

Initial Algorithm2:(score based)

行为共 5 类/每次计分:

- 0 - 无行为 - 0 分
- 1 - 点击 - a 分 (rec: 1)
- 2 - 收藏 - b 分 (rec: 16)
- 3 - 购物车 - c 分 (rec: 32)
- 4 - 购买 - d 分 (rec: 64)

注: 基础分数设计需仔细思考, 如按 rec 的值则没有上限, 且会较大; 建立打分的时候建议仍建立一个类别矩阵 (如 category based 中)

基于假设:

1. 确定为购买行为后预测未来不会重复购买 (little sacrifice)
2. 确定为购物车行为后直接预测未来会购买 (little mistake)

推荐方法:

基于 SVD, 对类别矩阵中类别标定为行为 2、1、0 的在分数矩阵中进行分数预测更新。推荐可以提供 TopN, 但是 TopN 中不应包含类别矩阵中原来就对应购买行为的。

或者, 我想可以计算出阈值 T, 对于预测大于 T 的进行推荐 (预测购买)。T 值的设定可以用最简单的学习算法利用原有的分数矩阵和行为矩阵 (这里行为分为购买和未购买两类), 求使损失函数最小的分数 T 为阈值

Necessary Work I think: (in Java)

1. 建立 user 和 brand 的索引表 (先建集合再遍历生成数组)

```
User = {'zhangsan', 'lisi', 'wangwu', ...};
```

```
Brand = {'book', 'CD', 'pencil', ...};
```

2. 建立 user 和 brand 对应的分数矩阵和行为矩阵

```
Score = { {80,      8,   24,  49,  0,   ...},
           {10,     7,   92,  31,  0,   ...},
           {101,    13,  65,  52,  0,   ...},
           {70,     20, 114, 19,  0,   ...},
           ...      };
```

```
Behavior = { {4, 1, 2, 3, 0, ...},
              {1, 1, 4, 2, 0, ...},
              {4, 1, 3, 3, 0, ...},
              {3, 2, 4, 2, 0, ...},
              ...      };
```

3. 建立预测行为矩阵并通过算法进行填充

```
Prediction = { { ... },
                { ... },
                { ... },
                ...      };
```

算法的填充原则：原来为 4 的直接填充 0；

原来为 3 的直接填充 4；

原来为 2、1、0 的通过 SVD 得分 TopN 填充为 4，其余为 0（或 SVD 得分大于阈值 T 的填充 4，小于等于的填充 0）；