[Paper Review]

Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions

Gediminas Adomavicius, Member, IEEE, and Alexander Tuzhilin, Member, IEEE

● 추천시스템 개요

- 추천시스템은 *ratings structure*에 의존한다. user에 의해 주어진 ratings을 통해 아직 만들어지지 않은 ratings을을 예측하고 가장 추정된 rating이 높은 item을 추천한다.

$$\forall c \in C,\ s'_c = \mathrm{argmax}_{s \in S}\ u(c,s).$$

C : set of all users

S : set of all items

u : utility function that measures the usefulness of item s to user c

: $C \times S \rightarrow \mathbb{R}$

R : totally ordered set

● 추천시스템 종류

1. predict the absolute values

- ***Content-based recommendations*** : The user will be recommended items similar to the ones the user preferred in the past
- ***Collaborative recommendations*** **:** The user will be recommended items that people with similar tastes and preferences liked in the past
- ***Hybrid approaches*** : Theses methods combine collaborative and content-based methods

2. predict the relative preferences

- ***preference-based filtering*** : These would focus on predicting the correct relative order of the items

- > 이러한 것들은 1)heuristic/memory(define)한 방법으로 추정할 수도, 2)estimating/statistics/model(optimize)한 방법으로 추정할 수도 있다.

● Content-Based Methods

- user c에 의해 점수가 높게 점수가 매겨진 s와 similar한 정도인 utilities $u(c,s_i)$에 근거해 상품을 추천
- content(s)는 item s로 부터 추출된 features의 집합인 item profile이다. text-based items에선 contents가 *keywords*로 묘사된다.

▶ utilities $u(c,s_i)$를 계산하는 방법

<variables>

N : 총 문서의 모든 단어 개수(중복 포함)

K : 총 문서의 모든 단어 개수(중복 제외)

d_j : 문서 j

k_j : 문서 j의 keyword

k_i : keyword i

n_i : keyword i의 총 등장 개수

$f_{i,j}$: 문서 j에서 keyword i가 나타난 빈도

(1) TF-IDF를 이용하여 d_j의 content를 표현

- TF-IDF(term-frequency/inverse document frequency)에서의 TF는 k_i의 d_j에서의 정규화 빈도를 표현한다. 하지만 해당 용어가 전반적으로 모든 문서에서 자주 나오는 용어라면 k_i는 d_j를 대표한다고 보기 어렵다. 따라서 전체 문서 중 d_j에서 얼마나 나왔는지를 반영해주기 위해 IDF를 사용하여 최종적으로 weight를 계산한다.

$$TF_{i.j} = \frac{f_{i,j}}{\max_z f_{z,j}},\ IDF_i = \log\frac{N}{n_i}$$

$$w_{i.j} = TF_{i,j} \times IDF_i$$

-> $Content(d_j) = \overrightarrow{w_{d_j}} = (w_{1j}, ..., w_{kj})$

-> $ContentBasedProfile(c) = \overrightarrow{w_c} = (w_{c1}, ..., w_{ck})$

-> $u(c,s) = score(ContentBasedProfile(c), Content(s))$

- w_{c1} : 해당 customer에게 측정된 keyword 1의 weight
- *ContentBasedProfile*을 계산하기 위해 각 content vectors의 평균을 사용

(2) 다양한 방법으로 utility function 정의

1) heuristic method

- **cosine similarity**

$$u(c,s) = \cos(\overrightarrow{w_c}, \overrightarrow{w_s}) = \frac{\overrightarrow{w_c} \bullet \overrightarrow{w_s}}{\|\overrightarrow{w_c}\|_2 \times \|\overrightarrow{w_s}\|_2}$$

$$= \frac{\sum_{i=1}^{K} w_i, w_{i.s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2}\sqrt{\sum_{i=1}^{K} w_{i,s}^2}}$$

2) model method(Bayesian classifier and various machine learning techniques(clustering/decision trees, artificial neural networks)**)**

- **Naive Bayesian classifier**

$P(C_i|k_{1,j}\&...\&k_{n,j})$: 해당 d_j에서 keyword 1~n까지 나왔을 때, C_i일 확률

- C_i : relevant or irrelevant

이 때, keyword는 모두 독립을 가정하기 때문에 $P(C_i)\Pi_x P(k_{x,j}|C_i)$을 최대로 하는 C_i를 찾으면 된다.(각 class의 분모는 모두 동일하기 때문에 계산에서 제외)

▶ **Content-Based Methods의 한계**

(1) Limited Content Analysis

1) Content-Based-Methods는 features에 의해 제한된다. text-based items과 같은 feature의 추출이 용이한 것도 있지만, multimedia data, graphical images, audio streams, video streams와 같이 feature 추출이 용이하지 않은 것도 있다.

