Recsys

Zhijie Xie

[1]School of Software Engineering, Tongji University.

[*]To whom correspondence should be addressed; E-mail: shiehshiehzhijie@gmail.com.

1 Group Member

谢志杰 1352975

2 Project Design

本程序提供以下推荐系统算法：

- Baseline estimator.
- Collaborative Filtering estimator
    1. User Based
    2. Item Based
    3. With/Without Dimensionality Reductin

- Time changing baseline predictors (1)
- Clustering for imputing missing value : http://infolab.stanford.edu/ ullman/mmds/ch9.pdf
- SVD based estimator (2)

3 Implementation Detail

本程序使用 Python + Cython 实现，Cython 主要用于数据预处理以及 K-Means 的实现，因为这部分程序的瓶颈在于 python 的解析执行。而推荐系统的瓶颈，即 SVD，Similarity 的计算等步骤，使用的是经过高度优化过的 C 模块，所以不需要做过多的速度优化。

数据提取，预处理　根据项目要求，同一 UserID 下，根据时间顺序，前 90% 作为 training set，后 10% 作为 testing set。由于本次的数据集规模不大，所有的数据都常规矩阵的形似存储。

一种策略是在这个时候对矩阵中的 missing value 进行一个填补，但是我不这么做，因为接下来的几个算法对矩阵的要求不同。比如，clustering 要求该矩阵存在 missing value，因为填补这些 missing value 就是 clustering 在做的工作。

以上是每一个算法共用的预处理方式，由于每个算法有各自的特点，将在每个小节单独说明其数据预处理工作。

3.1 baseline

两种情况:

- trainging set 中，该物品重来没有被用户评论过，此时用该用户的平均评分作为预测值。

- trainging set 中，该用户至少被评论过一次，将用户的平均评分加上物品的偏差作为预测值。

物品倾向：物品的平均评分减去整个数据集的平均评分。

3.2 Collaborative Filtering

没有额外数据处理需求，使用 Pearson correlation 作为相似度度量。

User based 跟 Item based 的实现代码的唯一差别，使用 Item based 的时候，会将矩阵进行转置。因为实现算法时，我尽量将代码泛化，查找最相似 K 个近邻的算法仅对一个矩阵的每行进行相似度对比，所以我可以很轻易的将代码扩展到 Item based（仅需一个矩阵转置)。

同时，也提供使用 SVD 进行降维，然后在降维后的向量空间中进行 similarity 的计算，实验证明该做法可以加快算法的处理速度，同时在降维后的向量空间中所进行的搜索，效率和准确度也更高，因为 SVD 降维可以减少大量的噪音。

3.3 Temporal Dynamic

使用 Pearson correlation 作为相似度度量，需要对数据进行额外处理：

- 在调用算法前，需要将 training set 中所有的评分分成多个 Bin。
- 对于每个用户，也需要将用户做出的评分分成多个时间节点，纪录该时间节点下，用户做出的评分的均值。

预测时，预测值等于：training set 所有评分均值 + 用户在该时间节点下的评分倾向 + 物品在该时间点所属 Bin 中的评分偏差。

3.4 Clustering Based

使用 Pearson correlation 作为相似度度量，需要对数据进行额外处理：

- 使用欧式距离，对用户和物品分别作聚类，得到一个 kx*ky 的新矩阵
- 将处于同一类（x，y）下的所有评分的均值作为新矩阵该类（x，y）的评分。

此时我们可以回填原矩阵，分三步：

- 找出用户 u ，物品 i 分别属于哪一类 x，y，得到新矩阵入口（x，y)。
- 如果（x，y）有值，使用改值作为原矩阵的相应入口（u，i）的值。
- 如果（x，y）无值，搜索跟（x，y）相似的 n 个 clustering，用均值回填（u，i)。

然后使用回填后的原矩阵进行真正的预测工作。

3.5 SVD based

数据预处理：

SVD 无法处理 missing value，为了去除 sparsity，首先需要填补空缺，这里有两个策略：

- 使用用户的平均值填补该用户空缺。
- 使用物品的平均值填补该物品空缺。

实验证明，物品均值效果更佳。(2)

然后是需要做 normalization。此时有两个策略：

- 将评分转化为 z-score.
- 每一行减去用户均值。

实验证明，减去用户均值效果更佳。(2)

SVD 流程：

- **对** training set **进行** SVD **分解，得** U，S **和** V。
- **将** SVD **分解后的特征值对角矩阵** S **减小到** K*K，**即保留前** K **个特征值** S_k。K **需要预先设定，可以通过制定** K **的大小，或指定保留的** variance **的百分比，让算法决定保留的** K **的大小。**
- **计算** S_k **的平方根得** $S_k^{\frac{1}{2}}$。
- **计算** $U_K S_k^{\frac{1}{2}}$ **和** $S_k^{\frac{1}{2}} V_k'$。

