

CH5. KNOWLEDGE-BASED RECOMMENDER SYSTEMS

5.3 Case-Based Recommenders

- In case-based recommenders, similarity metrics are used to retrieve examples that are similar to the specified targets (or cases).

Table 5.2: Examples of attributes in a recommendation application for buying homes

Item-Id	Beds.	Baths.	Locality	Type	Floor Area	Price
1	3	2	Bronx	Townhouse	1600	220,000
2	5	2.5	Chappaqua	Split-level	3600	973,000
3	4	2	Yorktown	Ranch	2600	630,000
4	2	1.5	Yorktown	Condo	1500	220,000
5	4	2	Ossining	Colonial	2700	430,000

- 예를 들어, 정확하게 유저의 요구를 만족하는 집이 없다고 하면, similarity function으로 유저의 query와 가장 비슷한 후보들을 수집하여 순위를 매길 수 있다. 때문에 constraint-based recommenders와 다르게 empty sets 문제는 case-based recommenders에서는 큰 이슈가 되지 않는다.
- 최초의 case-based systems는 user query requirements를 적합한 solution이 찾아질때 까지 반복적으로 수정하는 방법을 사용하였다. 이후 **critiquing** 방법이 개발되었다. 이 critiquing 방법의 아이디어는 user가 retrieved results를 하나 이상 선택할 수 있고 나아가 다음 형태와 같이 몇 가지 queries를 구체화 할 수 있게된다.
 - "Z에 따라서 특성 Y가 따르고 X와 유사한 items을 더 보여줘"
- critiquing 의 주 목적은 item space를 interactive browsing하는 것이다. 그러면서 users는 수집되는 examples를 통해 가능한 옵션들을 알 수 있게된다. Interactive browsing of item space는 users가 iterative하게 query를 하는 과정 속에서 item space에 대한 학습이 가능하다는 이점이 있다.
- The specification of target features vs. the specification of target address

EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[ENTRY POINT]

I WOULD LIKE TO BUY A HOUSE SIMILAR TO ONE WITH THE FOLLOWING FEATURES:

NUMBER OF BR

NUMBER OF BATH

HOME STYLE

PRICE RANGE

ZIP CODE

SUBMIT SEARCH

I WOULD LIKE TO BUY AN HOUSE JUST LIKE THE ONE AT THE FOLLOWING ADDRESS:

812 SCENIC DRIVE

MOHEGAN LAKE

NY

SUBMIT SEARCH

- 수집된 결과들에서, user는 마음에 들지 않는 특정 조건이 포함되어 있다는 것을 제외하고 특정 집 에 대한 선호를 결정할 수 있다. 이 경우 users는 이 집을 anchor로 하고 특정 조건을 바꾸어서 좀 더 선호하는 옵션의 집을 검색해볼 수 있다.
- 수정된 target인 새로운 query로 이전 query의 결과보다 줄어든 후보군들을 얻을 수 있을 것이 다. 대부분의 경우 새롭게 순위를 정하는 것이 아니라 관련이 없는 cases들의 검색 결과를 잘라내 는 효과를 가지고 있다. 그래서 constraint-based systems와는 다르게 iterations의 한 cycle 을 거치고 나면 일반적으로 결과가 줄어든다.
- 후보를 더 늘려가는 방향으로 검색을 반복할 수도 있는데, 고려할 수 있는 후보가 많아지지만 그만 큼 관심에 없는 후보들 까지 늘어날 수 있다는 trade-off가 있다. 이 책에서는 후보를 줄여가는 과 정으로 가정한다.
- 처음엔 어느 정도가 적당한 조건인지 모르던 유저는 조금씩 상호작용을 하며 후보를 줄여나가면서 gap을 좁혀갈 수 있다. 그런데 후보를 줄이다 보면 조건을 만족하는 후보가 아예 없는 경우가 있 다. 이는 constraint-based system에서 후보가 없어지는 것과는 다른 의미를 가진다. 그 이유 는 user가 현재 요구 사항에 대해서 relaxing할 수 있는 option을 가지고 있기 때문이다. (아마 조건을 완화할 수 있는데도, 더 좁혀나간거 보면 유저가 확실히 원하는 조건이란 뜻이라서?)
- In order for a case-based recommender system to work effectively, there are two crucial aspects of the system that must be designed effectively:

1. **Similarity metrics** : case-based systems에서 관련 결과를 얻기 위해서 효과적인 similarity metrics를 디자인하는 것은 매우 중요하다. 다양한 특성들의 중요도가 적절하게 similarity function에 적용되어야한다.
2. **Critiquing methods** : interactive exploration of the item space는 critiquing methods와 함께 사용된다. 다양한 critiquing methods 들은 다른 exploration goals를 지원한다.

