

# Slope one predictors for online rating-based collaborative filtering

Daniel Lemire and Anna  
协同过滤算法： Slope One

查找翻译词汇

Related Work	相关工作
CF Algorithms	CF 算法
WEIGHTED	加权

## Abstract摘要&介绍

## 1 Introduction—相关介绍

作者认为协同过滤必须具备的要素

- 1易于实现和维护
- 2可随时更新
- 3查询效率高
- 4对新用户的推荐度高
- 5较高的准确性

## 2 Related Work—相关工作

作者主要提出三种SLOPE ONE算法

The SLOPE ONE Scheme

SLOPE ONE算法

$$P^{wSl}(u)_j = \frac{\sum_{i \in S(u) - \{j\}} (\text{dev}_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u) - \{j\}} c_{j,i}}$$

在预测一个对新item的评分时，将用户过去对所有item的评分加以平均

The WEIGHTED SLOPE ONE Scheme

加权SLOPE ONE算法

$$P^{wSl}(u)_j = \frac{\sum_{i \in S(u) - \{j\}} (\text{dev}_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u) - \{j\}} c_{j,i}}$$

$$c_{j,i} = \text{card}(S_{j,i}(\chi))$$

理解加权的意义：  
Slope One中在计算 item i 相对于 item j 的平均偏差 devj,i 时没有考虑到使用不同的用户数量平均得到的 devj,i， 其可信度不同。所以作者通对slope one进行加权

例如：有item i, item j, item k  
其中有10000人同时评价了 i 和 j 而只有100人评价 j 和 k 那么 i 和 j 的评分应该占较高的权重所以获得devj,i比获得devj,k更重要

$$P^{bpSl}(u)_j = \frac{\sum_{i \in S^{like}(u) - \{j\}} p_{j,i}^{like} c_{j,i}^{like} + \sum_{i \in S^{dislike}(u) - \{j\}} p_{j,i}^{dislike} c_{j,i}^{dislike}}{\sum_{i \in S^{like}(u) - \{j\}} c_{j,i}^{like} + \sum_{i \in S^{dislike}(u) - \{j\}} c_{j,i}^{dislike}}$$

The BI-POLAR SLOPE ONE Scheme

BI-POLAR SLOPE ONE算法

对BI-POLAE SLOPE ONE算法的一些理解：  
也许是作者也考虑很多人如果喜欢某一类商品时会对这类商品加分（like） 而不喜欢某类商品时会减分（dislike） 基于两类集合的评分来求得一个合理的推荐

Scheme	EachMovie	Movielens
BI-POLAR SLOPE ONE	0.194	0.188
WEIGHTED SLOPE ONE	0.198	0.188
SLOPE ONE	0.200	0.188
BIAS FROM MEAN	0.203	0.191
ADJUSTED COSINE ITEM-BASED	0.209	0.198
PER USER AVERAGE	0.231	0.208
PEARSON	0.194	0.190

最后作者对各种算法之间进行测试比较：  
所有方案的平均误差率都是针对  
EachMovie和Movielens数据集，

越低越好