Recommender Systems

"top-n" 추천 리스트 메트릭에 기반한 추천기 평가에 대해서 연구자들은 좀 더 관심을 가지게 되었다.

Introduction

특정 유저에게 어떤 책이 꼭 보여야 하는지 결정하는 소프트웨어 시스템이 바로 추천 시스템이다.

모든 추천 시스템은 반드시 유저 모델이나 유저 프로파일을 개발하거나 유지 해야 한다.

이 책은 두개의 파트로 나눠져 있는데 ,

파트 1, collarborative, content-based, knowledge-based recommendation, hybridization

explaining the reasons for recommending an item, evaluating the quality of recommendation systems

파트 2, 는 뭔가 재미 없을 거 같음

파트1. 기조 컨셉에 대한 소개

1.1.1 Collaborative recommendation

유저가 과거에 유사한 기호를 공유 했다면 ? 미래에도 비슷한 특성을 가질 것이다.

> 유저 A와 유저 B가 강하게 오버랩되는 구매 히스토리를 가지고, A가 B가 보지 못한 책을 샀다면, 기본적으로 이 책을 B에게 제시해야 한다.

CF의 배경에는 아래와 같은 의문이 있다.

- 추천에 필요한 비슷한 취향을 가진 고객을 어떻게 찾을 것인가?
- 유사도를 어떻게 측정할 것인가?
- 새로운 유저에게 어떻게 해야함, 아직 구매를 안했으면 ?
- 아무도 안산 아이템은 어떻게 함?
- 이용할 rating이 몇 가지 없으면?
- 유저가 이 아이템을 좋아할 지 예측하는 다른 기술?

순수한 CF는 아이템들에 대한 지식을 필요로 하지 않는다.

> 서점을 예로 들면, 책이 어떤 건지, 책의 장르, 누가 썼는지는 알 필요가 없다.

1.1.2 Content-based recommendation

추천 시스템은 두 가지 다른 목적으로 제공 될 수 있다. 유저가 특정한 책을 사거나 특정한 영화를 보도록 자극할 수 있다. 그리고 정보 과다를 해결하는 도구로서 큰 데이터 셋에서 가장 흥미로운 아이템을 고르게 한다.

정보 검색과 정보 필터링

컨텐츠-기반 추천은 아이템 프로 파일과 설명문의 유효성(availability)에 기반 한다.

Content-based recommendation의 배경에는 아래와 같은 의문이 있다.

- 시스템이 어떻게 유저 프로파일을 자동적으로 확보 하거나 지속적으로 개선 시키는가?
- 어떻게 유저의 관심사와 특정 아이템이 매치하는지? 적어도 유사한지? 어떻게 결정하는가?
- 자동으로 아이템 설명을 뽑거나 학습하기 위해 어떤 테크닉이 사용될 수 있는가?

Content-Based Recommendation은 두 가지 강점을 가진다.

1. 추천 정확도를 얻기 위해서 큰 유저 그룹을 필요로 하지 않는다

2. 새로운 아이템들이 즉시 추천 될 수 있다. 아이템 속성이 사용 가능해지면

1.1.3 Knowledge-based recommendation

추천 시스템은 필요 하다. 부가적인 그리고 수단과 목적(means-end) 방향의 지식을 필요로 한다.

이러한 지식 기반 접근 방법에서, 추천 시스템은 일반적으로 현재 사용자와 이용가능한 아이템들에 대한 부가적 이거나 메뉴얼 적으로 주어지는 정보를 사용한다. Contraint-based 추천 시스템은 지식 기반 접근의 예로 다양한 측면에 대한 고려한다.

> 카메라라고 예를 들자면, 화소, 무게, 가격과 같은 제품의 디테일한 상품의 정보를 가지는데 높은 화소를 가지는 카메라는 사진 출력을 원하는 고객에게 추천 하는 추천 시스템

고객의 프로파일 정보를 잘 유지하고 있다면 좀 더 다양한 추천을 할 수 있다.

이 챕터에서 다뤄지는 측면은 "고객 상호작용"이다.

아이템들의 특정들에 대한 디테일한 기술적인 이해 뿐만 아니라 많은 수의 아이템 특징들 기반한 시나리오를 생성할 필요가 있다.

다이얼로그를 통해 선호도를 알아 낼 수 있는 대화를 주고받는 듯한 스타일의 추천 시스템

1.1.4 하이브리드 접근 방법

각 접근법의 장점과 단점을 결합한다.

