Chapter 4. Content-Based Recommender Systems

Chapter 4.1 Introduction

2,3 단원에서 다룬 CF 모델은 유저들 사이의 평점 패턴간의 상관관계를 활용해 추천을 제공했음.
반면에, 이러한 방법론들은 예측을 위해 상품의 특성을 활용하지 않음. 이는 다소 낭비스러워 보인다.

1) 컨텐츠 기반으로 추천을 제공하는 경우 유저 자신의 평점과 다른 영화에 대한 행동이 유의미한 추천을 발견하는 데 충분함.

2) 이러한 접근은 아이템이 새로 등장해서 rating이 많이 없을 때 유용함.

3) CF와 결합할 경우 이를 hybrid라고 함.

4) Content-based 추천 모델(이하 CB)은 크게 2개의 데이터 소스에 의존함
1. 상품의 설명과 같은 컨텐츠 중심의 특성(주로 판매자가 제공하는 text 설명)
2. 유저 프로파일 정보
-> 고객이 명시한 취향에 대한 정보를 활용 가능
-> 아마 demographic 정보는 사용하지 않는 것으로 보임(?)

5) 다른 유저의 평점은 CB 추천시스템에서 아무런 영향을 끼치지 않는다는 것에 주목

6) CB모델은 새로운 아이템의 cold-start문제를 완화시킴. item에 대한 rating 정보가 하나도 없기때문에, CF로는 추천 불가 → 새로운 아이템에서 attributes를 추출해서 예측에 활용.
↔ 반면, CB 모델은 신규 유저의 cold-start 문제를 해결할수는 없음
(cold start item문제는 완화 but cold start user 문제는 해결불가)
: CB에서는 user profile을 만드는 과정이 어렵기때문에, 새로운 user에 추천 힘듦.

→*cold-start item 시나리오: 충분한 정보가 쌓이기 전까지는 CB사용. 그 이후 CF 사용*

7) 다양성과 참신성이 훼손됨. 유저가 좋았다고 표현한 것들만을 활용함.

8) CB 모델은 비정형이고, 텍스트가 풍부한 도메인의 추천에 특히 유용함.

9) CB 모델은 KB(knowledge based) 모델과 매우 밀접한 관계를 가짐.
가장 핵심적인 차이는 KB는 유저 요구사항의 명시적 스펙(explicit specification)과 함께 유저와 추천시스템간의 상호적 인터페이스를 지원한다는 것.
반면에 CB는 일반적으로 과거 평점을 사용한 학습 기반 접근법을 취함.

그러므로, KB는 추천 프로세스에서의 유저 컨트롤에 더 우수하고,
CB는 과거 행동을 더욱 효율적으로 활용함.

10) 그럼에도, 이러한 차이는 엄청나게 유의하지는 않음.
일부 CB는 명시적으로 유저의 관심사를 명시하도록 만듦.

11) hybrid: 일부 시스템은 학습 기반과 상호작용 기반을 모두 통합해 사용함. 이를 하이브리드라 부름.

4.2 Basic Components of Content-Based Systems Content-based
다양한 종류의 CB 모델에서도 한가지 공유하고 있는 변하지 않는 특성이 있음. 이는 텍스트 도메인 위주라는 것.
뉴스 추천의 경우 매우 text-centric하다. 이러한 text를 처리하기 위해서는 unstructured를 structured로 바꾸는 것이 중요

1) 크게 3가지로 구성됨
1. offline preprocessing (전처리 및 특징 추출)
: Web pages, product descriptions, news, music features 등 다양한 형태의 비정형 데이터를 keyword based vector-space representation으로 변환해주는 작업. 매우 domain-specific한 작업임.

2. offline learning (유저의 과거 평점 혹은 행동을 기반으로 모델 학습함.)
: 과거 평점/활동 데이터와 상품 특성 데이터는 결합되어 training set을 만들어냄.
결과 모델은 "user profile"으로 일컫어지는데, 이는 개념적으로 user의 흥미와 item attribute를 연관짓기 때문.

