

传统的推荐算法分为三大类：

: 性质, 注释 : 优点 : 缺点

1、基于内容的推荐 (Content Based, CB)，基于用户特征属性和item的特征之间的匹配程度来做推荐，推荐效果强依赖于特征工程的好坏。取决于 FE 能否准确提取 item-user 的固有特征

好处是用户独立性，不需要协同考虑其他用户特征，这也意味着一个缺点就是可供我们分析的内容有限。
即使 user-item 不断增多对 CB 也无影响

再一个好处是没与用户互动（评分、购买、加购物车等）过的新的item也可以被推荐。

最大的缺点是过度特化，推荐的items没有创新性和惊喜。

2、协同过滤 (CF)，利用的信息是用户和item之间的互动信息（多用的是用户评分矩阵），CF在评分矩阵信息较为稠密的时候表现好于CB，并且CF可以捕捉到一些复杂的难以表达的信息；在实际应用中CF经常会产生意外的惊喜的推荐，这一点也暗示着CF方法不适合做公共基金推荐等严肃的推荐任务（因为CF主要基于稀疏的、不完整的数据做推荐）；再一个CF存在冷启动问题。

CF分为：
Collaborative Filtering
CF缺点：
I. Data Sparsity
II. Cold start

(1) Memory based rec，基于记忆的推荐又可以分为：基于用户的推荐和基于item的推荐。

基于用户的推荐：给目标用户做推荐时考虑与其最相似的N个用户的喜好信息，这一类方法有一个潜在的假设就是说如果两个用户过去有相似的偏好，那么他们未来也有相似的偏好，用户偏好不随时间变化。
前提

基于item的推荐，有两种，以电影推荐为例：一种是如果用户看过电影A，直接将与A相似的电影推荐给该用户；再一种是根据用户看过的电影与A的相似度来决定。

基于item的推荐更常用，因为：

KNN 成为瓶颈

- 1) User-based 方法随着用户数量的不断增多，“最近邻用户搜索”会成为整个算法的瓶颈。Item-based 方法通过计算items之间的相似性来代替用户之间的相似性。对于items来讲，它们之间的相似性要稳定很多，因此可以离线进行相似性计算步骤，提高推荐效率。✓
- 2) 同一个item得到的评分比较类似，分布较稳定，而一个用户对不同items的评分方差却很大。
- 3) 一个item的属性特征变化不大，而用户画像变化频繁，兴趣经常变化。

(2) Model based rec，基于一些分类、聚类、关联规则等机器学习模型来做推荐，最出名的是矩阵分解类方法。(MF). 因子分解机(FM)

Sparse Linear Methods

(3) 也有人把CF分为三大类，除了前两类外，还有SLIM及其改进方法。

3、混合推荐方法，CF+CB，分为两大类：

Memory-based
Model-based
SLIM - GLSLIM ...

(1) 松耦合混合推荐：将用户或item的属性信息作为辅助信息来为CF提供特征，整个过程是单向的，CF中的UI互动矩阵并没有反馈来促进辅助信息中特征的提取。

User-Item interaction matrix

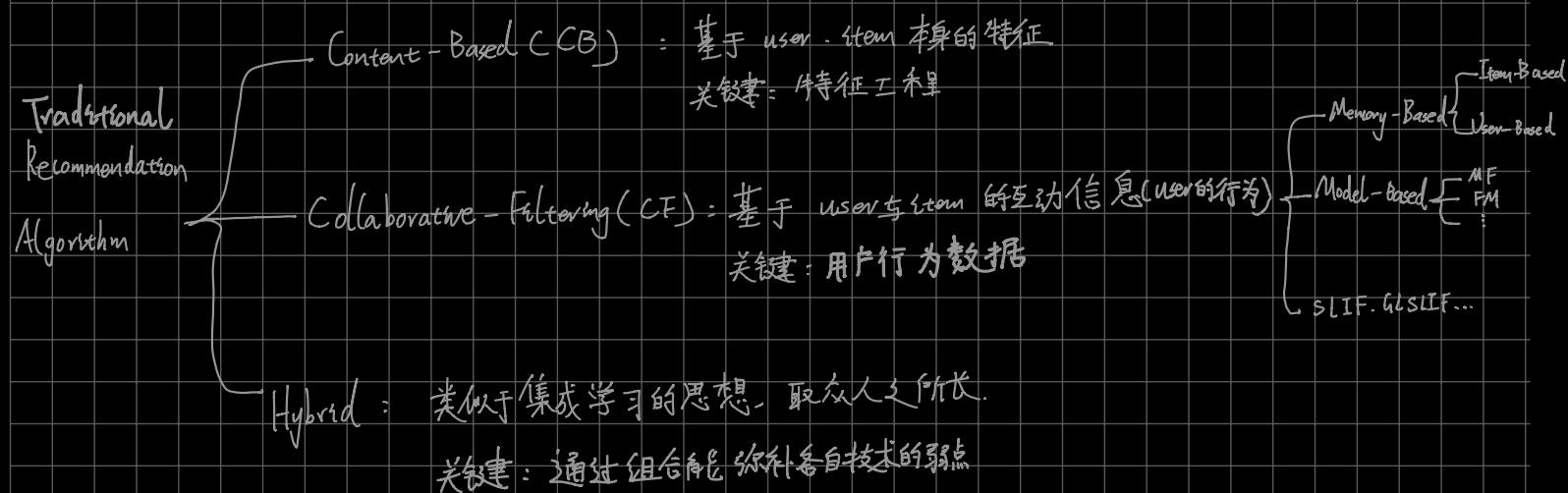
(2) 紧耦合混合推荐：双向的，CF中的UI互动矩阵信息guide辅助信息的特征提取过程，而提取的特征再来improve CF方法的性能

