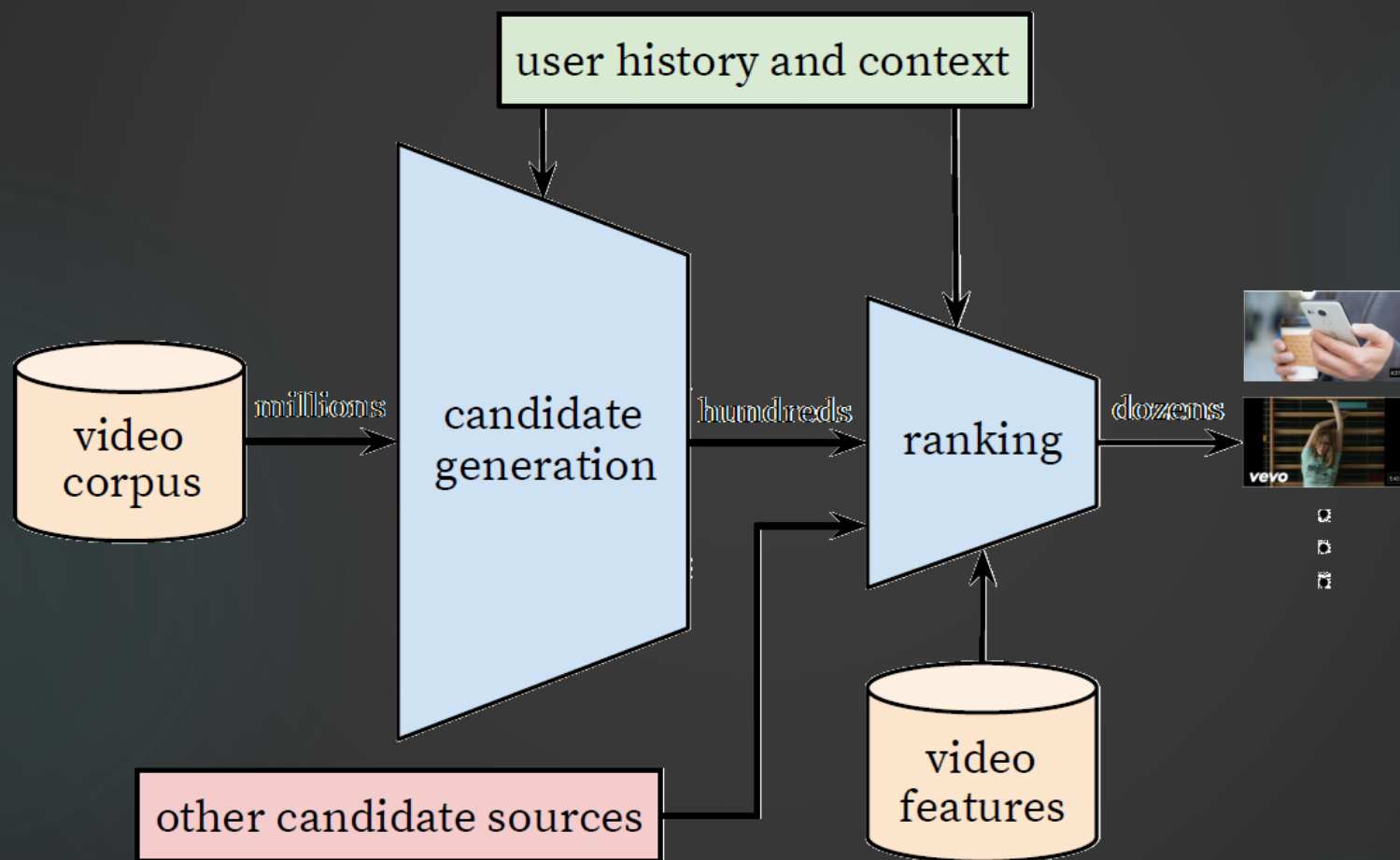


# 推荐系统的深度学习演进路线

丁仑 2018/12/12

# 推荐系统基本结构



# 候选集生成（召回）方法

## 强可解释性

- ▶ 基于运营策略
  - ▶ 用户行为、场景
  - ▶ 从运营角度制定多个策略
  - ▶ 实时性强
- ▶ 基于关联规则（图）
  - ▶ Youtube(2008、2010)
    - ▶ 离线预计算

