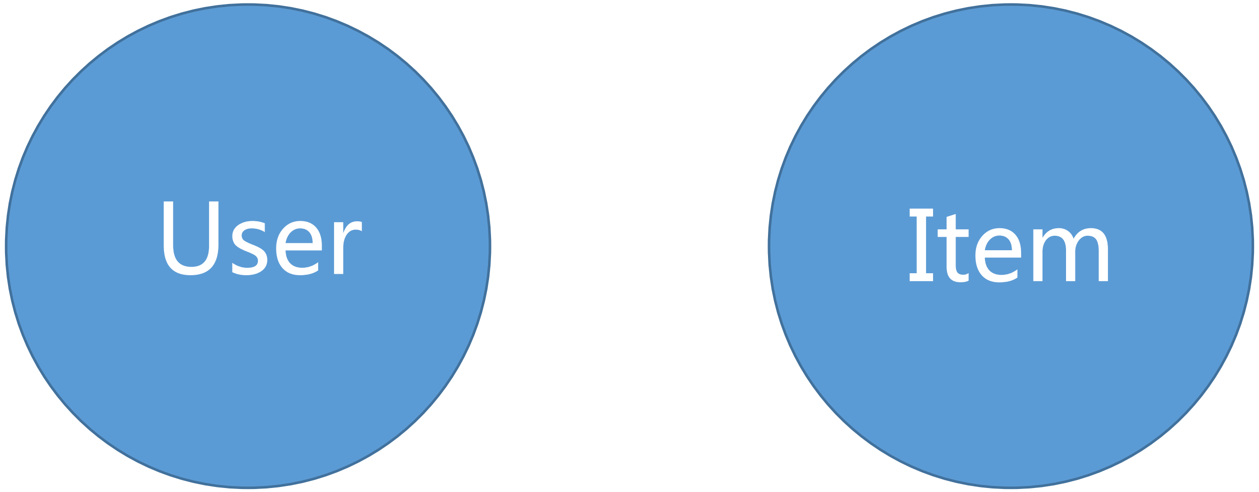
ARCF Introduction

--Xinpeng Guo

Basic Concepts

Two Basic Entity



1. Song
2. Singer
3. Song List
4. MV
5. Album

...

Three Basic Relationships

User-User

1. Circle
2. Discover New People
   (Famous People)
3. People U may know
   (Social Relation Chain)

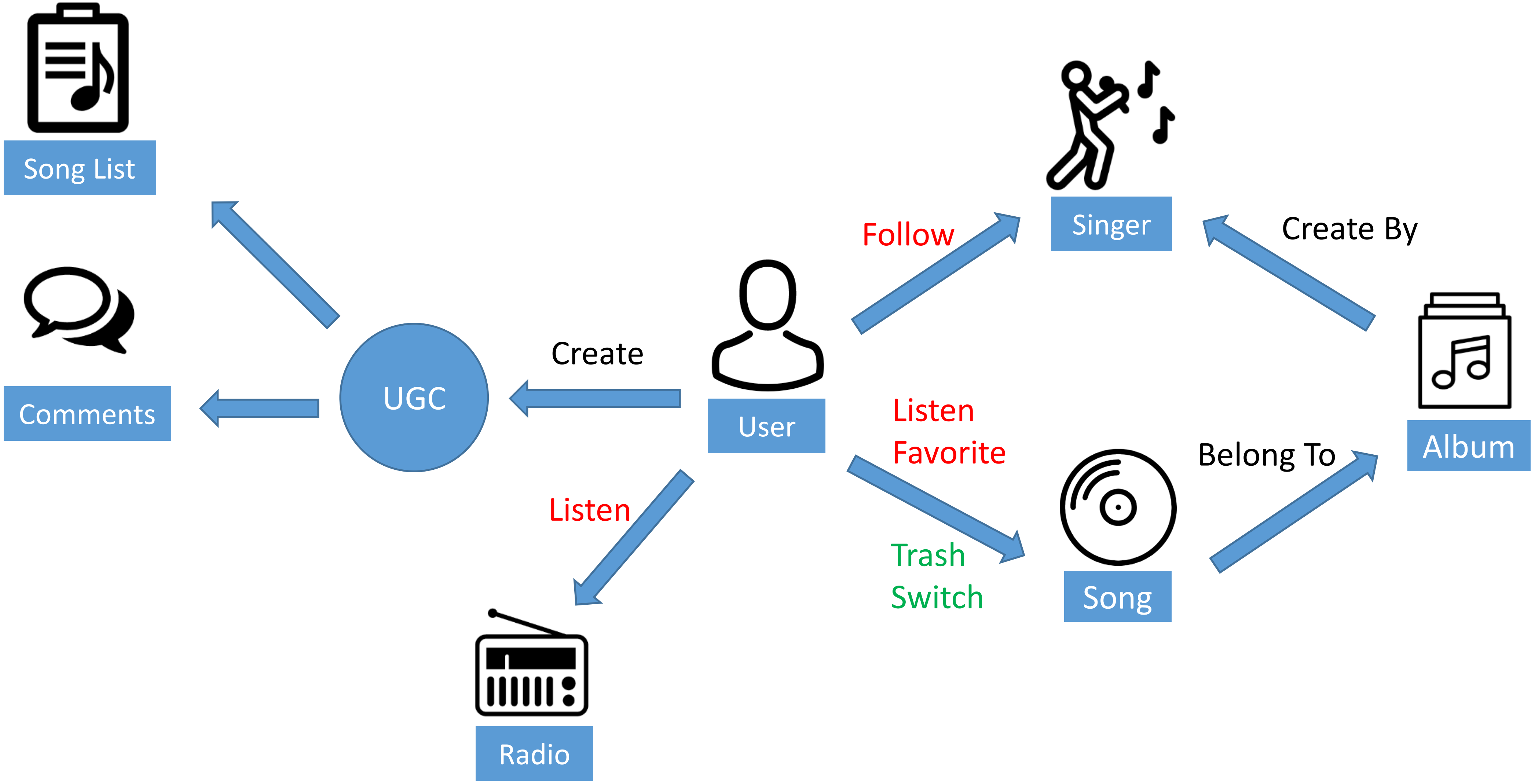
Guess U like

User-Item

Item-Item

1. Similar Songs
2. Single Song Radio
3. Buy This Also Buy
4. View This Also View

Knowledge Graph



Data Flow

Three Steps

User
Item
Action

→

User
Item
Preference

→

U2U
U2I
I2I

把事情说清楚

明细

把事情总结一下

汇总

数据挖掘结果

应用

UIA=Who+Where+When+What

t_7d_imusic_user_item_action			
Field	**Type**	**Instruction**	**Example**
user_id	string	用户ID	12345
item_id	string	物品ID	54321
action	int	动作枚举值，1=点击，2=下载，3=收藏，4=删除，5=观看	2
location	int	位置，1000001=首页banner，100002=榜单页新歌榜	1000001
timestamp	bigint	linux时间戳	1504784445
extend	string	扩展位置（如观看时长长度等），分号分隔	60
Partition	**Type**	**Instruction**	**Sample**
ds	string	日期，格式：YYYY-MM-DD	“2017-9-7”
item_type	int	物品类型，1=歌曲，2=歌手，3=歌单，4=mv	1

