Model-based CF

이혜승

Contents

Introduction of Model-based CF

Introduction of Model-based CF

- Supervised, unsupervised machine learning methods와 같이 **미리** 모델을 생성.

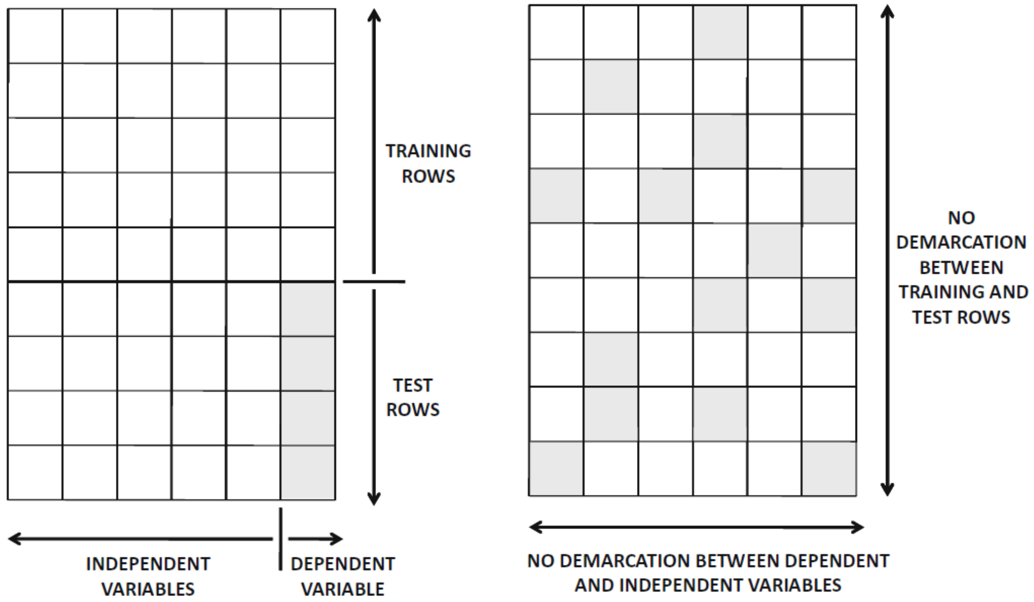
→ 트레이닝 / model building 단계와 예측 단계 분리.

- (a) 분류 문제: m x n matrix가 주어짐

- Feature와 class 변수의 구분 O
- 트레이닝 셋과 테스트 셋의 구분 O

- (b) Collaborative filtering:

- 각각의 열은 독립변수와 종속변수를 모두 포함(뚜렷한 구분 x)
- Rows로 트레이닝 / 테스트 셋 구분 X. 값이 있는 요소는 트레이닝, 없는 요소는 테스트 데이터로 판단



(a) Classification (b) Collaborative filtering

Introduction of Model-based CF

- Neighborhood-based methods와 비교한 장점들

1) Space efficiency

   : ratings matrix (m x n)보다 공간을 적게 차지한다.

2) Training and prediction speed

   : 사전처리속도, 예측속도 빠름(compact & summarized model)

3) Avoiding overfitting

   : overfitting 문제는 summarization approach + regularization을 통해 극복 가능

Decision and Regression Trees

Decision and Regression Trees

1. Decision tree를 분류 문제에 적용

- 분할 기준(Split criteria)을 사용하여 data space를 계층적으로 분할.

→ Feature variable에 대해 0 / 1값을 가지면, 각각 같은 branch에 해당됨.

→ 계속 분류되면서 branch가 pure해짐.(같은 클래스에 속하는 데이터끼리 분리)

- Gini index(지니 계수): $G(S) = 1 - \sum_{i=1}^{r} p_i^2$

Split이 얼마나 잘 되었는지 알려주는 지표.(값이 작을수록 같은 클래스끼리 분리가 잘 된 것.)