-> off-the-shelf classification model을 사용
: 재고품의, 기성품인 ... 즉, 쉽게 구해올 수 있는 일반적인 머신러닝 모델을 의미함

3. online predicting (filtering and recommendation)
: 이전 단계에서 학습된 모델로 특정 유저에 대한 예측을 수행함. 실시간으로 수행하기 위해서는 효율성이 매우 중요한 문제임.

4.3 Preprocessing and Feature Extraction
CB 모델의 첫번째 단계는 아이템의 차별적인(discriminative) 특징을 추출해 내는 것.
이러한 차별적인 특징은 유저의 관심사를 잘 예측하는 특징을 말함. 이는 도메인에 매우 의존함.

4.3.1 Feature Extraction
1) data representation 그 자체를 쓰는 것보다는 키워드를 추출해서 사용하는 것이 효율적.

2) 제목/내용과 같은 위계적인 구조의 경우 더 중요한 제목을 더 무게를 주어 접근

- 영화 추천의 경우
  - 영화는 시놉시스(줄거리 요약), 감독, 배우, 장르 등의 구성 요소가 있음
  - 왓챠에서는 특정 배우나 감독의 영화를 추천해 주기도 함
  - 여러 요소들에 있는 단어들의 중요도가 모두 같지 않아, 중요도를 도메인 지식으로 결정하기도 함. 보통은 주연 배우, 감독 등의 키워드가 시놉시스에 있는 단어보다 중요
- 웹 페이지 추천의 경우
  - 분석 알고리즘에서는 웹 문서의 제목, 메타데이터, body 내용 등 여러 요소들의 중요성을 매김. 예) 제목은 body보다 높은 weight
  - anchor text: 링크로 연결된 웹 페이지에 대한 description이 있음. 다만 해당 페이지 자체와는 관련이 없을 수 있어서, 해당 페이지에서는 제거되고 anchor text가 가리키는 페이지에 포함되기도 함. anchor text는 가리키는 문서에 대한 description이기 때문
  - 웹 페이지에는 광고, disclaimer 등 해당 문서의 내용과 연관되어 있지 않은 내용들이 있기 때문에, main block 내용만을 추출하기도 함. 다만 사이트의 구조에 따라서 main block을 정의하기 어려울 수도 있어, 머신러닝을 통해 사이트의 tag tree를 가지고 구조를 학습하기도 함. main block을 라벨링하는 것도 일종의 분류 문제라고 할 수 있음
- 음악 추천의 경우
  - music genome project: 모든 음악들을 구조화하고 태깅하는 프로젝트. synth riffs, straight drum beats 같이 트랙들의 피쳐를 추출
  - 이 프로젝트 데이터를 Pandora Internet radio라는 추천 엔진에서 사용. 유저들은 이 중 **자신이 원하는 피쳐를 선택**하면, 해당 피쳐를 이용하여 비슷한 노래가 재생됨 -> knowledge-based에 가까움
  - 노래에 좋아요/싫어요 평가 가능. 이런 유저 피드백은 더 정교한 추천 모델을 만드는 데 사용 -> Content-based가 됨

4.3.2 Feature Representation and Cleaning

비정형 데이터에서 representaion을 추출하기 위해서는 적절한 전처리가 선행되어야함.

이 전처리의 대표적인 3가지를 설명

1) stop-word removal(불용어 처리)
: a, an, the
2) stemming
: hoping -> hope
3) phrase extraction(구절 추출)
: hot dog -> "hot dog" (not 'hot', 'dog')

4) tf-idf 계산

: 문장 내 단어의 중요도를 계산

- tf(d,t): 문서 d에서 단어 t의 개수
- idf(t): 단어 t가 등장한 문서 개수. log(전체 문서 개수 / 해당 단어가 있는 문서 개수). inverse이기 때문에, 특정 문서에서만 등장할수록 커짐.
- 모든 문서에서 자주 등장하는 단어는 중요도가 낮다고 판단하며, 특정 문서에서만 자주 등장하는 단어는 중요도가 높다고 판단함.
- the나 a와 같은 관사의 경우 모든 문서에 자주 등장하기 때문에 중요도가 작아짐