三级分隔符	分号
	逗号
	冒号

UIP =Who+How+What

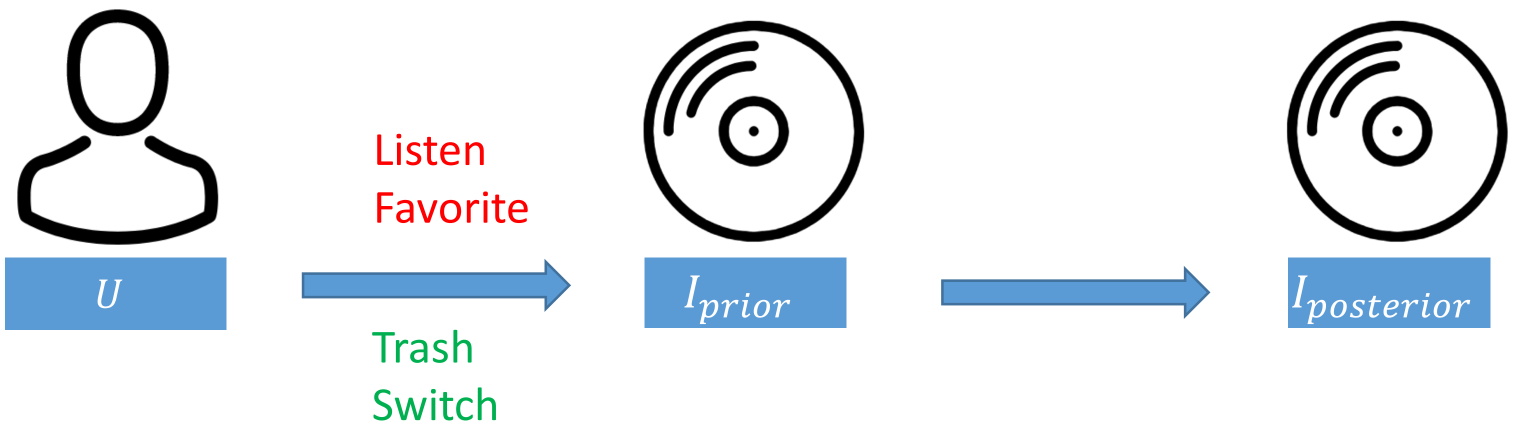
Real Matrix

t_7d_imusic_user_item_preference			
Field	**Type**	**Instruction**	**Example**
user_id	string	用户ID	12345
item_id	string	物品ID	54321
preference	double	分值（0-5分）	1.432
Partition	**Type**	**Instruction**	**Sample**
ds	string	日期，格式：YYYY-MM-DD	"2017-9-7"
item_type	int	物品类型，1=歌曲，2=歌手，3=歌单，4=mv	1

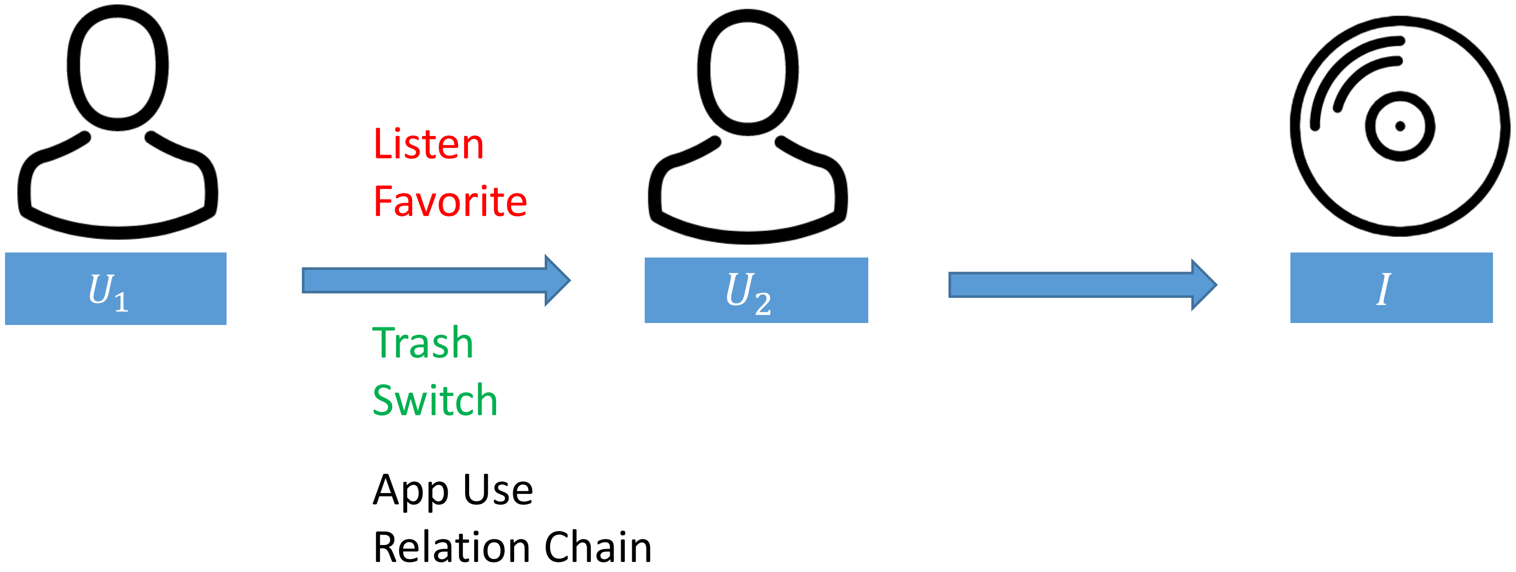
Bool Matrix

t_7d_imusic_user_item_preference_bool_matrix			
Field	**Type**	**Instruction**	**Example**
user_id	string	用户ID	12345
item_id	string	物品ID	54321
preference	int	1喜欢，0不喜欢/不知道	1.432
Partition	**Type**	**Instruction**	**Sample**
ds	string	日期，格式：YYYY-MM-DD	"2017-9-7"
item_type	int	物品类型，1=歌曲，2=歌手，3=歌单，4=mv	1