## 弱可解释性

- ▶ 基于协同过滤
  - ▶ Model-Based
    - ▶ MF, SVD
  - ▶ User/Item-Based
    - ▶ 定义和计算相似度
  - ▶ 自动化程度高，低时间复杂度
- ▶ 基于深度学习
  - ▶ Youtube(2016)
    - ▶ 实时计算

# 排序（重排序）方法

## 传统模型

- ▶ LR
- ▶ MLR
  - ▶ 阿里（2012, 2017）
- ▶ GBDT+LR
  - ▶ Facebook（2014）
- ▶ FM、FFM
  - ▶ 美团（2015），小米
- ▶ 模型集成

## 深度模型

- ▶ Wide and Deep
  - ▶ Google, 2016
- ▶ DeepFM
  - ▶ 华为（2017）
- ▶ xDeepFM
  - ▶ 微软（2018）
- ▶ 深度强化学习
  - ▶ 美团（2018）

# 一种可行的技术演进路线



**MF**

通过Model-Based的协同过滤实现海量商品快速召回，利用离线跑批的形式完成个性化推荐。



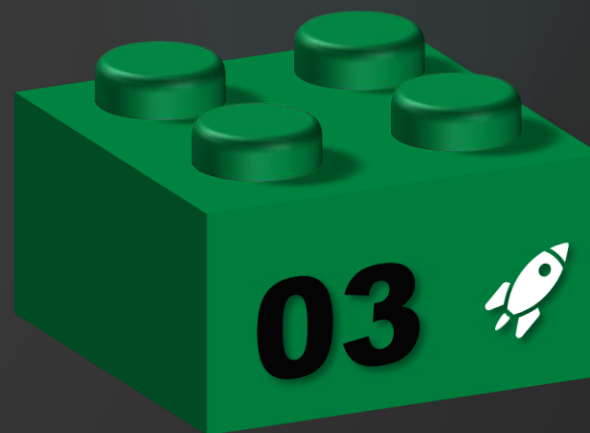
**FM**

以MF为基础，演进到基于FM的排序模型。根据用户近期行为对已找回商品进行精细CTR预测。



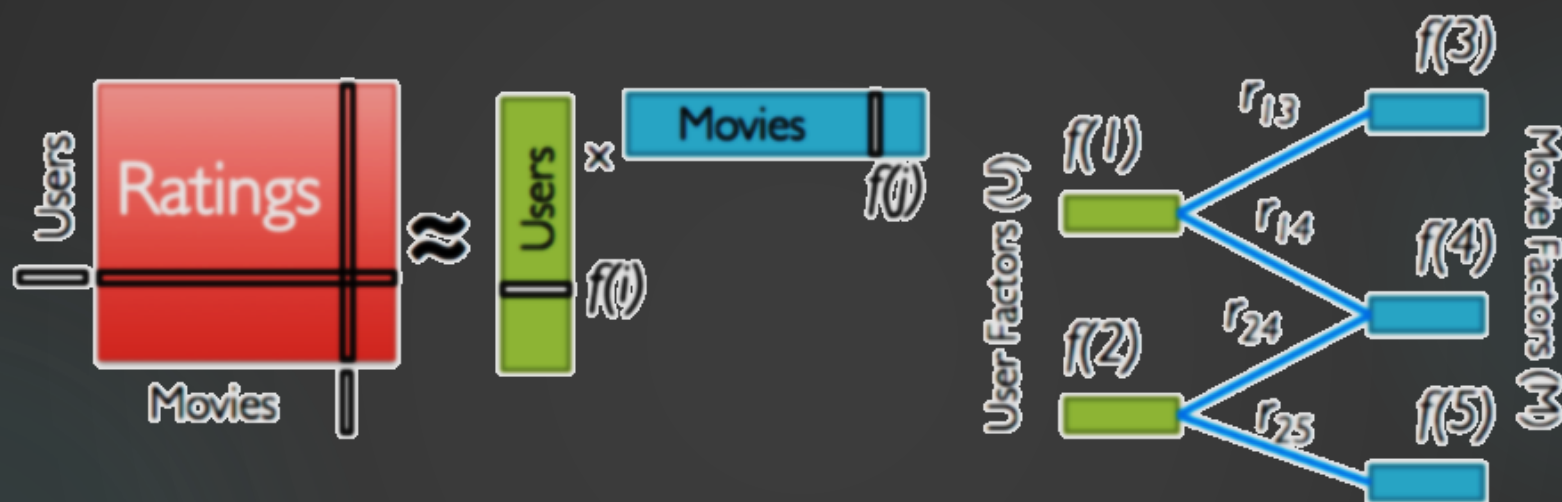
**DeepFM**

以FM为基础，利用Wide&Deep架构将排序模型演进至基于DeepFM算法的深度学习架构



# MF

## Low-Rank Matrix Factorization:



Iterate:

$$f[i] = \arg \min_{w \in \mathbb{R}^d} \sum_{j \in \text{Nbrs}(i)} (r_{ij} - w^T f[j])^2 + \lambda \|w\|_2^2$$

Taken from the BerkeleyX Course Big Data Analysis with Spark

# MF的计算、优化算法

## 交替最小二乘法（ALS）

- ▶ 以隐反馈为例
- ▶ 隐反馈的损失函数

$$\min_{x, y} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left( \sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