2) 만약 두개의 다른 items이 똑같은 set of features를 가지고 있다면 두개를 구분할 수 없다. 만약 한 개는 좋은 item이고, 다른 한 개는 나쁜 item일지라도 구분할 수 없다.

(2) overspecialization

- user는 이미 좋게 평점을 매긴 것과 비슷한 items들만을 추천받을 것이다. 즉 추천의 *diversity*가 결여된다.

(3) New User Problem

- user가 충분한 수의 items에 평점을 매겨야 해당 user의 선호를 잘 알 수 있는데, new user의 경우 평점을 매긴 items이 별로 없기 때문에 적절한 추천을 할 수 없다.

● **Collaborative Methods**

- *other users(*특정 user와 비슷한)에 의해 이전에 평점이 매겨진 items에 기초해서 특정 user의 items의 utility를 예측한다.

1) Memory-based algorithms(heuristic method)

$r_{c,s} = agg_{c' \in \hat{C}} r_{c',s}$

$\hat{C}$: N명의 가장 비슷한 다른 users

$k = 1/\sum_{c' \in \hat{C}} |\mathrm{sim}(\mathrm{c},\mathrm{c}')|$, (normalizing factor)

(a) simple average : $r_{c,s} = \frac{1}{N}\sum_{c' \in \hat{C}} r_{c',s}$ (각 rating의 평균)

(b) weighted sum : $k\sum_{c' \in \hat{C}} \mathrm{sim}(\mathrm{c},\mathrm{c}') \times \mathrm{r}_{\mathrm{c}',\mathrm{s}}$ (user간의 similarity에 따른 weighted sum)

- 한계점 : 각 user마다 평점의 scale을 다르게 줄 수 있음(전반적으로 평점을 잘 주는 user가 있을 수도 있고, 전반적으로 평점을 잘 주지 않는 user가 있을 수도 있음)

(c) adjusted weighted sum : $r_{c,s} = \overline{r_c} + k\sum_{c' \in \hat{C}} \mathrm{sim}(\mathrm{c},\mathrm{c}') \times (\mathrm{r}_{\mathrm{c}',\mathrm{s}} - \overline{\mathrm{r}_{\mathrm{c}'}})$

- 원래는 이 정도의 평점을 주는 user인데 이번엔 얼마의 평점을 주었는지를 weighted sum하여 최종 rating을 predict

$r_c = (1/|S_c|)\sum_{s \in S} r_{c,s}$, $where\ S_c = \{\mathrm{s} \in \mathrm{S} | \mathrm{r}_{\mathrm{c},\mathrm{s}} \neq \oslash\}$

- r_c : 해당 user가 보통 주는 평점의 scale

▶ $\mathrm{sim}(\mathrm{x},\mathrm{y})$**를 구하는 방법**

- S_{xy}를 통해 similarity를 계산

$S_{xy} = \{s \in S | r_{x,s} \neq \varnothing \& r_{y,s} \neq \varnothing\}$ (user x, y가 모두 평점을 매긴 item의 집합)

(a) the Pearson correlation coefficient : $= \frac{\sum_{S \in S_{xy}} (r_{x,s} - \overline{r_x})(r_{y,s} - \overline{r_y})}{\sum_{S \in S_{xy}} (r_{x,s} - \overline{r_x})^2 \sum_{S \in S_{xy}} (r_{y,s} - \overline{r_y})^2}$

(b) cosine-similarity

(c) mean squared difference

- 대부분의 방법은 미리 similarity를 계산해놓고 필요할 때 가져다 사용한다.($\hat{C}$가 단기간에 드라마틱하게 바뀔 때 다시 계산)

- 개선 방법

(a) default voting에선 missing value에 대한 몇 개의 default rating value를 추정함으로써 성능을 높이고자 했다.(S_{xy}의 개수가 많지 않으면 정확도가 떨어지기 때문)

(b) Collaborative methods에서 user간의 similarity를 사용하여 계산하는 것이 아닌 *items*간의 similarity를 사용하여 계산하면 더 좋은 컴퓨팅 성능과 결과가 가능하다.