$$U2I_{posterior} = UI_{prior} + I_{prior}2I_{posterior}$$



$$U_1 2I = U_1 2U_2 + U_2 I$$

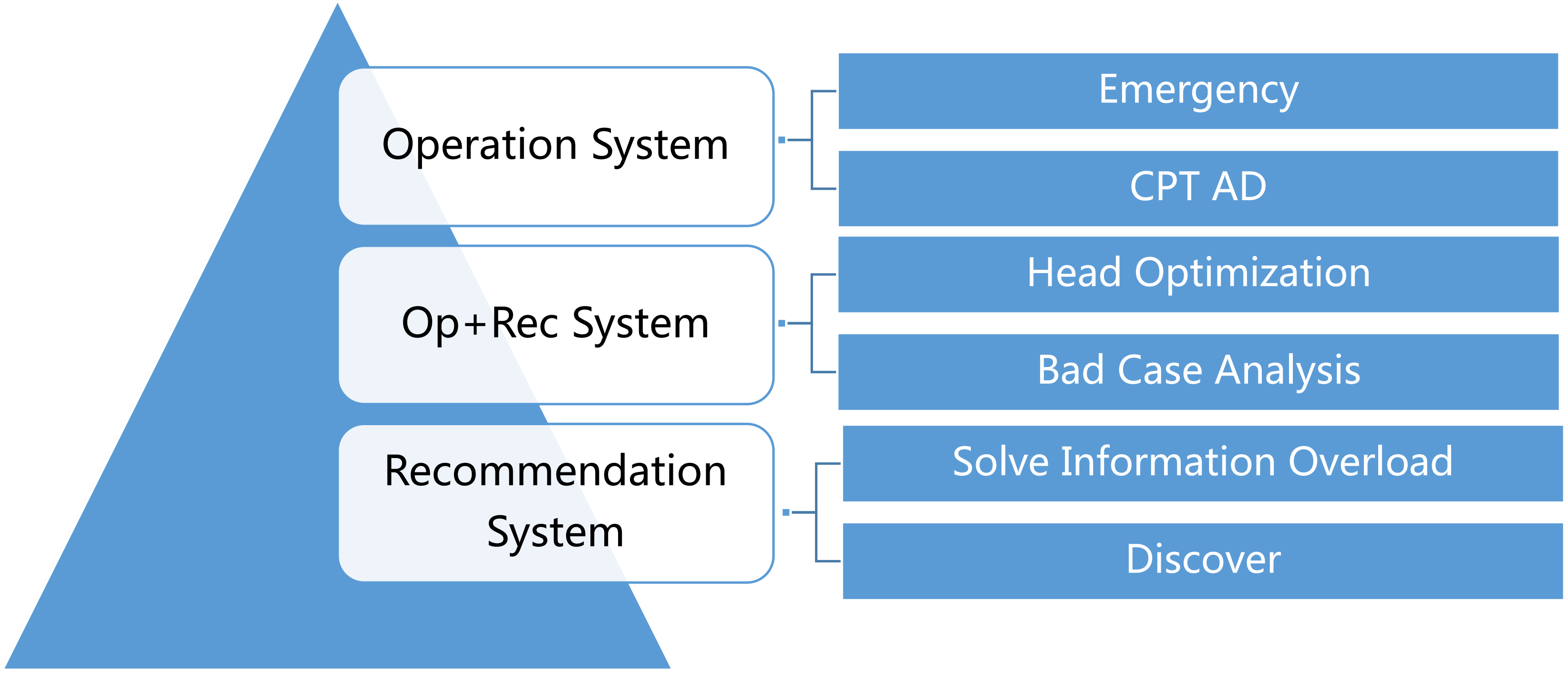


Two Thoughts

- Action Based : CF
- Content Based : Profile + pCTR

Architecture

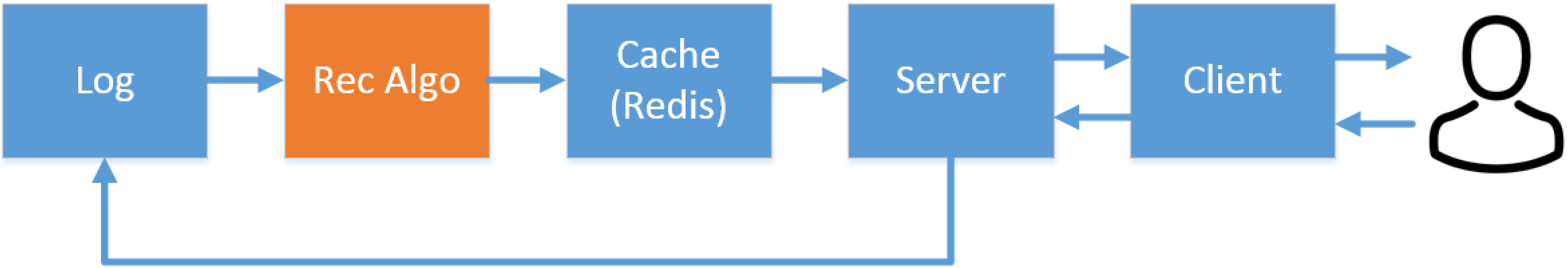
Three Layers



Framework


业务应用层
歌曲推荐
歌单推荐
基础平台层
画像系统
用户画像
歌曲画像
关联系统
相似歌曲
相似歌手
实时化
特征实时
模型实时
算法平台
协同过滤
矩阵分解
EE探索
机器学习
评估系统
点击率
覆盖率
多样性
新颖性
监控&测试
ABTest系统
监控平台
数据整合层
0-1偏好矩阵
实数偏好矩阵
行为数据整合
训练数据生成
原始数据层
实体数据
歌单数据
歌手数据
专辑数据
歌曲数据
电台数据
MV数据
行为数据
播放
点击
收藏
切歌
搜索
下载
分享

