움.

따라서 이러한 system은 유저의 interactive한 feedback을 사용함. 이는 복잡한 제품 공간을 탐색하게 하고, 다양한 옵션간의 상충하는 성질들을 배우게끔 유도함.

6) 검색과 탐색 과정은 제품 도메인의 다양한 피쳐들간의 trade-off, 효익을 묘사하는 지식 기반에 의해 활용될 수 있음.

7) 여기서 지식 기반(knowledge bases)은 효과적인 검색과 탐색에 매우 중요하기 때문에, 이러한 시스템을 knowledge based recommender system이라 부름.

8) 이러한 KB 시스템은 정기적으로 구매되지 않는 제품을 추천할 때 아주 적합함. 또한 이러한 도메인에서는, 유저들은 일반적으로 그들의 요구사항을 명백히 드러내는 경향이 있음.

9) 일반적으로 KB가 적합한 상황

1. 유저가 그들의 요구사항을 명백하게 상술하기를 바랄 경우

-> 따라서 상호작용 친화성이 이러한 시스템의 매우 중요한 부분임. CB나 CF는 이러한 상술된 feedback을 허용하지 않음.

2. 특정 상품의 평점을 얻기 어려움. 상당한 제품 도메인의 복잡도에 기인함.

3. 평점이 시간에 민감한 속성을 가짐 (e.g. 컴퓨터) 구형 차량 혹은 컴퓨터에 대한 평점은 유용하지 않음.

10) KB를 CB/CF와 구별짓는 중요한 속성은 KB는 특정 분야에 대한 높은 수준의 customization임. 이러한 customization은 유관한 도메인 지식을 유사도 혹은 제약 축도의 형태로 encoding하여 사용함으로써 얻어낼 수 있음.

상품 정보뿐만 아니라 유저 정보도 활용해 도메인에 특화된 속성을 만들어낼 수 있음.

그러나 이러한 속성은 KB 시스템에 범용적인 것이 아니라, 유저의 요구사항에 특화되어 있음.

11) KB 추천 시스템에는 두가지 주된 타입이 있음.

1. 제약 기반 추천 시스템(constraint-based Recsys):

이러한 시스템에서, 유저는 요구사항 혹은 제약사항을 상술함. (e.g., 어떤 특성의 상한 혹은 하한)

여기에 도메인 특화된 규칙들이 유저의 요구사항을 맞추기 위해 추가적으로 적용됨. 이 규칙은 제약으로 작용하여, 추천할 상품을 filtering함.

2. 사례 기반 추천 시스템(case-based Recsys):

이러한 시스템에서, 유저는 특정 사례/표본들을 특정지어주고, 이는 목표 혹은 중점 고려사항으로 활용됨. 유사도 기반으로 이 사례들과 유사한 상품들을 검색하고 제공해줌.

따라서, 유사도 척도가 시스템 내부에서 사용될 도메인 지식을 형성함.

12) 시스템은 유저가 상술한 요구사항을 수정하도록 여지를 제공함. 이러한 방식으로 유저가 궁극적으로 원하는 상품을 찾아내도록 유도함.

그러나 이를 유도하는 방식이 위의 두 KB 시스템에서 차이를 보임.

13) 유저와 시스템의 상호작용은 크게 세가지로 분류됨.

1. 대화 시스템(Conversational systems):

feedback loop의 맥락에서 유저 선호가 결정됨. 이는 도메인 영역이 복잡하고, 반복적인 대화 형식의 맥락에서만 결정될 수 있을 때 사용됨.

2. 탐색 기반 시스템(Search-based systems):

이미 정리된 일련의 질문에 의해 유저의 선호가 명시됨.

3. 네비게이션 기반 추천(Navigation-based recommendation):

현재 추천된 아이템에 대해 수정 요청사항을 유저가 제공함. 반복적인 요청 변경을 통해, 유저가 원하는 제품에 도달할 수 있게 됨.

constraint-based recsys

compatibility condition: 인원을 정해주면 방의 크기를 정해준다는 등의 방식으로 상품을 고객의 상황에 맞게 조정

case-based recsys

critique method: 유저가 좋아하지 않은 상품이 나왔을 때 어떻게 대처할지?

1) simple critique: 기존 feature중 하나만 바꿔 가지고 classification 수행

장점 -> 추천이 명확하고 직관적임, 유저에게 어필 가능

단점 -> laborious navigation(조건이 더 길어지고, 다른 피처에 대해 통제력이 떨어짐.. 원래 원하는 추천에서 멀어질 수 있음)

2) compound critique: 여러 feature를 혼합하여 수행

3) dynamic critique: 유저들이 빈번하게 바꾸는 조합을 활용.