5.3.1 Similarity Metrics

- d개의 attributes를 가지는 상품을 다룬다고 가정해보자. 우리는 두 partial attribute vectors 사이의 유사도를 계산하고 싶을 것이다. 이때 이 두 partial attribute vectors는 d개의 attributes에서 s 개의 subset으로 정의된다(i.e., $|S| = s \leq d$).
 $X = (x_1, \dots, x_d), T = (t_1, \dots, t_d)$ 로 각각 d-dim vector라고 하자 이때 T는 target 이 된다.
- 여기서 우리는 partial attribute vectors를 사용한다. 왜냐하면 대부분의 경우 user에 의해서 구체화 되는 attributes는 small subset이기 때문이다.

$$f(\bar{T}, \bar{X}) = \frac{\sum_{i \in S} w_i \cdot \text{Sim}(t_i, x_i)}{\sum_{i \in S} w_i}$$

- An example of a symmetric metric is as follows:

$$\text{Sim}(t_i, x_i) = 1 - \frac{|t_i - x_i|}{\max_i - \min_i}$$

- the similarity function with the standard deviation σ_i (3 곱하면 99% 0~8, 6sigma 는 -8 ~ 8)

$$\text{Sim}(t_i, x_i) = \max\{0, 1 - \frac{|t_i - x_i|}{3 \cdot \sigma_i}\}$$

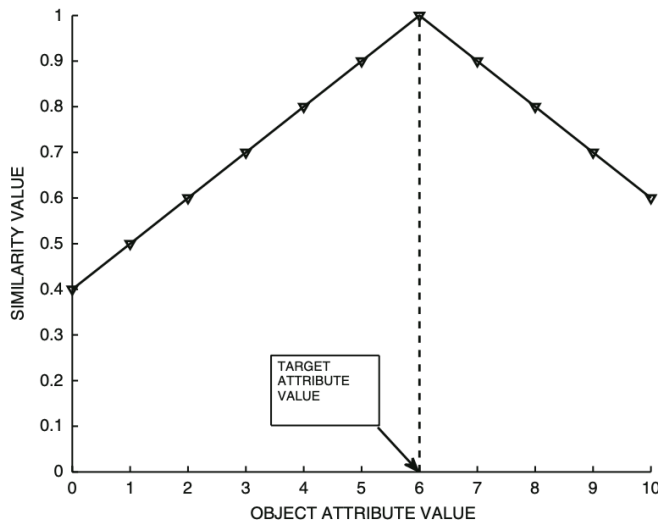
- For the case of attributes in which larger values are better, an example of a possible similarity function is as follows: (어떤 특징 i가 target 보다 크게 더 좋은 경우 (e.g., camera resolution))

$$\text{Sim}(t_i, x_i) = 1 - \frac{|t_i - x_i|}{\max_i - \min_i} + \underbrace{\alpha_i \cdot I(x_i > t_i) \cdot \frac{|t_i - x_i|}{\max_i - \min_i}}_{\text{Asymmetric reward}}$$

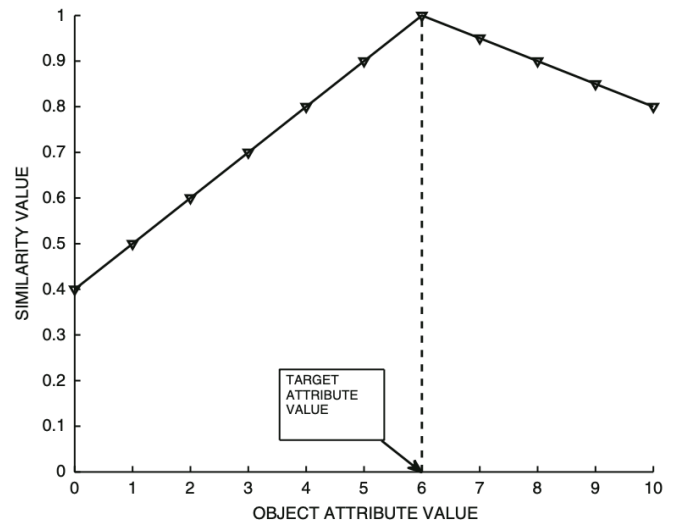
- For cases in which smaller values are better (e.g., price), the reward function is similar, except that smaller values are rewarded by the indicator function: (어떤 특징 i가 target 보다 낮은게 더 좋은 경우)

$$Sim(t_i, x_i) = 1 - \frac{|t_i - x_i|}{max_i - min_i} + \underbrace{\alpha_i \cdot I(x_i < t_i) \cdot \frac{|t_i - x_i|}{max_i - min_i}}_{\text{Asymmetric reward}}$$