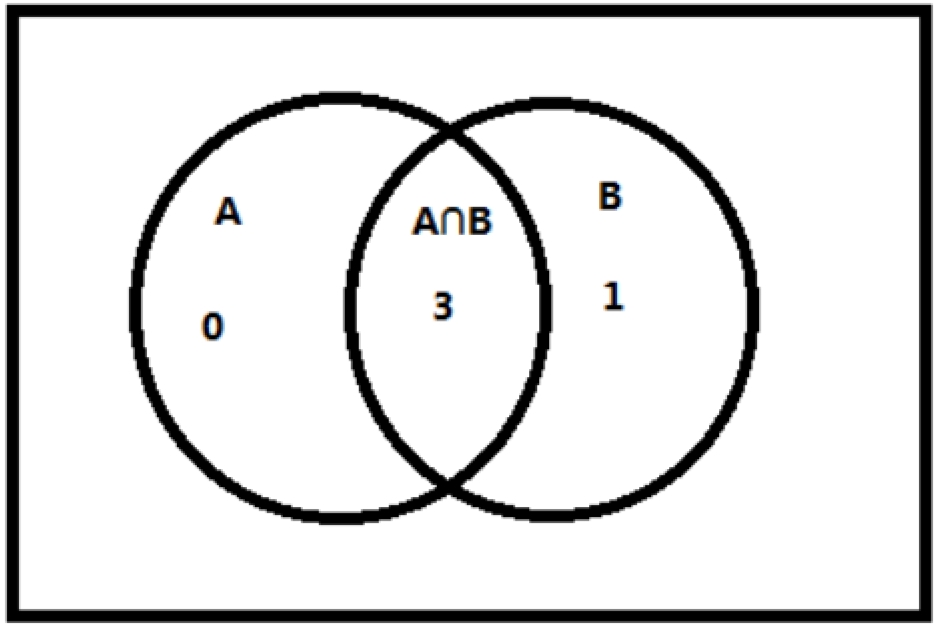
Data Closed Loop



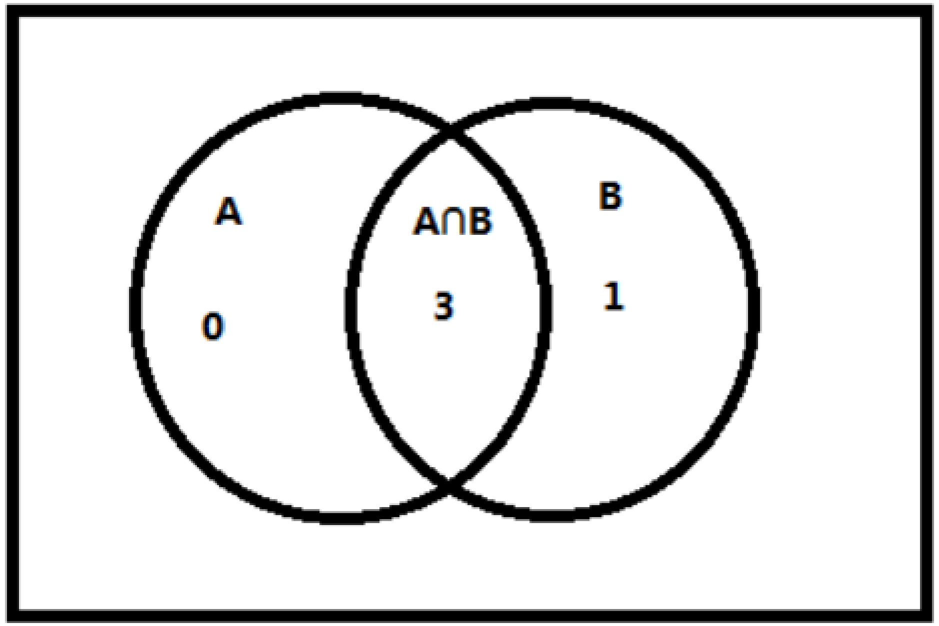
AR—Bool Matrix

Basket Analysis

购物篮
A B
B C
A B C
A B
C D
D E
C E

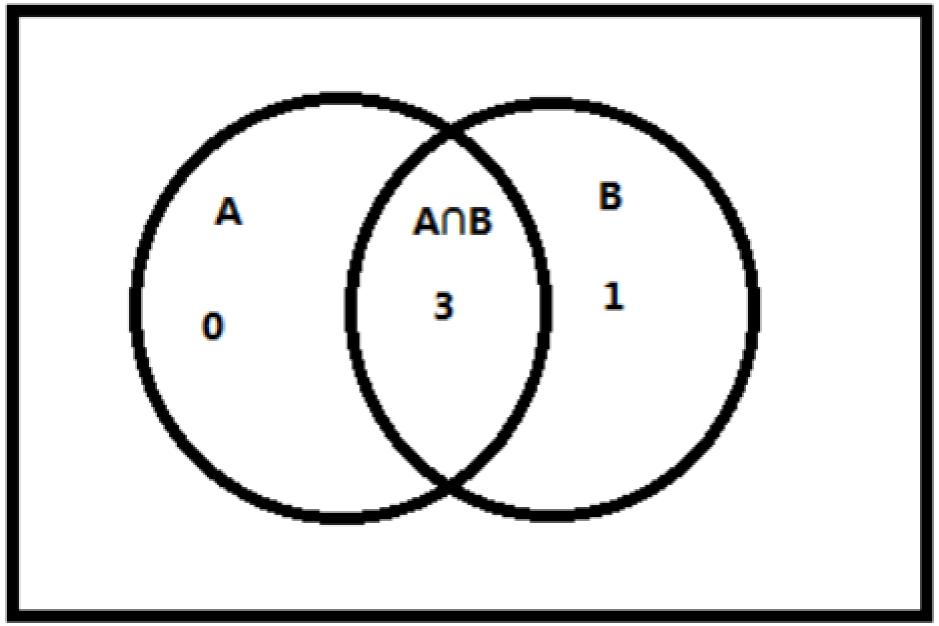


AR Metrics—Pair Frequency



`Count(A, B) = 3`

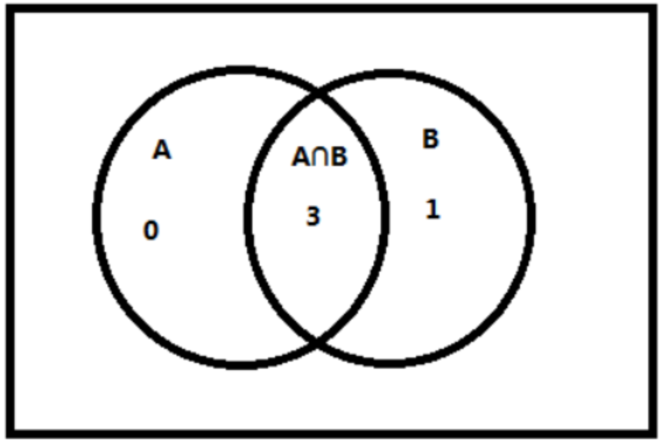
AR Metrics—Jaccard



`Jaccard(A, B) = count(A∩B) / count(A∪B) = 3/4`

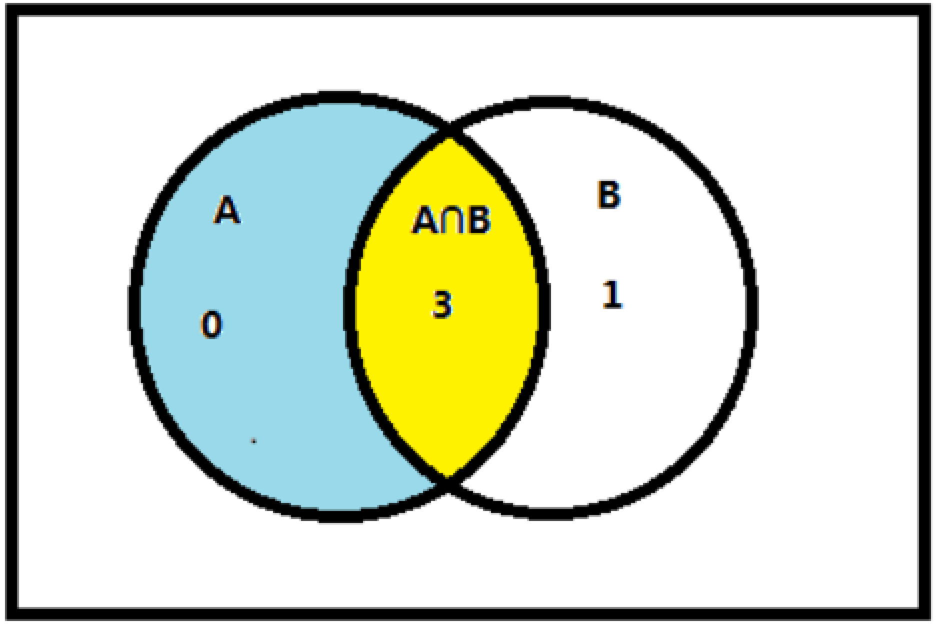
AR Metrics—Support

购物篮
A B
B C
A B C
A B
C D
D E
C E



Support (A, B) = count(A∩B) / count(ALL) = 3 / 7

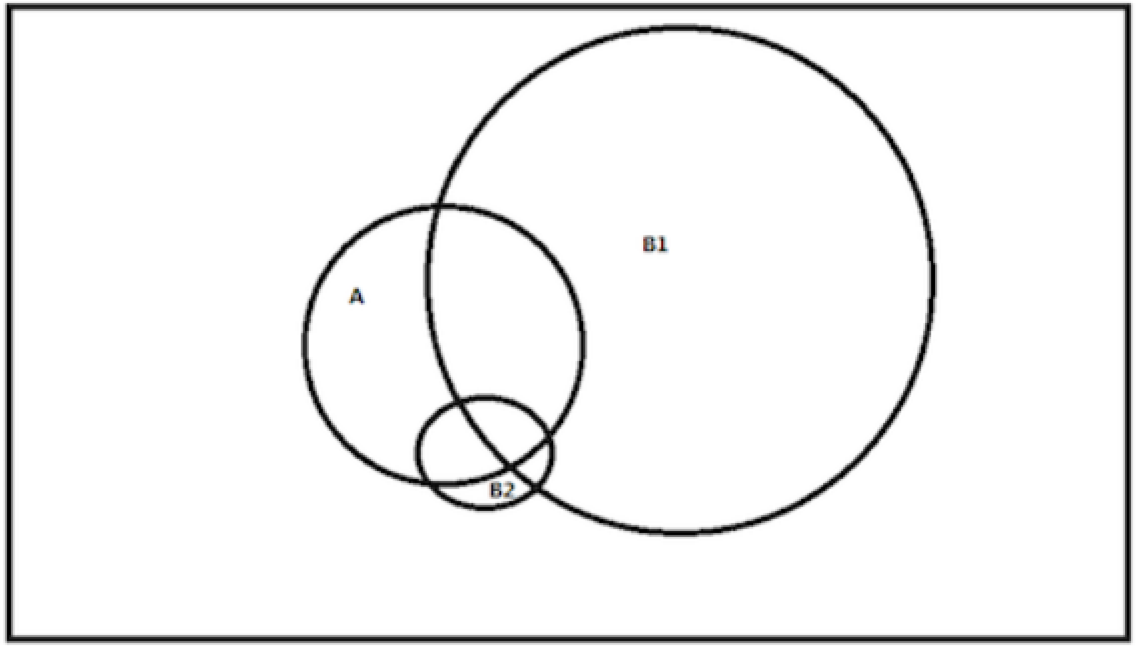
AR Metrics—Confidence



$$\text{Confidence}(A\text{->}B) = P(A \cap B | A) = P(\text{yellow}|\text{yellow+blue}) = 1.0$$

AR Metrics—Confidence Pain Point

【置信度的痛点】 但是推荐过程中，使用会出现偏热的现象，是因为后验B太热门导致的。

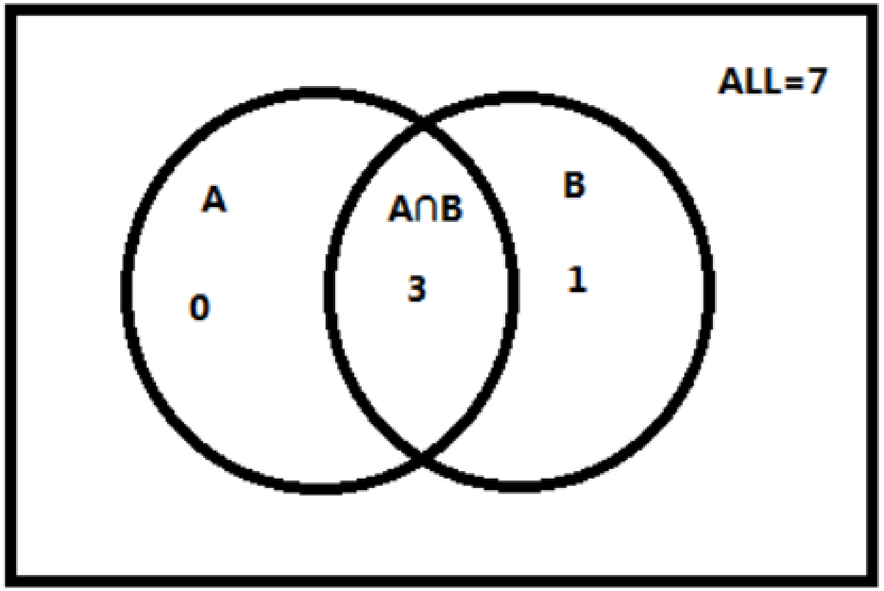


A是用户的先验（比如是手管的app）； B1是一款热门软件（比如微信）； B2是一款相对冷门的软件（比如同步助手）；

可以发现由于B1的体量太大，导致conf(B1|A)很大； 由于B2的体量较小，导致conf(B2|A)偏小； 但是明显应该推荐B2，因为B2都快和A重合进去了，应该对B1的体量进行惩罚。

AR Metrics—Lift

【提升度】 提升度是置信度B1体量惩罚的其中一种方法，含义为：



lift(B|A) = conf(B|A) / P(B) = p(A∩B) | (P(A) * P(B)) = count(A, B) * ALL / (count(A) * count(B)) = (3 乘 7) / (3 乘 4) = 21/12