2. Collaborative recommendation

과거 행동에 대한 정보 혹은 유저 커뮤니티의 의견을 이용해서 현재 시스템을 사용하는 유저가 가장 좋아하거나 흥미를 보일만한 아이템을 예측하는 것이다.

2.1 User-based nearest neighbor recommendation

Rating 데이터에 기반해 특정 유저와 선호도가 비슷한 유저를 구체화한다. 그리고 전체 아이템에 대해 그 유저가 보지 않은 아이템의 Rating을 계산한다.

2.1.2 Better similarity and weighting metrics

Pearson's correlation coefficient, adjusted cosine similarity, Spearman's rank correlation coefficient, mean squared difference, cosine similarity(outperforms)

2.1.3 Neighborhood selection

특정 유저에 대해 전체 유저 정보는 낭비다. Neighbor의 사이즈를 축소할 필요가 있다. 흔한 방법으로 threshold를 정하거나 k 번째로 유사한 특정 유저만 사용하기 가 있다.

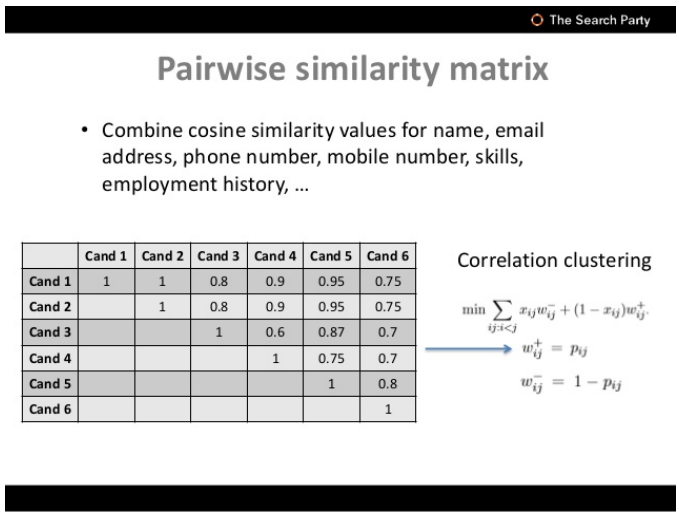
2.2 Item-based nearest neighbor recommendation

아이템 기반 알고리즘의 메인 아이디어는 아이템 간의 유사도를 사용해서 예측 하는 것이다.

2.2.1 The cosine similarity measure

2.2.2 Preprocessing data for item-based filtering

아이템 유사도 매트릭스를 개선해서 구축하는 방법은 모든 아이템에 대해 pairwise similarity



	Cand 1	Cand 2	Cand 3	Cand 4	Cand 5	Cand 6
Cand 1	1	1	0.8	0.9	0.95	0.75
Cand 2		1	0.8	0.9	0.95	0.75
Cand 3			1	0.6	0.87	0.7
Cand 4				1	0.75	0.7
Cand 5					1	0.8
Cand 6						1

모델 기반의 subsampling, 결국에는 computational cost 를 줄이기 위함

2.3 About ratings

2.3.1 Implicit and explicit ratings

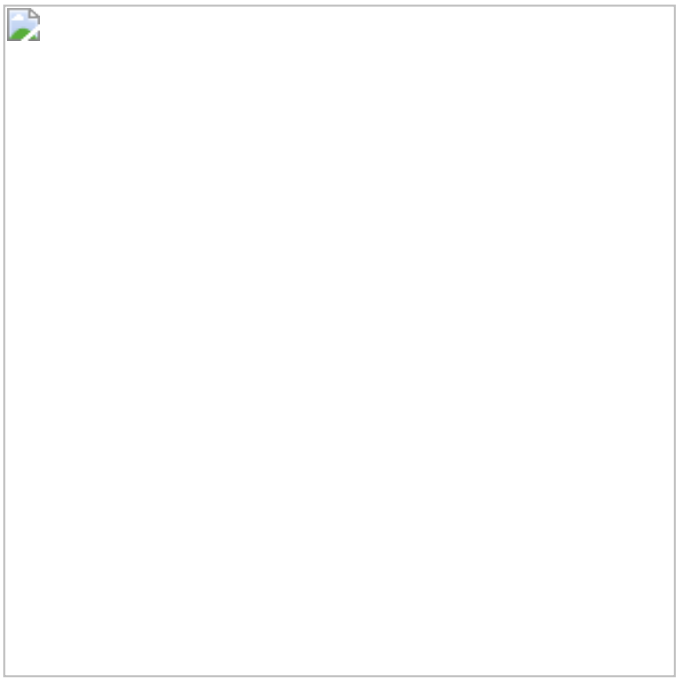
유저의 의견을 모으는 데서 가장 정확한 것은, explit item rating을 묻는 것이 가장 정확하다.