목적: 적절한(제일 pure하게 분류되는) 분류 기준을 찾기 위함

- 그 외 split criterion

- variance: numeric 변수에 적합
- Error rate
- entropy

Decision and Regression Trees

2. Decision tree를 CF에 적용

- Issues

1) ratings matrix sparsity 문제

→ 임계값보다 크면 left node, 작으면 right node. 만약 missing value이면?

→ 1) 양 branch 에 넣어 줌(경로가 unique하지 않아 충돌 가능성) 2)차원 축소

2) predicted / observed 값들이 column-wise하게 구분x

3) 독립, 종속변수 명확히 구분x

→ 그렇다면 어떤 아이템을 이 decision tree로 예측해야 할까?

→ (solution) 각 item들의 ratings 예측을 위한 decision tree를 별개로 생성

: 해당 item을 종속변수(y), 남은 items를 독립변수(X)로 고정

: decision tree의 개수 = item의 개수 = n

Decision and Regression Trees

2. Decision tree를 CF에 적용

- 1) j번째 item을 예측하고자 함 → R = **m x n**에서 j번째 열 제거, 나머지 열은 독립변수(X)로 생각.
- 2) n – 1 feature items의 공분산 행렬 생성
- 3) m x (n-1) → m x d로 축소: decision tree를 트레이닝하는 매트릭스(**m x d 행렬** 생성)

- I열의 고유 벡터에 있는 I_u의 rated item j 와 j에 대한 사용자 u의 ratings → 평균 기여도 계산.
- 평균 기여도는 1 ~ j-1까지의 기여도 합 / 항목 수 . 각 고유 벡터에 대한 각 사용자의 평균 기여도를 찾아서 m x d 행렬을 얻는다.

- 4) j ={1, . . .n}에 대해서 반복. (지금 만든 것은 item j의 rating만 예측할 수 있음)

Rule-Based Collaborative Filtering

Rule-Based Collaborative Filtering

Association rules(연관 규칙)

- Transaction database: $T = \{ T_1, \cdots, T_m \}$
- I: item n개에 대한 전체 집합

- **Support(지지도):** 전체 상품 구입 데이터 중 X 상품이 구입된 수

$$supp(X) = \frac{|X \subset \mathrm{I}|}{|T|} = P(X)$$

: frequency item sets를 구할 때 사용

- **Confidence(신뢰도):** X를 구매했을 때, Y도 같이 구매할 확률(조건부 확률로 정의)

$$conf(X \Rightarrow \mathrm{Y}) = \frac{\mathrm{supp(X \cup Y)}}{supp(X)} = P(Y|X)$$

:아이템 집합 간의 연관성 강도 측정.

Rule-Based Collaborative Filtering

Association rules(연관 규칙)

- **Association rules(연관규칙)**

: $support \geq minimum\ support\ s\ \&\&\ confidence \geq minimum\ confidence\ c$

지지도와 신뢰도의 값이 지정한 임계치 s, c를 넘을 때 유용하다고 판단.

Association rules를 찾는 2 steps

- (1) frequent item sets 결정

: $support \geq minimum\ support\ s$

- (2) (X, Z-X)로 X->Z-X라는 잠재적인 규칙을 생성 → 최소 confidence를 충족하는 것만 남긴다.

: $confidence \geq minimum\ confidence\ c$

Rule-Based Collaborative Filtering

Association rules를 CF에 적용

- Unary ratings matrix 상황 # item-wise

: Like표현→ 1로 표현, dislike 표현 불가 / 누락된 값은 0으로 설정

- 추천 대상 user A (이 규칙들은 모델. 특정 사용자에 대해 추천할 때만 사용가능.)
- Step1. 사전에 정한 minimum support *s*, minimum confidence *c* 수준에서 가능한, user A로 부터 시작된 모든 association rules를 찾는다. (s, c는 accuracy를 최대화하기 위한 parameter)

  즉, rules의 선행 항목에 있는 itemset가 해당 user가 선호하는 항목 중 하나.

- Step2. 모든 association rules는 confidence를 내림차 순으로 정렬한다.