4.3.3 Collecting User Likes and Dislikes

Rocchio classifier: Nearest Centroid classifier

1. 유저의 선호는 4가지의 형태로 구해질 수 있음.
    1) rating
    2) implicit feedback
    3) text opinion : 처리 필요
    4) cases
        : 유저는 어떤 특정 예시 혹은 사례를 들어 그들이 관심있는 상품을 특정하려 할 수도 있음.
        이러한 경우에는 Rocchio, KNN classifier를 가지고 활용할 수 있음.
        그러나 이런 유사도 방식의 검색보다는 KB 모델의 세부 분야인 Case-based system이 더 적절할 수 있음.
        이 방식은 학습 방식이 아닌 도메인 지식을 활용해 추천을 제공하는 차이점이 있음.

4.3.4. Supervised Feature Selection and Weighting

변수 선택과 가중치 부여의 목적은 가장 정보가 많은 단어들이 vector-space representaion에 사용될 수 있도록 보장하기 위함임.

이를 위해 gini index, entropy, 카이제곱 통계량, 표준화 편차(Normalized Deviation), Feature weighting 등이 사용됨.

카이제곱 검정은 카이제곱 분포에 기초한 통계적 방법으로, 관찰된 빈도가 기대되는 빈도와 의미있게 다른지의 여부를 검증하기 위해 사용되는 검증방법임.

키워드가 등장한 경우 고객이 특정 상품에 관심이 있는지 여부가 유관한지를 알고 싶다고 할 때,

등장 여부와 구매 여부를 기준으로 contingency table을 만듦.

그리고 단어의 등장과 고객의 흥미 여부가 독립이라고 가정.

만약 두 수가 독립적이면, 학습과정에서 유의미하지 않다고 판단할 수 있음.

1) 중요한 단어를 찾아내는 과정에서, 불용어처리와 tf-idf는 각각 feature selection과 feature weighting의 한 예시임.

그러나 이들은 각각 유저의 선호 정보와는 무관한 방식으로 얻어지는 unsupervised

4.3.4.1 Gini Index

-> feature selection으로 사용됨.

지니 계수란, 불순도를 측정하는 지표로써 m개의 범주 내에 얼마나 잘 구분되어 있는지를 표현한 값임.

만일 모든 범주 내에 정확히 분리해 낸다면, 불순도는 0이 될 것.

이전 상태의 gini index와 현재 상태의 gini index를 information gain이라고 하고,

이 값이 가장 커지는 변수를 사용해 분지를 실시한다(decision tree)

4.3.4.2 Entropy

-> Gini index와 매우 유사함. 그러나 log를 취함으로써 정규화 과정을 거치게 됨

4.3.4.3 χ2-Statistic

-> 카이제곱 통계량은 feature selection에 사용되는 방법임.

1) 예시

특정 단어가 유저의 구매 관심과 연관이 있는지를 확인하고 싶은 경우를 가정.

상품 중 10%를 구매하고, 특정 단어 w가 전체 상품 중 20%에 등장한다고 해보자.

전체 상품과, 그에 해당하는 문서의 전체 개수는 1,000개. 이 경우, 다음과 같은 contingency table이 가능.

	term w occurs in description	term w doesn't occur
User bought item	1000*0.1*0.2 = 20	1000*0.1*0.8 = 80
User not bought	1000*0.9*0.2 = 180	1000*0.9*0.8 = 720

이 값들은 단어의 등장과 해당 상품에 대한 유저의 관심은 독립이라고 가정된 채로 계산됨.

만일 두 수량이(특정 단어 등장과 유저의 구매 수량이) 독립적이라고 한다면, 해당 단어는 학습 과정에 무관하게 될 것임.

그러나, 실제로는, 상품과 구매한 상품(item at hand)은 깊이 연관되어 있을 것임.