其中， lift等于1，表示先验知识A的知道与否对B的概率没有影响。 lift大于1，表示促进作用； lift小于1，表示抑制作用；

发现， lift(A|B)=lift(B|A)，即lift是对称的。

AR Metrics—KULR/IR

Given two itemsets, A and B, the **Kulczynski** measure of A and B (abbreviated as **Kulc**) is defined as

$$Kulc(A, B) = \frac{1}{2}(P(A|B) + P(B|A)). \tag{6.11}$$

"*Among the* all_confidence, max_confidence, Kulczynski, *and* cosine *measures, which is best at indicating interesting pattern relationships?*"

To answer this question, we introduce the **imbalance ratio** (**IR**), which assesses the imbalance of two itemsets, A and B, in rule implications. It is defined as

$$IR(A, B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}, \tag{6.13}$$

AR Metrics—LLR

其他：【卡方值】和lift特性非常相似的还有卡方值，其和lift指标二选一即可，此处不再介绍。【LLR】全称，log-likelihood ratio，是根据熵的变化计算相似度的一种方式。对于item项A和B来说，有：

	Event A	Everything but A
Event B	A and B together (k_11)	B, but not A (k_12)
Everything but B	A without B (k_21)	Neither A nor B (k_22)

LLR的计算公式为：

LLR = 2 sum(k) (H(k) - H(rowSums(k)) - H(colSums(k)))