→ first-k 아이템들은 user A에게 top-k items로 추천된다. (confidence 가 높은 k개의 아이템 추천)

Naïve Bayes Collaborative Filtering

Naïve Bayes Collaborative Filtering

- Ratings 값: $v_1 \sim v_l$
- u-th user

  I_u : $user\ u$가 ratings 남긴 itemset

  r_{uj} : $user\ u$가 item j에 남긴 점수. r_{uj}는 I_u로 예측 가능

  $s = \{1,2,..,l\}$ 일 때,

$$P\left(r_{uj} = v_s \middle| Observed\ ratings\ in\ I_u\right) \propto P(r_{uj} = v_s) \cdot \prod_{k \in I_u} P\left(r_{uk} \middle| r_{uj} = v_s\right)$$

- 각 v_s 에 대한 확률을 계산했다면, r_{uj} 값은 어떻게 예측할 것인지?

Naïve Bayes Collaborative Filtering

- $\widehat{r_{uj}}$추정 방식

1) $s = \{1, 2, .., l\}$에 대해 모두 계산한 후, 가장 큰 확률 값을 갖는 v_s로 추정.

$$\widehat{r_{uj}} = argmax_{v_s} P\left(r_{uj} = v_s \middle| Observed\ ratings\ in\ I_u\right)$$

: 등급 개수 l이 작을 때, 주로 사용

2) 계산된 모든 ratings의 가중 평균

$$\widehat{r_{uj}} = \frac{\sum_{s=1}^{l} v_s \cdot P\left(r_{uj} = v_s \middle| Observed\ ratings\ in\ I_u\right)}{\sum_{s=1}^{l} P\left(r_{uj} = v_s \middle| Observed\ ratings\ in\ I_u\right)}$$

: 등급이 세분화되어 있을 때, 주로 사용

Q) 정확한 의미?

Naïve Bayes Collaborative Filtering

Example)

Target user: user 3

$v_1, v_2 = \{-1, 1\}$

목표: $r_{3,1}$과 $r_{3,6}$ 값의 확률 계산 → 예측

Table 3.2: Illustration of the Bayes method with a binary ratings matrix

Item-Id ⇒ User-Id ⇓	1	2	3	4	5	6
1	1	-1	1	-1	1	-1
2	1	1	?	-1	-1	-1
3	1	1	1	-1	-1	-1
4	-1	-1	-1	1	1	1
5	-1	?	-1	1	1	1

$P(r_{31} = 1 | r_{32,} r_{33}, r_{34}, r_{35}) \propto P(r_{31} = 1) \cdot P(r_{32} = 1 | r_{31} = 1) \cdot P(r_{33} = 1 | r_{31} = 1) \cdot P(r_{34} = 1 | r_{31} = 1) \cdot P(r_{35} = 1 | r_{31} = 1$

$P(r_{31} = 1 | r_{32,} r_{33}, r_{34}, r_{35}) > P(r_{31} = -1 | r_{32,} r_{33}, r_{34}, r_{35})$이므로, r_{31}값은 1로 예측한다.

같은 방식으로 r_{36}을 구하면, -1로 예측.

결론적으로 user 3에게 item1(top-1 item)을 추천해준다.

Naïve Bayes Collaborative Filtering

Overfitting 극복 방법

- Challenge

기존의 ratings matrix가 sparse하면,

우리의 예측 방식이 not robust해진다.

- Solution

Laplacian smoothing: 실제로 관찰한 것보다 α 번씩 더 관찰했음을 가정하는 방법

$$P(r_{uj} = v_s) = \frac{q_s + \alpha}{\sum_{t=1}^{l} q_t + l \cdot \alpha}$$

α: smoothing의 정도를 조절하는 parameter.

α ↑ smoothing ↑ → data에 둔감해질 수 있다.

- Laplacian smoothin의 활용 이유: 확률 값이 0 혹은 0에 가까운 값이 나오면, 곱해져서 나온 최종 확률값이 0으로 단정되어, 학습에의 어려움이 있음. Padding의 목적.

Q n A