일례로, 해당 단어를 포함한 경우 특정 상품을 구매할 확률이 매우 높다고 해보자. 이 경우, contingency table은 아래처럼 달라질 것임.

	term w occurs in description	term w doesn't occur
User bought item	60	80
User not bought	140	760

2) 카이제곱 통계량은 contingency table 내부의 다양한 cell에 대한 실제 관측된 값과 예측값의 표준화된 편차를 계산함.

3) 또한, 카이제곱 통계량을 예측값을 굳이 계산하지 않고서도 관측값의 함수로써 계산할수도 있다. 이는 예측값 역시 관측값의 행/열간 전체 값의 함수값이기 때문임.

4) 이 값은 카이제곱 분포를 사용한 확률적 신뢰구간을 계산하는 데 사용될 수도 있음.

5) 그러나, 카이제곱 통계량의 값이 클수록 특정 단어와 상품이 높은 수준으로 관련이 돼 있다는 의미로 이해하는 것으로도 충분하다.

6) 반대로, 완전히 기댓값과 동일한 경우를 상정할 경우 카이제곱 통계량은 0이 나오게 되는데, 이는 특정 단어와 상품이 전혀 무관하다는 것이 된다.

7) 결과적으로 많은 단어 중 가장 높은 카이제곱 통계량을 갖는 단어를 가지고 피처를 선택할 수 있다.

4.3.4.4 표준화된 편차

앞에 언급된 방법들의 문제는 모두 0/1과 같은 discrete한 feedback만 활용함. 따라서 rating간의 상대적인 순서를 반영하지는 못함.

rating 간의 상대적인 평가 순서가 중요한 정보를 함축할 경우, 표준화된 편차 방식이 유용할 수 있음.

μ+(w)는 단어 w를 포함한 문서의 평균 평점, μ-(w)는 w를 포함하지 않은 문서의 평균 평점임.

이 둘을 빼주고 절대값으로 감싸 줌. 그리고 모든 문서 평점의 표준편차로 나누어 주어, 표준화를 시켜줌.

-> 이 값이 클수록 discriminatory word임을 의미함.

-> 비슷한 지표로 Fisher's discrimination index가 있음.
이는 평점 공간 내가 아닌 피처 공간 내에서의 클래스 내부의 분할을 비교함.
단, 이 방식은 범주형 독립 변수에 더 적합함. (평점 데이터가 아닌 0/1 데이터)

4.3.4.5 Feature Weighting

soft한 방식의 feature selection이라고 할 수 있음.
앞서 살펴본 방식들의 단어별 결과값을 가중치로 놓는 방식으로 활용 가능.

4.4 Learning User Profiles and Filtering

- 분류/회귀 알고리즘과 유사한 방식으로 유저의 Rating을 학습
  - Rating이 discrete할 경우(좋아요/싫어요): 분류 문제
  - Rating이 numeric할 경우(0~10점으로 평가): 회귀 문제
- Train Items의 Label(Rating)을 통해 Test Items의 Rating 예측. 예측에 사용되는 독립변수는 각 아이템의 Item Attributes (vector-space representaion)
- Nearest Neighborhood Classifier(cosine similarity와 같이 아이템 간의 유사도 계산), Bayes Classifier, Rule-Based Classifier 등으로 예측

Chapter 4.5 Content-Based Versus Collaborative Recommendations

Content-Based 추천 방법의 장점

1. Cold start problems for new items 어느정도 해결
   : 새로운 item(ratings이 없는)추천 가능.
2. item의 feature측면에서 추천에의 설명 제공 가능
3. off-the-shelf text classifiers와 함께 사용됨.

Content-Based 추천 방법의 단점

1. Overspecialization: user가 이전에 봤던 item과 비슷한 것만 추천해줌.
   - user는 사실 많은 interest가 있지만, e-commerce 사이트(예를 들면 쿠팡)에서 갖고 있는 해당 유저에 대한 user profile이외의 것은 추천하지 않는 문제
   - 샀던/봤던 특정 아이템 추천에만 과도하게 치우치는 문제
2. cold start problems for new **users**
   : text classification모델은 overfitting방지를 위해 충분히 큰 트레이닝 도큐먼트를 필요로 하므로 CB에서의 문제가 더 심각.