- ▶ 注意到是加权最小二乘， $c_{ui}$ 为权重。某种程度上等价于逻辑回归，将隐反馈问题转化为ALS可解的0、1 Binary问题。
- ▶ 随机初始化 $y_i$ ，通过最小二乘法计算 $x_u$
- ▶ 根据 $x_u$ ，反向计算 $y_i$
- ▶ 可并行。计算速度快，收敛速度快。

## Online Learning

- ▶ 采用随机梯度下降方案
- ▶ 对于最小二乘问题（显、隐反馈）
  - ▶ 随机初始化 $x_u$ 与 $y_i$
  - ▶ 对于每个已有样本计算梯度
  - ▶ 同时迭代 $x_u$ 与 $y_i$ ，经过数个epoch收敛
- ▶ 对于转化为Logit回归问题（隐反馈）
  - ▶ 对评分做Logit变换
  - ▶ 迭代方法同上



# 一种可行的技术演进路线



**MF**

通过Model-Based的协同过滤实现海量商品快速召回，利用离线跑批的形式完成个性化推荐。



**FM**

以MF为基础，演进到基于FM的排序模型。根据用户近期行为对已找回商品进行精细CTR预测。



**DeepFM**

以FM为基础，利用Wide&Deep架构将排序模型演进至基于DeepFM算法的深度学习架构





# 为什么用FM



## 可扩展性

- 可作为Embedding层，解决特征高维稀疏性问题。
- 经过FM加工过的高维稀疏特征可作为输入层给树类模型使用
- 可增加Deep层。如基于Wide&Deep实现深度模型。



## 低时间复杂度

- 化简后具有 $O(kn)$ 的训练和预测复杂度
- 良好的速度在工程应用中极为重要



## 传承LR



- LR是FM的一种特殊形式
- FM实现了LR的自动二次交叉，免去了耗时的人工特征提取
- 一种二阶特征组合进行自动学习的模型

## 传承MF



- MF是FM的一种特殊形式
- 采用随机梯度下降优化的MF可以轻松的优化梯度公式从而实现在线更新的FM
- 简化后亦可以实现过程的在线更新化

# 为什么不使用LR

- 线性方程的局限性，直接导致了LR时代需要非常重的人工特征干预

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^p w_j x_j$$

- 对于未出现的特征组合通用性不佳
- FM通过Embedding解决这些问题

# FM与LR、MF的传承关系

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \sum_{f=1}^k v_{j,f} v_{j',f}$$

