7 Evaluating Recommender Systemts

7.3 General Goals of Evaluation Design

- 정확도 뿐만 아니라 다양성, 우연성, 새로움, 강건함, 확장성 등을 평가해야합니다.
- 구체적인 정량화가 가능한 것도 있고, user 경험에 기반한 것도 있다. 이런 경우 설문조사를 해야합니다.

7.3.1 ~ 7.3.6 Accuracy, Coverage, Confidence and Trust, Novelty, Serendipity, Diversity

7.3.7 Robustness and Stability

- 추천 시스템이 가짜 평점이나 특히 시간에 따라 이런 패턴이 증가할때 큰 영향을 받지 않는 것을 stable하고 robust하다고 합니다.
- 예를 들어 책을 출간한 작가나 출판사에서 의도적으로 좋은 평을 남기거나, 경쟁사의 제품에 악플 테러를 할 수도 있습니다.
- *Attack models for recommender systems* 는 12장에서 더 자세히 이야기 나누겠습니다.

7.3.8 Scalability

- 최근에는 대용량의 사용자 평점이나 내재적 피드백 정보 등 수집이 가능해지면서 시간이 지남에 따라 데이터 셋의 크기가 증가하는 추세입니다.
- 이런 대량의 데이터를 효과적이고 효율적으로 처리하는 방법은 추천 시스템에서 핵심적인 모델 디자인 요소가 되었습니다.
    - Training time : 모델을 학습하는 과정입니다. neighborhood based collaborative filtering 알고리즘은 미리 user의 peer group을 계산해둬야 할 것입니다. 그리고 matrix factorization system의 경우 미리 latent factors를 구해야할 것입니다. 이렇게 학습 시간에 소요되는 시간 자체가 모델 학습시 평가 요소가 될 수 있습니다. 물론 대부분의 학습은 offline으로 진행되기 때문에 학습에 걸리는 시간에 대해서는 빡빡하게 평가하진 않습니다.
    - Prediction time : 실전에 배치되어 예측을 하는 단계는 빠를 수록 좋습니다. 왜냐하면 모델의 판단이 느려질 수록 사용자가 기다려야하는 시간이 늘어나기 때문입니다.
    - Memory requirements : rating matrix가 커지면 전체 matrix를 메인 메모리에 올려두기 힘들게 됩니다. 이때 요구되는 메모리 양을 최소화하는 알고리즘을 개발하는 것이 필요합니다.

7.4 Design Issues in Offline Recommender Evaluation

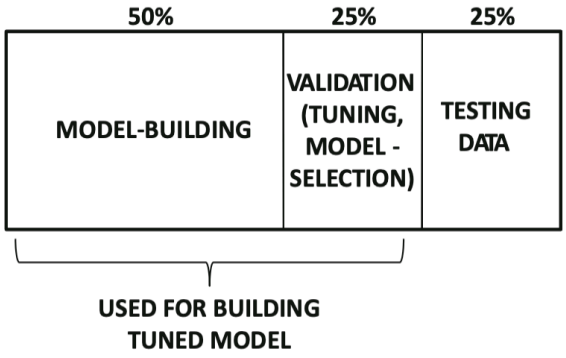
- 이 파트에서는 추천시스템 평가 디자인을 하는데 있어 존재하는 이슈들을 다루게 됩니다.
- 추천시스템을 평가할때 정확도가 고평가되거나 저평가되지 않도록 주의해서 디자인해야합니다.
- 예를 들어 train과 evaluation할때 같은 데이터를 사용하면 안됩니다. 이렇게 되면 알고리즘이 과대평가 될 수 있습니다.
- rating matrix는 *entry-wise* 로 sampling됩니다. 일정 부분을 학습용으로 뽑고 나머지를 평가용으로 사용합니다.
- 이러한 데이터셋을 분리하는 방법은 분류나 회귀 문제에서도 많이 사용됩니다. 하지만 분류/회귀에서와는 다른 점이 있는데, 분류/회귀에서는 데이터셋을 row 방향으로 sampling이 됩니다. 하지만 rating matrix에서는 entry 별로 sampling이 된다는 차이가 있습니다.
- 또 일반적으로 많이 실수하는 부분은 파라미터를 튜닝하는 데이터와 테스트를 위한 데이터가 같다는 것입니다. 이러한 접근 방법 역시 과대평가/오버피팅될 수 있습니다.
- 따라서 이를 막기 위해 일반적으로 다음과 같이 데이터셋을 세 파트로 나누어 사용합니다.
    1. Training data

모델을 학습하기 위해 사용됩니다. 또 이 데이터는 여러개의 모델을 만들고 마지막에 하나의 모델을 선택하기 위해 사용되기도 합니다.(?)