(2) 效果是要好于 (1) 的。

E.g.: YouTube Video Recommendation Algorithm: CB + CF (Item-Based)

+ Model-Based, 只计算同一 topic 下 视频相似度 \Rightarrow 更好利用长尾
根据 CB
根据 CF
数据，增强了推荐系统的覆盖率，且提升了推荐效果。
根据 MB

Classification of traditional RS Algorithm:



Top-K

Recommendation

Neighborhood-Based

User-Based

Item-Based

\Rightarrow 计算速度快，
 \Rightarrow 推荐效果差

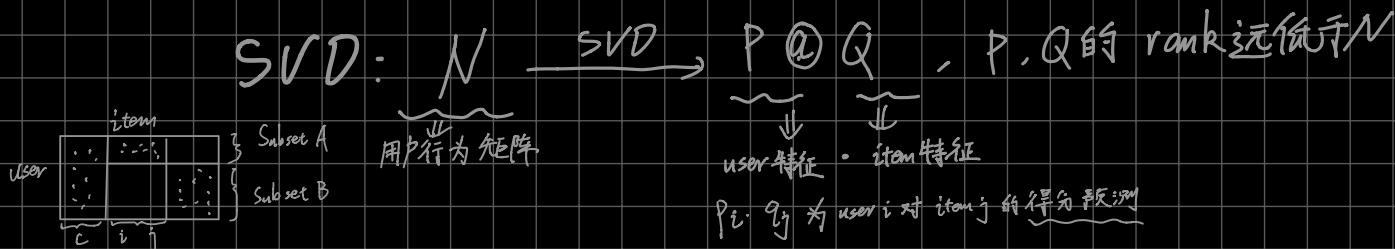
Model-Based : SVD, SVD++, ALS... \Rightarrow
计算速度慢 (model training 需
大量时间)， 推荐效果好

SLIM, GLSLIM : SVD, 会议 blogs \Rightarrow
提升计算速度，又保证了高的准确率

SLIM (Sparse Linear Method)

Prior Knowledge:

SLIM 与 SVD, SVD++ algorithm 类似，但是基于矩阵
在 RS 中的 SVD 与 LA 中 SVD 不同
的方法做推荐。



- Steps:
1. Get matrix N from data collector
 2. Init P, Q random values

3. $\hat{R} = \text{np.dot}(P \cdot Q)$
得分预测

4. $L = \underset{P \cdot Q}{\operatorname{argmin}} \sum_{(u, i) \in K} (r_{ui} - p_u \cdot q_i^T)^2 + \lambda (\|P\|^2 + \|Q\|^2)$
正则化系数，控制方差

5. GD 更新 P, Q 中的值，重复 3-5，直到终止

SVD++:

在实际应用中，SVD 存在两个问题：

- ① 数据存在偏见 \Rightarrow 某些 item, user 的偏高
- ② 显式数据 << 隐式数据

因此提出 SVD++ \Rightarrow

\hat{R} 得分 = 显式兴趣 + 隐式兴趣 + 偏见
 $\hat{R} = \text{P} @ \text{Q}$
 局部评分 item bias user bias

$$\hat{r}_{ui} = p_u \cdot q_i^T + \left(\underbrace{|N(u)|^{-\frac{1}{2}}}_{\substack{\downarrow \\ \text{用户 } u \text{ 的行为物品集}}} \sum_{i \in N(u)} y_i \right) \cdot q_i^T + u_t b_i + b_u$$