Feature vector $\mathbf{x}$																				Target $y$	
$\mathbf{x}_1$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5 $y_1$
$\mathbf{x}_2$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3 $y_2$
$\mathbf{x}_3$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1 $y_3$
$\mathbf{x}_4$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4 $y_4$
$\mathbf{x}_5$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5 $y_5$
$\mathbf{x}_6$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1 $y_6$
$\mathbf{x}_7$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5 $y_7$
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...	
	User				Movie					Other Movies rated						Last Movie rated					

# 一种可行的技术演进路线



**MF**

通过Model-Based的协同过滤实现海量商品快速召回，利用离线跑批的形式完成个性化推荐。



**FM**

以MF为基础，演进到基于FM的排序模型。根据用户近期行为对已找回商品进行精细CTR预测。



**DeepFM**

以FM为基础，利用Wide&Deep架构将排序模型演进至基于DeepFM算法的深度学习架构



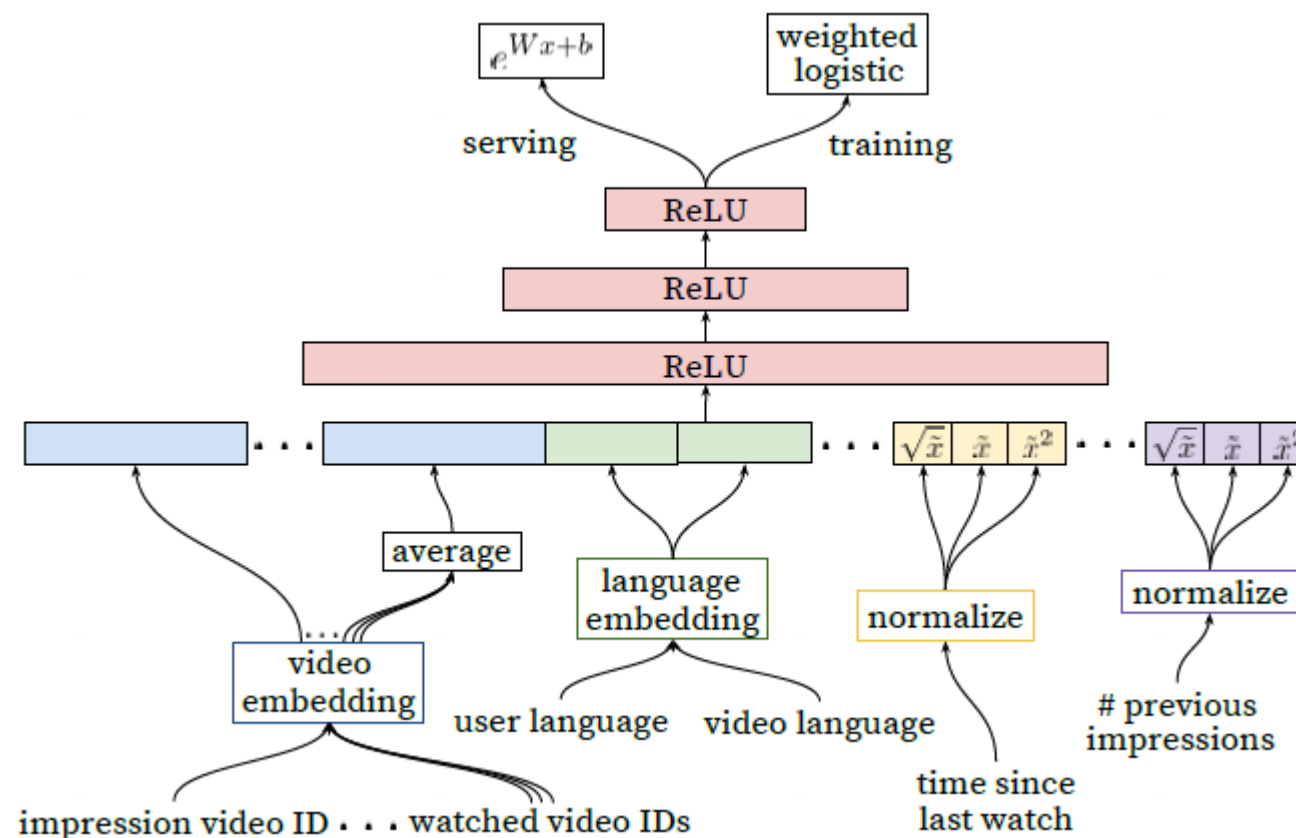
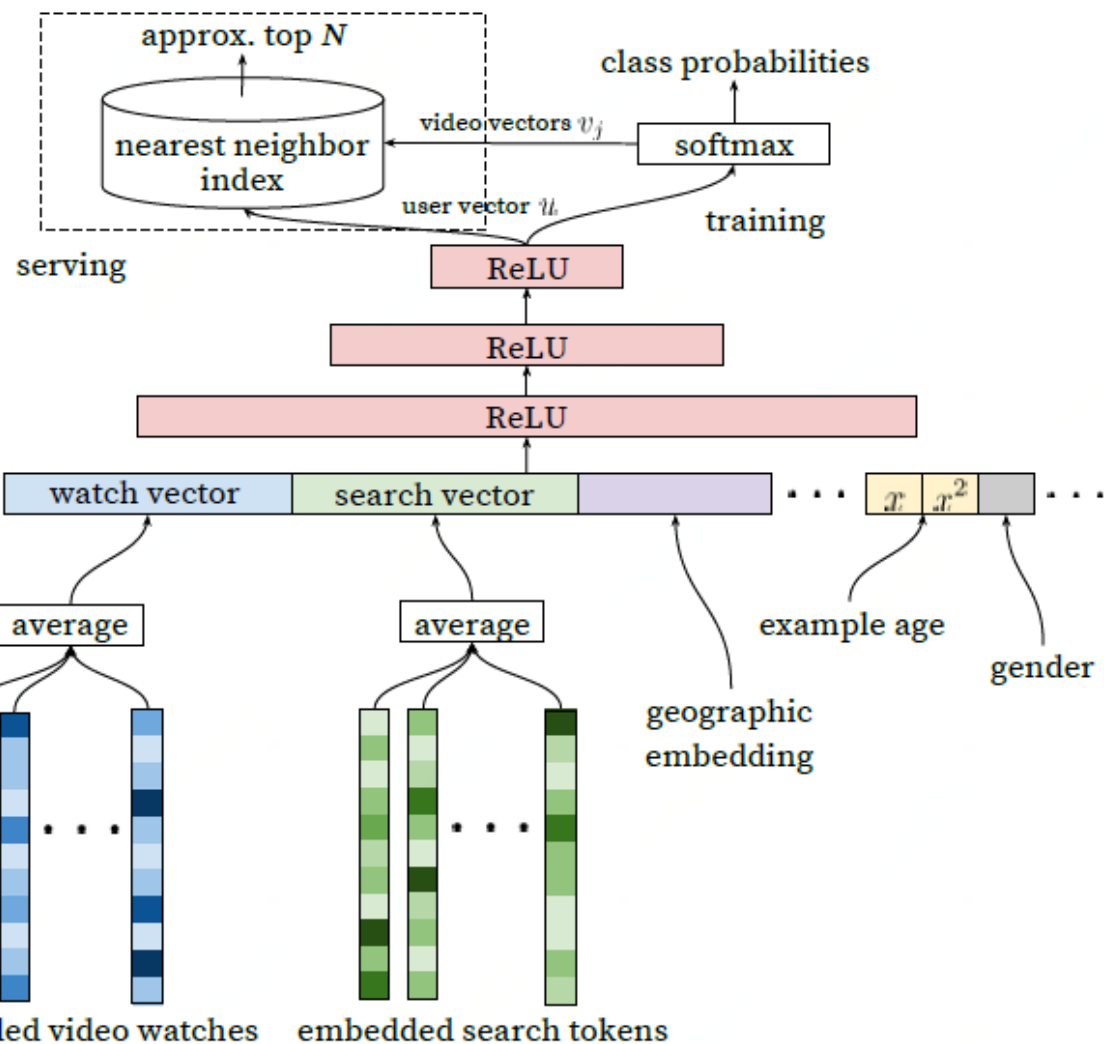
# 算法向深度学习的演进