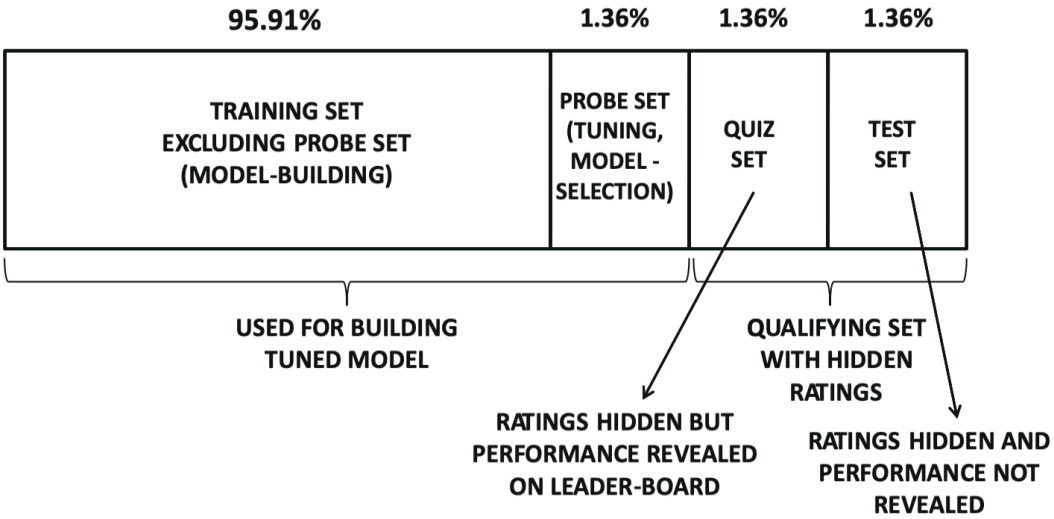
2. Validation data
   이 데이터는 모델을 결정하거나 파라미터를 튜닝하기 위해 사용됩니다.
3. Testing data
   이 데이터는 가장 마지막에 딱 한번 사용됩니다. 만약 이 데이터로 평가한 결과를 가지고 모델을 어떤 식으로든 수정하게 되면 그 결과는 테스트 데이터에서 얻은 *knowledge* 로 *contaminated* 된게 됩니다.

- Validation data는 training data의 일부분에서 떨어져 나오기도 합니다.
- 보통 2:1:1의 비율로 나누어 줍니다. 아니면 전체 데이터의 절반을 학습에 사용하고 남은 절반의 반씩을 model-selection과 testing을 위해 사용합니다.
- 최근에 데이터의 크기가 커지면서 더 적은 비율로 validation과 testing 데이터를 확보하기도 합니다.

7.4.1 Case Study of the Netflix Prize Data Set



(a) Proportional division of ratings



(b) Division in Netflix Prize data set (not drawn to scale)

- 사실 위 그림이 전부 입니다.
- 넷플릭스 데이터 전체 중 97.27%가 모델 빌드와 probe data(모델 튜닝, 선택)로 제공되었는데, 약 95.91%가 학습을 위해, 1.36%가 모델 튜닝을 위해 사용되었습니다.
- Probe set는 validation set과 매우 유사한 역할로 사용되었습니다.
- 다른 참가자들은 probe set을 다양한 방식으로 사용하였는데, 특히 probe set이 비교적 최근 데이터이기 때문에 training 과 probe sets의 평점의 통계적 분포가 조금 달랐습니다.
- 앙상블 메소드의 경우, probe set은 다양한 앙상블 컴포넌트들의 가중치를 부여하기 위해 사용되었습니다.
- training set과 probe set의 통계적 분포가 다르다고 했었습니다. 오히려 probe set은 *qualifying set* 의 통계적 특성을

더 반영하고 있었습니다. 왜냐하면 training data보다 probe, qualifying set으로 갈 수록 더 최근 데이터를 담고 있기 때문입니다.

- 나머지 2.7%의 ratings는 제공되지 않습니다. 단 <User, Movie, GradeDate> 만 제공됩니다. 보통의 test set과의 차이점은 참가자들이 qualifying set에 대한 성능을 예측 값을 제출한뒤 리더보드를 통해 확인할 수 있다는 것입니다. 이때 사용되는 데이터가 qualifying set 의 절반으로 quiz set이라 합니다.
- quiz set을 통해 경쟁자들과 자신의 모델의 성능을 비교하고 상대적 위치를 확인할 수 있습니다.
- 마지막으로 quiz set이 아닌 나머지 qualifying set의 데이터는 test set으로 마지막 최종 모델 평가를 위해 사용됩니다.
- 참여자들은 quiz set을 통해 모델에 대한 상대적 수준만 확인할 수 있을뿐 test set과의 연관성을 찾을 순 없습니다.