物品 j 所表达的隐式兴趣

$$L = \underset{p, q}{\operatorname{arg \min}} \sum_{(u, i) \in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda \cdot \left[\|p_u\|^2 + \|p_u\|^2 + \|b_u\|^2 + \|b_u\|^2 \right]$$

其余步骤与 SVD 相同。

Why SLIM:

- Neighborhood - Based method 虽然计算速度很快 (N 很稀疏)，但预测准确率不高。
- Model - Based method 虽然预测准确率高，但是训练 model 采用 ML 的方法 (GD, Adam...)，速度较慢。
- 而 SLIM 能结合两种方法优点，提升计算速度，又保证了高的准确率，因此被工业界广泛使用。

Learning Steps:

$$\underset{W}{\operatorname{minimize}} \quad \underbrace{\frac{1}{2} \|N - NW\|_F^2}_{\text{loss}} + \lambda |W| + \underbrace{\frac{\beta}{2} \|W\|_F^2}_{\substack{L_1 \\ L_2}}$$

最小二乘法

$\underbrace{W}_{\substack{\text{权重矩阵} \\ (\text{本版为物品间相似度矩阵})}}$

subject to $\begin{cases} W \geq 0 \Rightarrow \text{只学习 item 间正相关关系} \\ \operatorname{diag}(W) = 0 \Rightarrow \text{避免推荐自身 (若无此限制, 在 GD minimize 时, 会倾向于只推荐自身)} \end{cases}$

速度提升：特征选择过程利用了 neighbourhood-based method 来缩短训练时间。

当 minimize 时，对 N_j ，只选择与 N_j 相似度较高的 W_k 相乘，大大加速了训练过程。

Neighborhood-Based \Rightarrow k 有几个取值，但 N_k 与 N_j 相似度较高

将与用户相似度较低的 W_k 置 0

GLSLIM (Global Local SLIM)

Main idea:

Calculate the score estimation of user u to item i :

$$\tilde{r}_{ui} = \sum_{l \in \mathcal{R}_u} g_u s_{li} + (1 - g_u) s_{li}^{p_u} \quad (3)$$

预测函数 global preference, 用户对全局信息的偏好程度

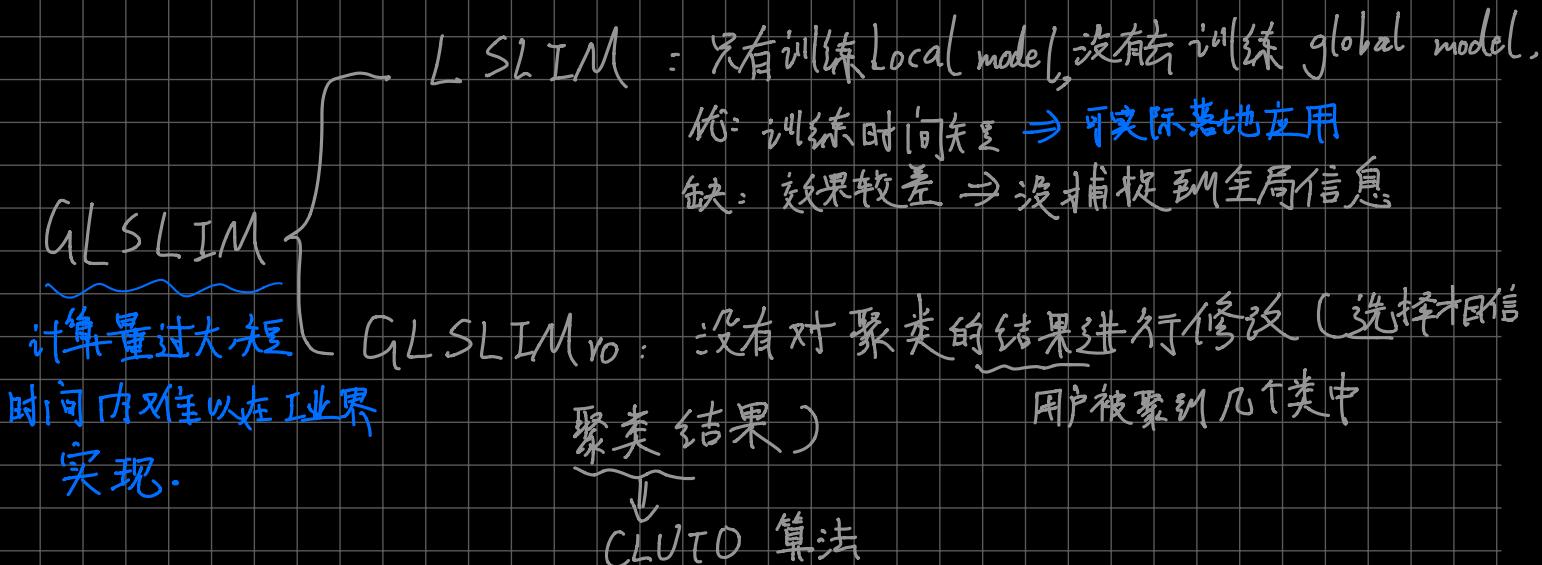
Calculate the loss function:

$$\begin{aligned} \underset{\mathbf{s}_i, \{\mathbf{s}_i^1, \dots, \mathbf{s}_i^k\}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{r}_i - \mathbf{g} \odot R\mathbf{s}_i - \mathbf{g}' \odot \sum_{p_u=1}^k R^{p_u} \mathbf{s}_i^{p_u}\|_2^2 + \\ & \frac{1}{2} \beta_g \|\mathbf{s}_i\|_2^2 + \lambda_g \|\mathbf{s}_i\|_1 + \\ & \sum_{p_u=1}^k \frac{1}{2} \beta_l \|\mathbf{s}_i^{p_u}\|_2^2 + \lambda_l \|\mathbf{s}_i^{p_u}\|_1, \\ \text{subject to} \quad & \left. \begin{array}{l} \mathbf{s}_i \geq 0, \\ \mathbf{s}_i^{p_u} \geq 0, \forall p_u \in \{1, \dots, k\}, \\ s_{ii} = 0, \\ s_{i,i}^{p_u} = 0, \forall p_u \in \{1, \dots, k\}, \end{array} \right\} \text{与 SLIM 相似} \end{aligned} \quad (4)$$

Calculate the user preference:

$$g_u = \frac{\sum_{i=1}^m (\sum_{l \in \mathcal{R}_u} s_{li} - \sum_{l \in \mathcal{R}_u} s_{li}^{p_u}) (r_{ui} - \sum_{l \in \mathcal{R}_u} s_{li}^{p_u})}{\sum_{i=1}^m (\sum_{l \in \mathcal{R}_u} s_{li} - \sum_{l \in \mathcal{R}_u} s_{li}^{p_u})^2}. \quad (5)$$

GLSLIM 的变种:



Algorithms

Algorithm 1 GLSLIM

```
1: Assign  $g_u = 0.5$ , to every user  $u$ .  
2: Compute the initial clustering of users with CLUTO [1].  
3: while number of users who switched clusters > 1% of  
    the total number of users do ↓聚类的结果不再调整  
4:   Estimate  $S$  and  $S^{pu}$ ,  $\forall p_u \in \{1, \dots, k\}$  with Equation  
    4. 物品在  $p_u$  cluster 中的 similarity  
5:   for all user  $u$  do  
6:     for all cluster  $p_u$  do → 用户  $u$  的偏好系数，未更偏向于 global model / local model  
7:       Compute  $g_u$  for cluster  $p_u$  with Equation 5.  
8:     Compute the training error.  
9:   end for  
10:  Assign user  $u$  to the cluster  $p_u$  that has the smallest  
    training error and update  $g_u$  to the corresponding  
    one for cluster  $p_u$ .  
11: end for  
12: end while
```

Algorithm 2 LSLIM

```
1: Compute the initial clustering of users with CLUTO.  
2: while number of users who switched clusters > 1% of  
    the total number of users do (与 original algorithm 一致)  
3:   Estimate  $S^{pu}$ ,  $\forall p_u \in \{1, \dots, k\}$  with Equation 8.  
4:   for all user  $u$  do  
5:     for all cluster  $p_u$  do  
6:       Compute the training error.  
7:     end for  
8:     Assign user  $u$  to the cluster  $p_u$  that has the smallest  
     training error.  
9:   end for  
10: end while
```

Algorithm 3 GLSLIMr0

```
1: Assign  $g_u = 0.5$ , to every user  $u$ .  
2: Compute the initial clustering of users with CLUTO. → 相信 initial clustering 结果  
3: while  $diff > 0.01\%$  do → 本轮迭代误差差与上轮相比差距不大时，用误差替代原终止条件.  
4:   Estimate  $S$  and  $S^{pu}$ ,  $\forall p_u \in \{1, \dots, k\}$  with Equation  
    4.  
5:    $\forall$  user  $u$  compute  $g_u$  with Equation 5.  
6:   Compute difference in the objective function ( $diff$ )  
    between subsequent iterations.  
7: end while
```