其中，熵的计算方法为：

H = function(k) {N = sum(k) ; return (sum(k/N * log(k/N + (k==0)))}

由公式可知：

1. 满足交换律，LLR（a，b）=LLR(b, a)
2. 表达式恒大于0，0表示不相关，>0表示相关，可能正相关，也可能负相关。

Sim—Real Matrix

Sim Metrics—Cos

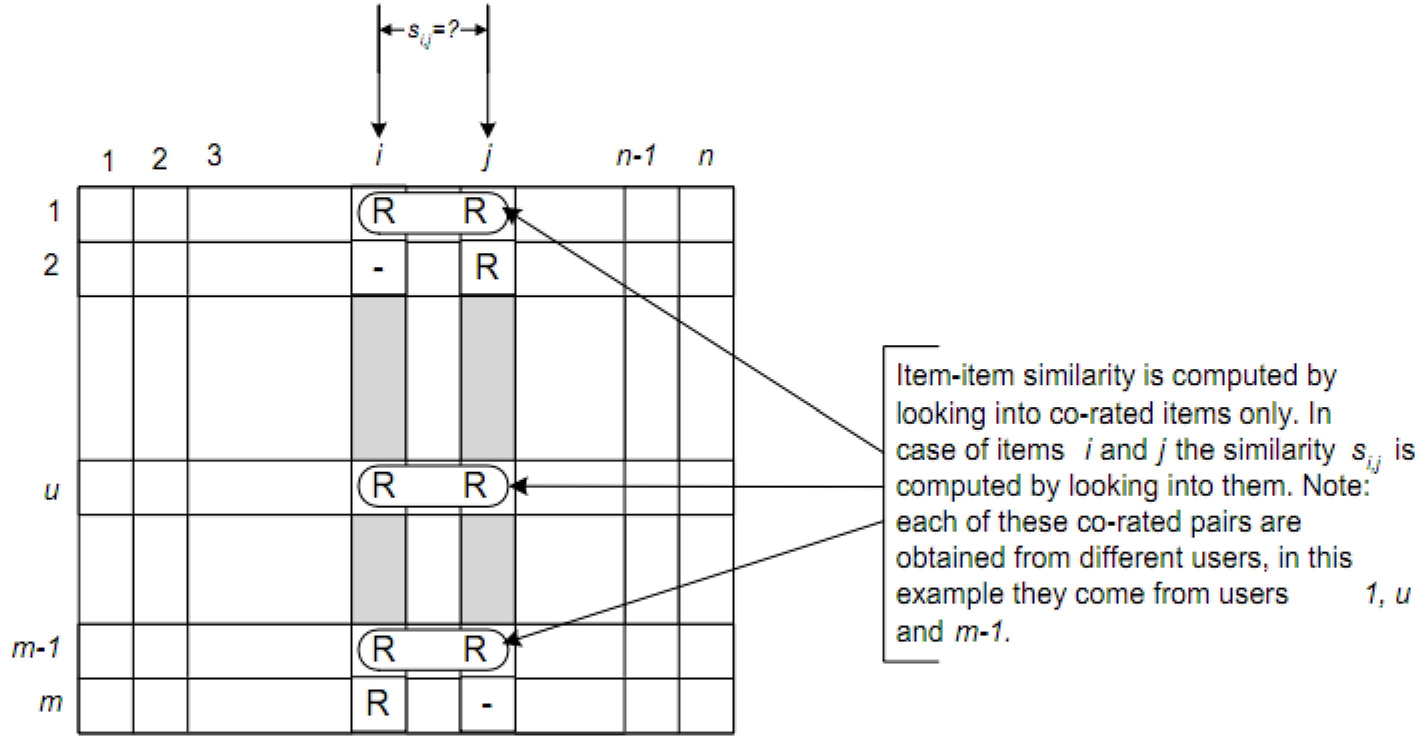


Figure 2: Isolation of the co-rated items and similarity computation

【cos】 上文已经提到了cos，此处的cos和上文的不同在于，user-item矩阵可以是实数型的，而关联矩阵必须是0-1矩阵。 假设A和B是两个item，有 cos(A, B) = A * B / (|A| * |B|)

Sim Metrics—Adjust Cos

【用adjust-cos去掉user-bias】 如果考虑到每user的打分标准都不同，user1喜欢打高分，user2喜欢打低分，应该把用户的打分标准减掉，有 adjust_cos(A, B) =

$$\frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}}.$$

Sim Metrics—Pearson

【用pearson去掉item-bias】如果把A和B的向量归一化去掉bias，再做cos，即pearson的相似度：

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

Sim Metrics—Hybrid

1. 得到user-item matrix打分矩阵
2. 去掉matrix bias
3. 去掉user bias
4. 去掉item bias
5. 做cos

Similarity

Similarity Metrics--Bool or Real		
Bool Matrix Only	**Both**	**Real Matrix Only**
Support	Cos	Adjust cos
Jaccard		Pearson
Confidence		
Lift		
KULC		
LLR		
Conviction		

Similarity Metrics--Symmetry or Not	
Symmetry	**Non Symmetry**
Support	Confidence
Jaccard	
Lift	
KULC	
IR	
Cos	
Adjust Cos	
Pearson	

