Internet Application

<HW #4>

산업공학과 황영석

2010-12086

1. (50점) MovieLens 데이터를 활용하여, 아래의 작업을 수행하시오. (1-1과 1-2의 경우 결과 출력에 사용한 **BasicRecommender.java 파일과, 문제에서 요구하는 txt 파일을 함께** 제출하시오. **1-3에서는 추가로 제시한 유사도 함수에 대하 여 1-1과 1-2에서 요구한 txt 파일을 다시 제출할 필요는 없으며, 결과 비교 및 의 견 서술 내용만을 MS-Word 문서 파일 형태로 작성**하여 제출하시오.)

1-1. (20점) User based recommender를 완성하고, 이를 사용하여 1~10번 user에 대한 상위 10개 item의 추천 결과를 user-based-result.txt 파일로 출력하시오. 이 때, 아래의 유사도 함수를 사용하시오. - Pearson correlation similarity

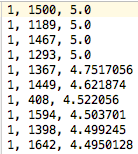
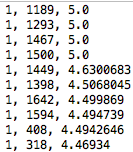
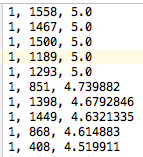
**“user-based-result.txt” file submitted**

1-2. (20점) Item based recommender를 완성하고, 이를 사용하여 1~10번 item에 대한 상위 10개 item의 추천 결과를 item-based-result.txt 파일로 출력하시오. 이 때, 아래의 유사도 함수를 사용하시오. - Pearson correlation similarity

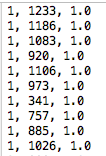
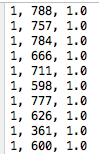
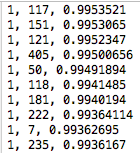
**“item-based-result.txt” file submitted**

1-3. (10점) 아래의 유사도 함수를 사용하여 1-1, 1-2 과정을 반복한 뒤, 그 결과를 서로 비교하시오. 각 유사도 함수를 사용했을 때의 차이점에 대해 본인의 의견을 서술하시오(u.user와 u.item 파일의 정보를 함께 활용 가능).

- Cosine similarity - Log likelihood similarity



UB\_pearson UB\_cosine UB\_log

IB\_Pearson IB\_cosine IB\_log

UB는 target user에게 맞는 아이템을 골라주므로,

IB는 item과 가장 비슷한 아이템을 골라주므로 위와 같은 결곽가 나온 것으로 보인다.

각각의 유사도에 따라 결과가 조금씩 다르게 나왔다. 유사도의 정의에 따라 Top 10이 조금 다르게 ranking될 것이기 때문이다. IB의 경우 장르에 대한 설명인 19가지 필드(0-1값만 가짐) 벡터가 동일한 경우가 많아 top10 전체가 1.0 혹은 거의다 1에 가까운 값을 갖는 경우가 생긴 것 같다.

**Pearson Correlation**: covariance of the features normalized by their standard deviation.

**Cosine similarity**: the cosine of the angle between the two feature vectors.

**Log-Likelihood**: the log of the probability that the item will be recommended given the characteristics you are recommending on.

2. (50점) MovieLens 데이터를 활용하여, 아래의 작업을 수행하시오. (**2-1의 경우 결과 출력에 사용한 MFRecommender.java 파일과, 문제에서 요구하는 txt 파일을 함께 제출**하시오. **2-2와 2-3의 경우 결과 분석 및 의견 서술 내용만을 MS-Word 문서 파일 형태로 작성**하여 제출하시오.)

* 2-1.  (20점) Matrix factorization(MF) based recommender를 제작하고, 이를 사용하여  1~10번 user에 대한 상위 10개 item의 추천 결과를 mf-based-result.txt 파일로  출력하시오.
* **“mf-based-result.txt” file submitted**
* 2-2.  (20점) 2-1의 추천 결과 성능을 아래의 지표를 종합적으로 활용하여 평가하시오. MF에 사용되는 SVD++ factorizer의 feature 수와 iteration 수를 변화시키면서 추천 결과 성능의 변화 양상을 분석하시오.  - AAD(average absolute difference), RMS(root mean squared difference)
* 0) Baseline(10,10) 평가 결과
* ../../../Dropbox/스크린샷/스크린샷%202016-06-05%2014.10.04.png
* 1) Num of Features
* # of iterations =10 (fixed), five times average each

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * # of Features | * 1 | * 2 | * 3 | * 5 | * 10 | * 15 | * 20 | * 100 |
| * AAD | * 0.83 | * 0.82 | * 0.82 | * 0.82 | * 0.82 | * 0.82 | * 0.82 | * 0.82 |
| * RMS | * 1.07 | * 1.07 | * 1.07 | * 1.07 | * 1.05 | * 1.06 | * 1.07 | * 1.07 |

2) Num of Iterations

* # of features =10 (fixed), five times average each.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * # of Iterations | * 1 | 2 | * 3 | * 5 | * 10 | * 15 | * 20 | * 100 |
| * AAD | * 0.84 | * 0.80 | * 0.81 | * 0.80 | * 0.82 | * 0.82 | * 0.84 | * 0.83 |
| * RMS | * 1.09 | * 1.03 | * 1.04 | * 1.04 | * 1.05 | * 1.08 | * 1.08 | * 1.08 |

2-3. (10점) 2-2에서 찾은 최적의 MF based recommender를 사용한 추천 결과의 성능과, 일반적인 User based recommender를 사용한 추천 결과의 성능을 아래의 지표를 종합적으로 활용하여 비교하시오. 성능에 차이를 보인 경우, 이에 대한 본인의 의견을 서술하시오.

- AAD(average absolute difference), RMS(root mean squared difference)

최적을 (Feature, Iteration) = (10,2), (10,5)로 설정하였다. 그에 따른 AAD, RMS 값은 다음과 같다.

* AAD: 0.80 / RMS: 1.03, AAD: 0.80 / RMS: 1.04

그리고 User-based average의 성능은 다음과 같았다.

* AAD: 0.845 / RMS: 1.07

일반적으로 parameter를 잘 설정하면 MF based recommender가 User-based보다 더 나은 효과를 보였다. 즉, Model을 사용한 Matrix Factorization method가 일반 user-based보다 더 최적의 예측 성능을 보여준다는 것을 의미한다. 별도로 재미있는 점이 있는데, feature와 iteration 수를 무조건 높게 한다고 AAD나 RMS가 꼭 좋게 나오진 않았다.