预测时，对于用户 u，**物品** i：**预测值为：** $C_{P_{pred}} = C + U_K S_k^{\frac{1}{2}}(u) \cdot S_k^{\frac{1}{2}} V_k'(i)$

4 Experiment Result

4.1 baseline

```
Processing sample: 96700
recsys time: 6629.615250 samples / 1s
RMSE: 0.943337
```

Figure 1: **使用了整个** testing set

4.2 cf user based

```
Processing sample: 1900
recsys time: 13.079562 samples / 1s
RMSE: 0.979284
```

Figure 2: **使用了** 2015 **个** testing sample, neighbor=30

4.3 cf item based

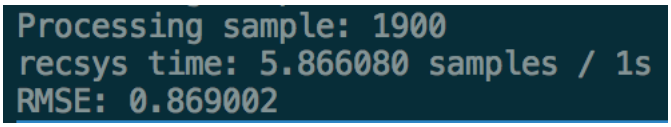

Figure 3: 使用了 2015 个 testing sample, neighbor=20

4.4 cf item based with dimensionality reduction

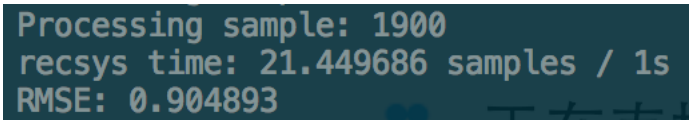

Figure 4: 使用了 2015 个 testing sample, neighbor=20, 1% energy loss.

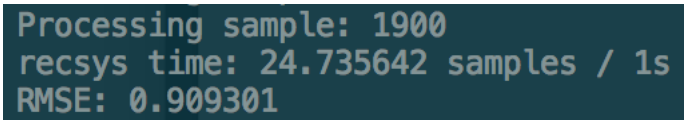

Figure 5: 使用了 2015 个 testing sample, neighbor=20, 10% energy loss.

4.5 temporal

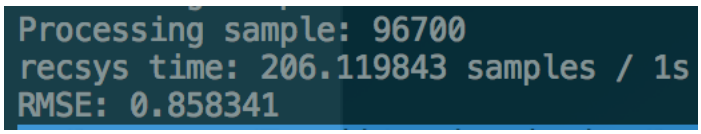

Figure 6: 使用了整个 testing set, bin=333

4.6 clustering

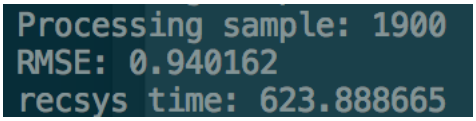

Figure 7: 使用了 2015 个 testing sample, # user cluster=50, # item cluster=25

4.7 svd

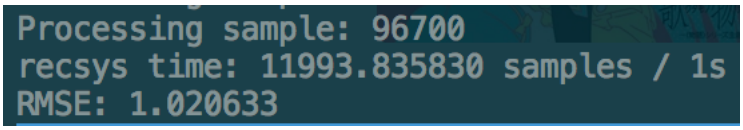

Figure 8: 使用了整个 testing set, 1% energy loss.

5 Conclusion

1. Collaborative Filter 的准确度较高，但是由于需要搜索近邻，导致算法的运行效率较低。实验显示，基于物品的 similarity 效果比基于用户的 similarity 效果要好。

2. SVD Based，由于进行了降维，内存消耗大大减少，同时预测速度极快，因为时间固定为一次向量乘法和一次加法，但是算法极大的依赖于数据预处理的效果，如果 missing value 的填充方法不好，容易导致算法效果不佳。

3. Collaborative filtering with temporal dynamics，算法将时序因素加以考虑，同时不依赖于近邻的计算，所以精度较高，速度也相对较快。不过由于引入了较多的 hyper parameters，需要做更多的 finetune，对样本的数量和时间分布也有要求，导致算法的使用难度加大。

4. 使用聚类对原矩阵填充，消除 missing value 的方法，理论上可以提高系统的精确

度，但是，由于聚类算法的性能差异很大，效果存在一定的随机性，回填原矩阵开销巨大，等等因素也一定程度上限制了这一算法的应用。

References and Notes

1. Y. Koren, Communications of the ACM 53, 89 (2010).

2. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of dimensionality reduction in recommender system-a case study, Tech. rep., DTIC Document (2000).