XX CFs

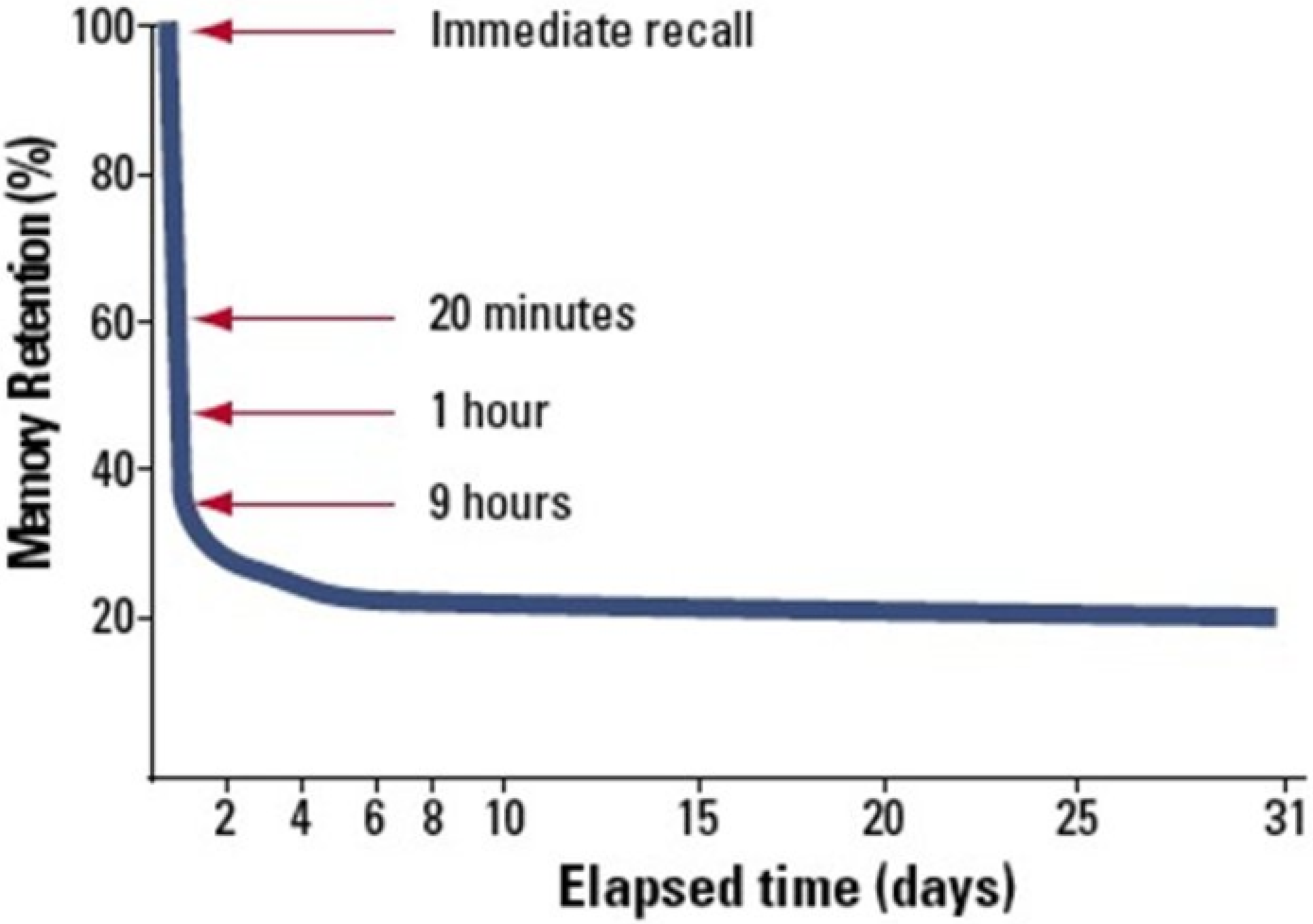
- Item2Vector CF
- Content CF
- LDA CF
- Simhash CF
- Node2Vector CF
- DNN Embedding CF

Refinement Prior

Action Rating

Action	Score
Play	1分
Favor	2分
Download	2分
Switch	-1'分
Trash	-2'分
Comment	3分
Search	2分
Share	2分

Time Decay



100
80
60
40
20
Memory Retention (%)
Immediate recall
20 minutes
1 hour
9 hours
2
4
6
8
10
15
20
25
31
Elapsed time (days)

Session Split

私人电台

最近30分钟无行为

Best Practice

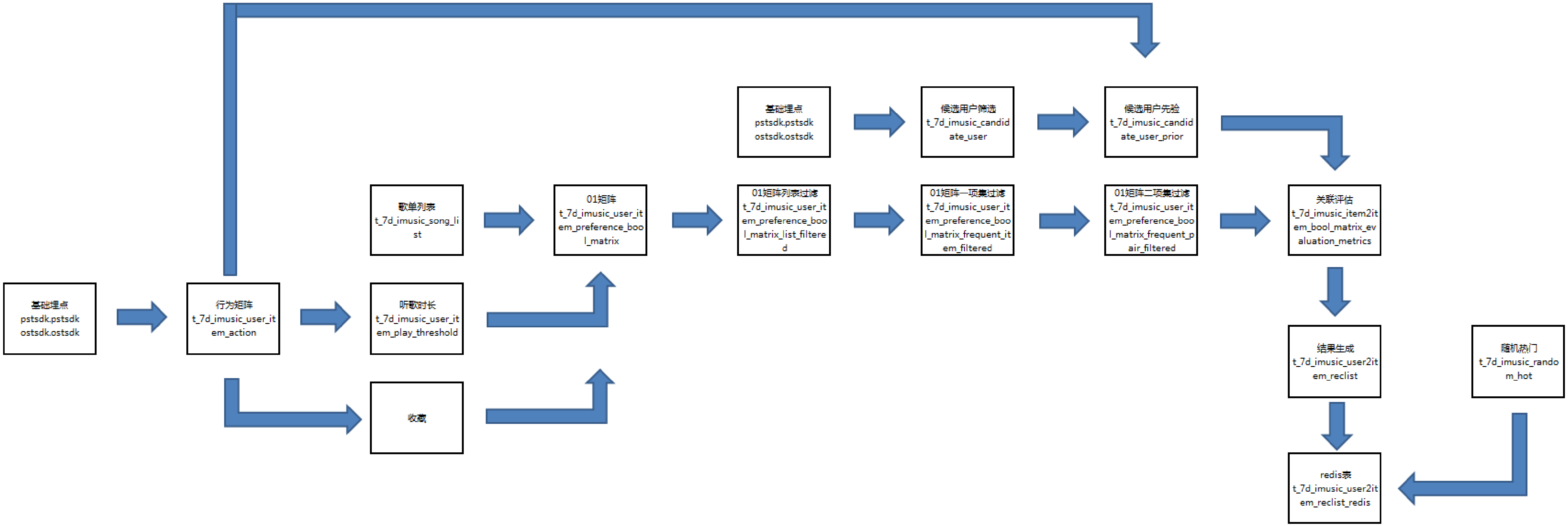
Data Sparsity

(u1, (i1, i2, i3))
(u2, (i1, i3))
(u3, (i2, i4))

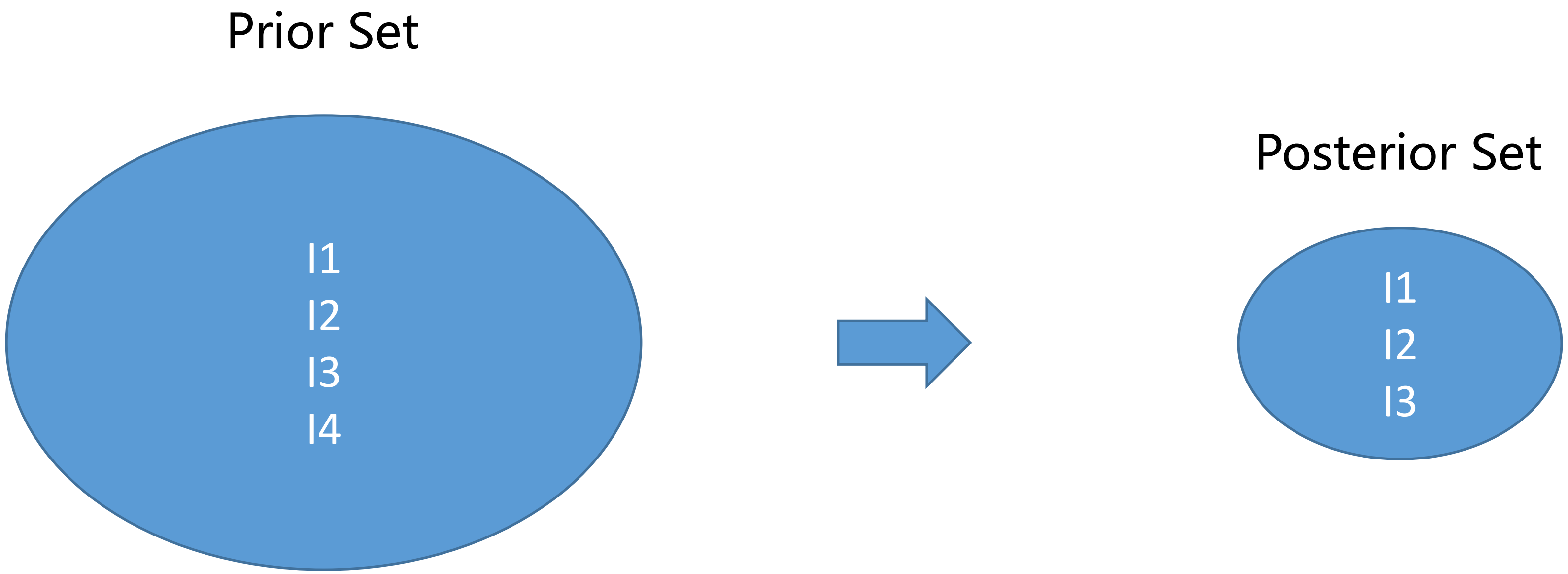
(i1, i2)
(i1, i3)
(i2, i3)
(i1, i3)
(i2, i4)