- ▶ 为什么使用Wide&Deep
- ▶ DeepFM的结构
- ▶ DeepFM的自交性
- ▶ DeepFM的衍生xDeepFM
- ▶ 深度学习的高计算复杂度导致性能问题

# YouTube的深度学习解决方案

- ▶ Label Partitioning For Sublinear Ranking这篇文章的意义
- ▶ YouTube如何在召回阶段实现深度学习
- ▶ 其他降低复杂度的方法（Multi-Task Rocket Training，阿里2017）

# YouTube基于深度学习的召回和排序

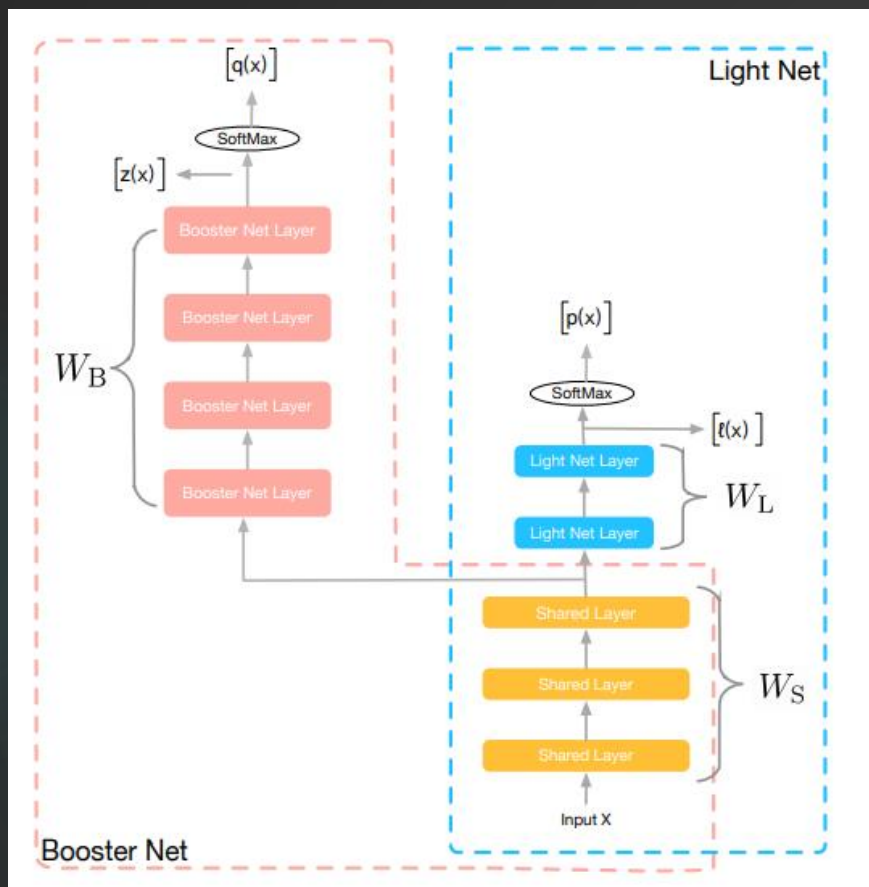




# 如何在召回阶段使用深度学习

1. Given a test input  $x$ , the input partitioner maps  $x$  to a set of partitions  $p = g(x)$ .
2. We retrieve the label sets assigned to each partition  $p_j$ :  $L = \cup_{j=1}^{|p|} \mathcal{L}_{p_j}$ , where  $\mathcal{L}_{p_j} \subseteq \mathcal{D}$  is the subset of labels assigned to partition  $p_j$ .
3. We score the labels  $y \in L$  with the label scorer  $f(x, y)$ , and rank them to produce our final result.

# 阿里在召回阶段的复杂度降低方案



The End