7.4.2 Segmenting the Ratings for Training and Testing

- real data는 training, validation, test data 이렇게 삼등분되어 있지 않습니다. 그래서 자동으로 이렇게 데이터를 삼등분할 수 있게 만들어야합니다.

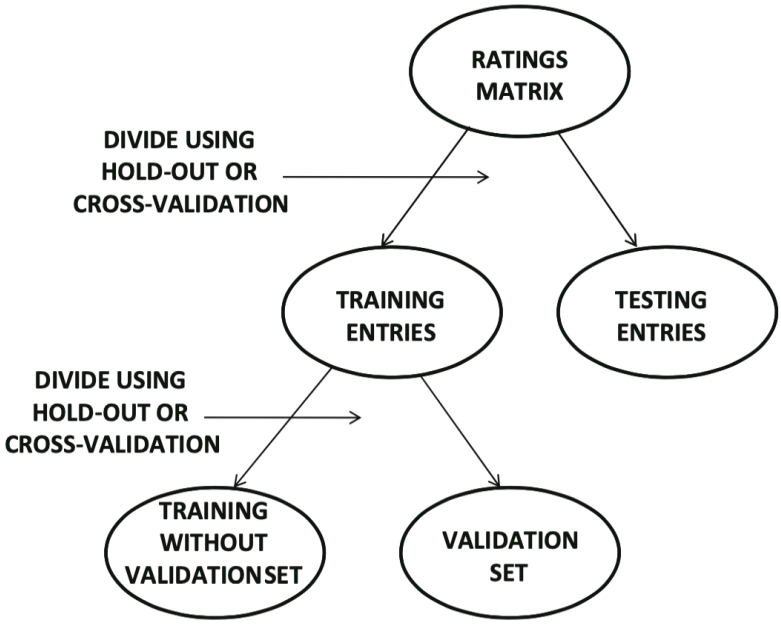


Figure 7.2: Hierarchical division of rated entries into training, validation, and testing portions

7.4.2.1 Hold-Out

- 특정 비율의 ratings가 가려져 있습니다. 그리고 나머지 entries로 학습을 진행합니다. 그리고 숨겨져 있던 entries에 대한 정확도를 모델 평가로 사용합니다.
- 이 방법의 단점은 전체 데이터를 충분히 사용하지 못 한다는 것입니다.
- 그리고 만약 held-out entries와 held-in entries의 분포가 다르다면 모델은 편향되어 정확한 예측을 하지 못할 것입니다.

7.4.2.2 Cross-Validation

- rating entries를 q 개의 동일한 사이즈로 나누어줍니다.
- 만약 S개의 entries set이 있을때, |S|/q 개씩 나누어지게 됩니다.
- q 개의 segments 중 하나는 testing을 위해 사용되고 나머지 q-1 segments는 training을 위해 사용됩니다.
- 각 학습 프로세스동안 |S|/q 개의 entries가 가려져있는 것입니다. 그리고 모델 평가를 위해 사용되는 것입니다.
- 각각 q segments를 test set으로 사용하면서 이 과정을 q 번 반복하게 됩니다.
- 그리고 q 개의 다른 test set에 대한 평균 정확도를 최종 성능으로 하게됩니다.
- 이러한 방법의 특이한 케이스로 q를 전체 데이터 개수로 잡아버리는 경우가 있습니다. 이렇게 되면 |S|-1 개를 학습으로 사

용하고 1나의 테스트 샘플로 평가를 하는 것입니다. 이를 *leave_one_out cross-validation* 이라고 합니다. 학습 시간이 오래 걸립니다.

- 실제론 거의 q는 10으로 고정되어 사용됩니다. 그럼에도 불구하고 leave-one-out cross-validation은 neighborhood-based collaborative filtering algorithm에서 어렵지 않게 사용될 수 있습니다.

7.4.3 Comparsion with Classification Design

- Collaborative filtering에서 평가 디자인은 classification에서 하는 것과 매우 유사합니다.
- collaborative filtering은 classification problem의 일반화된 형태로 이해할 수 있습니다.
- 한 가지 classification design 과 다른점은 hidden entries에 대한 평가가 종종 실제 시스템의 성능을 반영하지 못 한다는 것입니다.
- 왜냐하면 hidden ratings는 matrix에서 random으로 뽑아지지 않기 때문입니다. 오히려, 이 hidden ratings는 소비자가 소비를 하면서 직접 고른 item입니다. 그러므로 이런 entries는 진짜 missing values와 비교했을때 매우 높은 values(값? 가치) 를 가집니다. 이를 *sample selection bias* 문제라고 합니다.
- 분류문제에서도 이런 문제가 있지만 이는 collaborative filtering 접근 방법에서 더 만연하게 나타는 문제입니다.
- 구체적인 내용은 7.6에서 살펴보겠습니다.