((i1, i2), 1)
((i1, i3), 2)
((i2, i3), 1)
((i2, i4), 1)

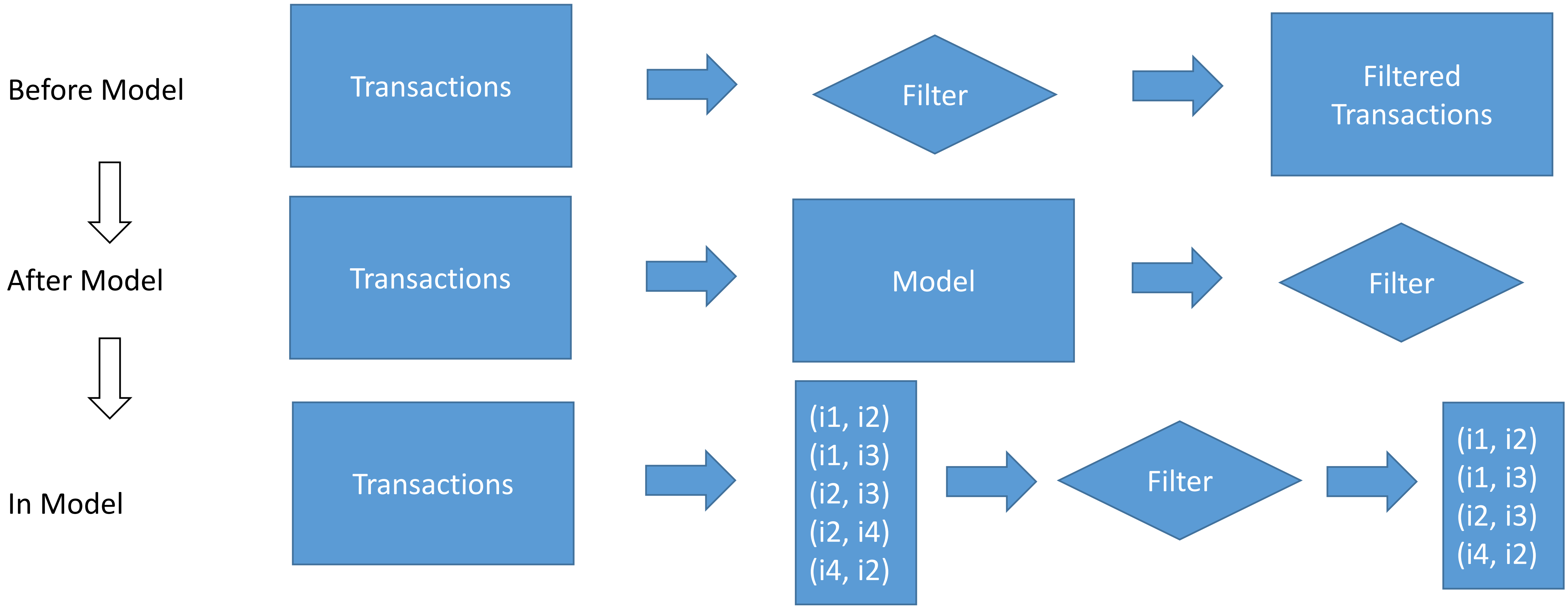
Data Flow


基础埋点
pstsdk.pstsdk
ostsdk.ostsdk
行为矩阵
t_7d_imusic_user_item_action
歌单列表
t_7d_imusic_song_list
听歌时长
t_7d_imusic_user_item_play_threshold
收藏
01矩阵
t_7d_imusic_user_item_preference_bool_matrix
基础埋点
pstsdk.pstsdk
ostsdk.ostsdk
01矩阵列表过滤
t_7d_imusic_user_item_preference_bool_matrix_list_filtered
候选用户筛选
t_7d_imusic_candidate_user
01矩阵一项集过滤
t_7d_imusic_user_item_preference_bool_matrix_frequent_item_filtered
候选用户先验
t_7d_imusic_candidate_user_prior
01矩阵二项集过滤
t_7d_imusic_user_item_preference_bool_matrix_frequent_pair_filtered
关联评估
t_7d_imusic_item2item_bool_matrix_evaluation_metrics
结果生成
t_7d_imusic_user2item_reclist
随机热门
t_7d_imusic_random_hot
redis表
t_7d_imusic_user2item_reclist_redis

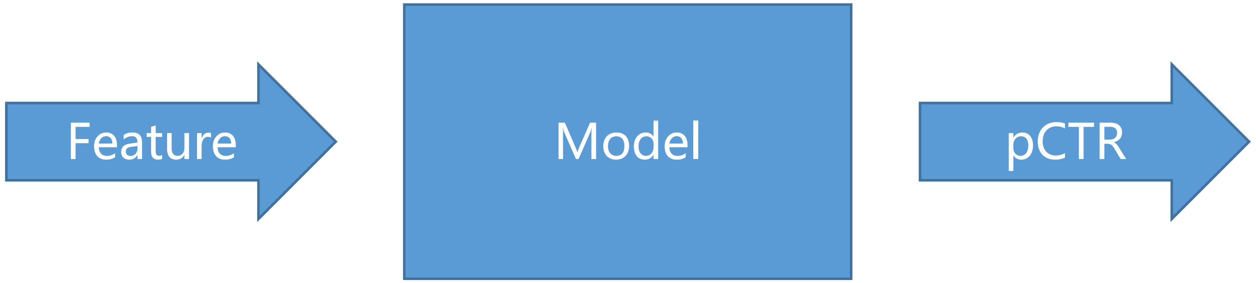
Prior Posterior Diff Set --1



Prior Posterior Diff Set --2



Online



pCTR 实时= Feature 实时 or Model 实时

Q&A