2) Model-based algorithms(statistic method)

- 경험적으로 content-based algorithms보다 성능이 좋음
- 어떠한 content 종류이던지 다룰 수 있음
- feature extraction techniques와 함께 자주 사용됨

ex)

$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)$

- 0과 n 사이의 정수로 rating values가 추정됨
- probability를 추정하는 두 가지 방법 : cluster models and Bayesian networks(하나의 클러스터에만 할당하여, 동시에 몇 개의 클러스터에 할당함으로써 얻는 이익을 누리지 못한다)

▶ 사용되는 Model의 종류

- K-means clustering, Gibbs sampling
- relational model, linear regression, maximum entropy model
- Markov decision process
- probabilistic latent semantic analysis
- Matrix Factorization – SVD, PCA
- Association rule – Shopping basket analysis

▶ Collaborative Methods 의 성능을 향상시키는 접근

- input data set 관련 : 노이즈, 중복, sparse함을 극복, input section techniques 사용
- memory based와의 결합 : 각 user의 선호 / 저장된 user profiles을 함께 사용

▶ Collaborative Methods 의 한계점

(1) New User Problem : hybrid methods 혹은 다양한 전략(item 인기/ item entropy / user 기호 등에 기초한)으로 극복하려고 하고 있다.
(2) New item Problem : user의 선호에 기반하여 추천하기 때문에 충분한 user 수에 의해 평점이 매겨지지 않으면 해당 item을 추천하지 않을 것이다.(hybrid methods로 극복 가능)
(3) Sparsity
(a) 평점을 매긴 수가 부족한 문제
(b) 평점을 매긴 user의 수에 기초하기 때문에 아무리 높게 평점이 매겨진 것이라도, 그 수가 얼마 되지 않는다면 추천이 이루어지지 않을 것이다.
(c) 취향이 일반적이지 않다면 추천 성능이 떨어질 것이다.
<극복방안>
(a) demographic filtering : user profile information을 이용
(b) associative retrieval framework and related spreading activation algorithms을 적용
(c) SVD와 같은 차원 축소 기법을 사용

- **Hybrid Methods**

- 각각의 methods의 한계를 극복하기 위해 collaborative and content-based methods를 합친 방법
- 각각을 합치는 방법에 따라 4가지로 나뉨

1) Combining Separate Recommenders
- cb(content-based)와 cf(collaborative filtering)를 따로 실행하고, 예측할 때 두 결과를 혼합해서 사용한다.

(1) 각각의 output을 합쳐서 하나의 최종 추천 모델(linear combination of ratings/voting scheme)을 사용하여 최종 예측한다.
(2)몇몇의 추천의 "quality" metric(ex confidence)에 기반하여 둘 중 더 좋은 하나의 추천을 선택한다.

2) Adding Content-Based Characteristics to Collaborative Models
- cf에 기반하면서도 각 user의 cb를 사용하는 기법

▶ 장점
(1) some sparsity-related problems를 극복할 수 있다.
(2) user가 좋아한 content를 기반으로 추천할 수도 있고, user와 비슷하게 추천한 다른 user를 기반으로 추천할 수도 있으므로 추천의 정확성이 올라감

3) Adding Collaborative Characteristics to Content-Based Models
- cb에 기반하면서도 cf를 사용하는 기법
- dimensionality reduction 기법(ex svd)을 사용할 수 있다.

4) Developing a Single Unifying Recommendation Model
- cb와 cf의 방법을 결합한 하나의 단일 모델을 개발

(1) single rule-based classifier

(2) probabilistic latent semantic analysis

(3) Bayesian mixed-effects regression models

- Markov chain Monte Carlo methods for parameter($\mu, \sigma^2, \Lambda, \Gamma$) estimation and prediction
- z_i와 w_j는 SVD와 같은 방식으로 item 기반, user 기반 matrix를 형성하여 해당 벡터들을 추출했다고 볼 수 있음

$r_{ij} = x_{ij}\mu + z_i\gamma_j + w_j\lambda_i + e_{i,j},$
$e_{ij} \sim N(0, \sigma^2),$
$\lambda_i \sim N(0, \Lambda),$
$\gamma_j \sim N(0, \Gamma),$

- 파라미터

z_i : user attributes

w_j : item attributes

x_{ij} : user attributes와 item attributes의 상호작용(item and user attributes를 결합한 것)

e_{ij} : noise

λ_i : unobserved sources of user heterogeneity

γ_j : unobserved sources of item heterogeneity

(4) Knowledge-based techniques

- case-based reasoning과 같은 기법이 있음
  - case-based reasoning : 여러 case를 생성하고 점점 맞춰가는 기법
- new user, new item problems와 같은 전통적인 추천 시스템의 문제를 해결하고 정확도를 개선할 수 있음
- knowledge를 얻어야 할 필요성이 있다는 문제가 있지만, domain knowledge가 쉽사리 구조화된 machine-readable form으로 가능한 application domain에서는 발전되어 오고 있음

- **Summary**

- 추천시스템에는 cb, cf, hybrid methods가 존재하고, rating estimation하는 방법으로 memory based/model based 방법이 있다. 다양한 종류의 추천 어플리케이션을 지원하기 위해 traditional memory-based 방법에서 contextual information을 고려한 방법으로 확장되고 있다.