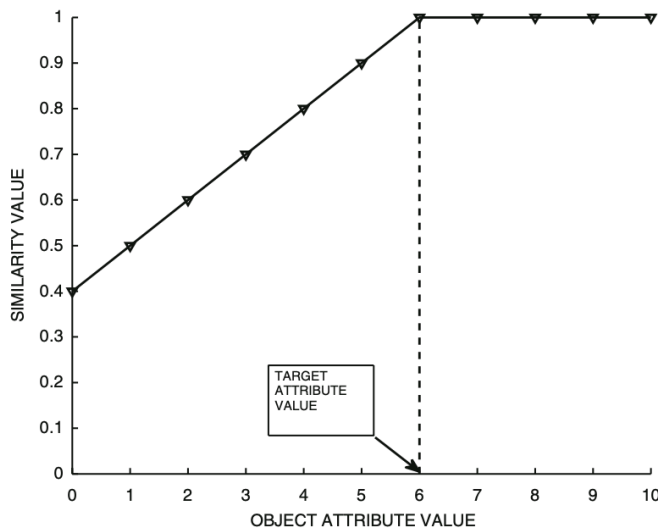
- $\alpha_i > 1$ 이 되면 실질적인 "similarity" 는 target 과의 거리가 멀어질 수록 큰 값으로 계산된다. 이런 경우 $Sim(t_i, x_i)$ 는 similarity function 보다는 *utility function*에 더 가깝다.
- $\alpha_i = 1$ 이면 특정 attribute가 target 보다 크거나 작기만 하면 된다는 의미이고
- α_i 가 $\in (0, 1)$ 이면 target과 비슷한 범위 안에서 크거나, 작은 item을 선호한다는 의미이다. (b)의 경우 자동차 마력이라 생각해보면 클 수록 좋겠지만 마냥 커버리면 연비가 떨어지는 단점이 있기때문에 (b)가 적합하다.



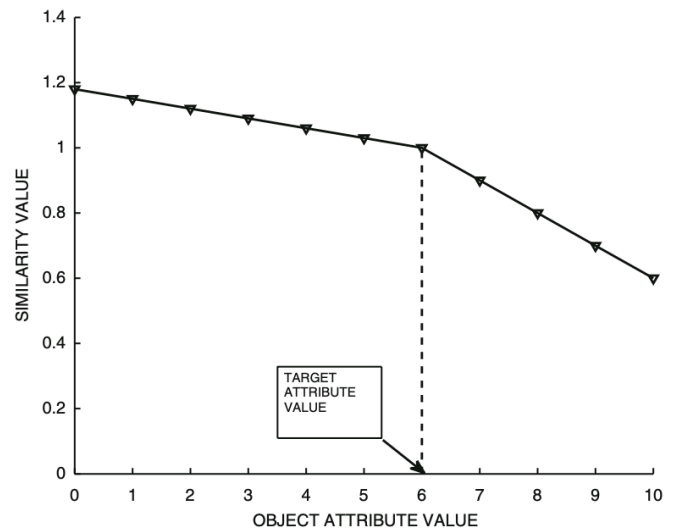
(a) Symmetric ($\alpha_i = 0$)
(penalty by absolute distance)



(b) Asymmetric ($\alpha_i = 0.5$)
(milder penalty for overshooting)



(a) Asymmetric ($\alpha_i = 1.0$)
(no penalty for overshooting)



(b) Asymmetric ($\alpha_i = 1.3$)
(less is always better)

Figure 5.6: Examples of different types of symmetric and asymmetric similarity

- categorical data의 경우, similarity values를 결정하는게 더 힘들다.

- 전문가가 나누거나, user에 의해서 나누어 질 수 있다.
- various attributes에서 특성별 중요도를 고려하는 것도 similarity function을 디자인하는데 중요한 issue이다. 상대적 중요성은 parameter w_i 에 의해서 조절된다.
- 실험적으로 domain 전문가가 w 를 설정할 수 있다. 또는 user의 feedback을 통해 학습 가능하다.