서로 다른 rating scale을 사용하는데 granularity를 높일 수록 추천의 질이 달라진다. 영화에서 평점 5점은 사실 잘 세분화 된 평가 항목이 아닐 수도 있다.

implicit ratings는 즉시 모이고 유저 단에서의 추가적으로 할 게 없지만, 고객 행동을 제대로 해석 했는가 하는 문제가 있다.

2.3.2 Data sparsity and the cold-start problem

rating은 sparse하다. 그걸 보충하는데 유저를 잘 분류하기 위해 유저에 대한 부가 정보를 이용한다



그래프-기반의 메소드가 주목이 받았을 때가 있었다. transitivity (전이성, 노드와 엣지의 관계성)

rating matrix를 바탕으로 유저 들과 아이템들 간의 bipartite graph를 구축한다.

Default Voting 방법도 98년도에 있긴 했는데 별로 좋지는 않았다.

2.4 Further model-based and preprocessing-based approaches

CF 추천 기술은 메모리-기반과 모델-기반 중 하나로 분류된다. 전통적인 유저 기반 기술은 rating 데이터가 메모리-기반으로 메모리에서 즉시 생성된다. 모델-기반의 기술은 데이터를 오프라인으로 처리하고, 필터링이나 차원 감소를 한 뒤에 모델을 학습한다. 모델-

기반의 접근 방법이 전체 데이터를 사용하기에 이론적으로 더 정확하지만, 전체 데이터를 다 사용하기에는 부담이 된다.

2.4.1 Matrix Factorization / latent factor models

matrix factorization 메소드는 추천 시스템에서 latent(hidden) factor를 rating 패턴과 유저와 아이템의 특징을 factor들의 벡터로 도출하는데 사용 된다.

SVD(Singular Value Decomposition), LSA(Latent Semantic Analysis), LSI(Latent Semantic Indexing)

- SVD - 주어진 매트릭스 M에서 총 3개의 매트릭스로 분해 된다. ((4×4) = (4×2) (2×2)(2×4))
    - U는 아이템, V는 유저

2.4.2 Association rule mining

큰 세일즈 판에서 관계의 패턴을 구체화 하는 것이다.

X가 아이템1 과 아이템2를 좋아하면, X는 아이템5도 좋아할 것이다. rule-mining 알고리즘의 목적은 자동으로 법칙을 발견하고 이러한 법칙의 품질을 계산하는 것이다.
association rule을 위한 표준 측정 방법은 support 와 confidence이다.

- support - x와 y가 전체 거래에서 함께 등장할 확률
- confidence - x가 등장하는 거래 중 x,y 가 함께 등장하는 확률

association rule에 추가적인 cut-off로 계산 복잡도를 줄이고 sparse한 아이템도 추천 잘 되게 만들었다

2.4.3 Probabilistic recommendation approaches

CF 를 확률 적인 메소드와 함께 구현하는 방법은 확률 문제를 분류 문제로 보는 것이다. 베이지안 분류기

과거에 샀었던 아이템들을 X(조건부 독립)로 특정 아이템의 rating을 예측한다

K-means, Bayesian Network

새로운 유저에 대한 문제를 active learning approach로 해결한다.는 의견

2.5 Recent practical approaches and systems

2.5.1 Slope One predictors

pair of item, rating을 매기지 않은 item의 rating을 예측하기 위해 co-rated item

- 다른 사람들이 매긴 그 아이템의 rating을 사용
- rating을 매긴 item을 사용

성능을 증가시키기 위해 "좋아요" "싫어요"에 대한 가중치 편차를 두는 것이다

2.5.2 The Google News personalization engine

실시간으로 채워지는 뉴스 개인화 컨텐츠 이기 때문에 모델-기반과 메모리-기반의 기술이 사용된다.

- PLSI (Probabilistic Latent Semantic Indexing)
  - CF를 위한 확률 기술. 확률적 클러스터링
- MinHash
- co-visit