百度大规模推荐系统实践

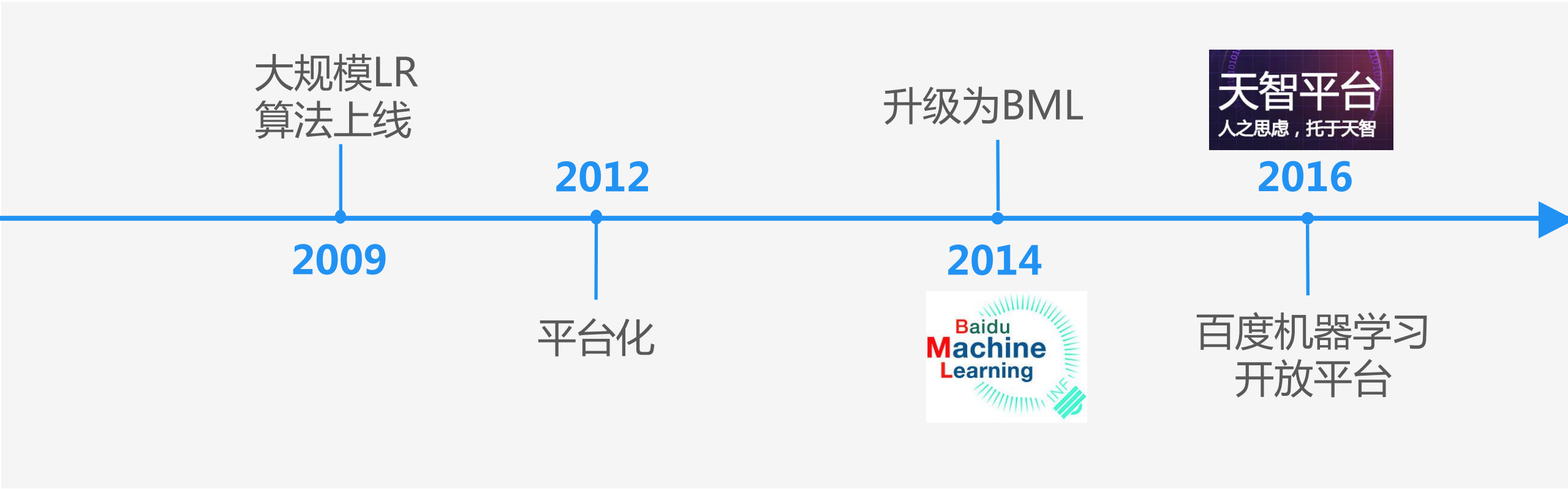
百度基础架构部

秦铎浩

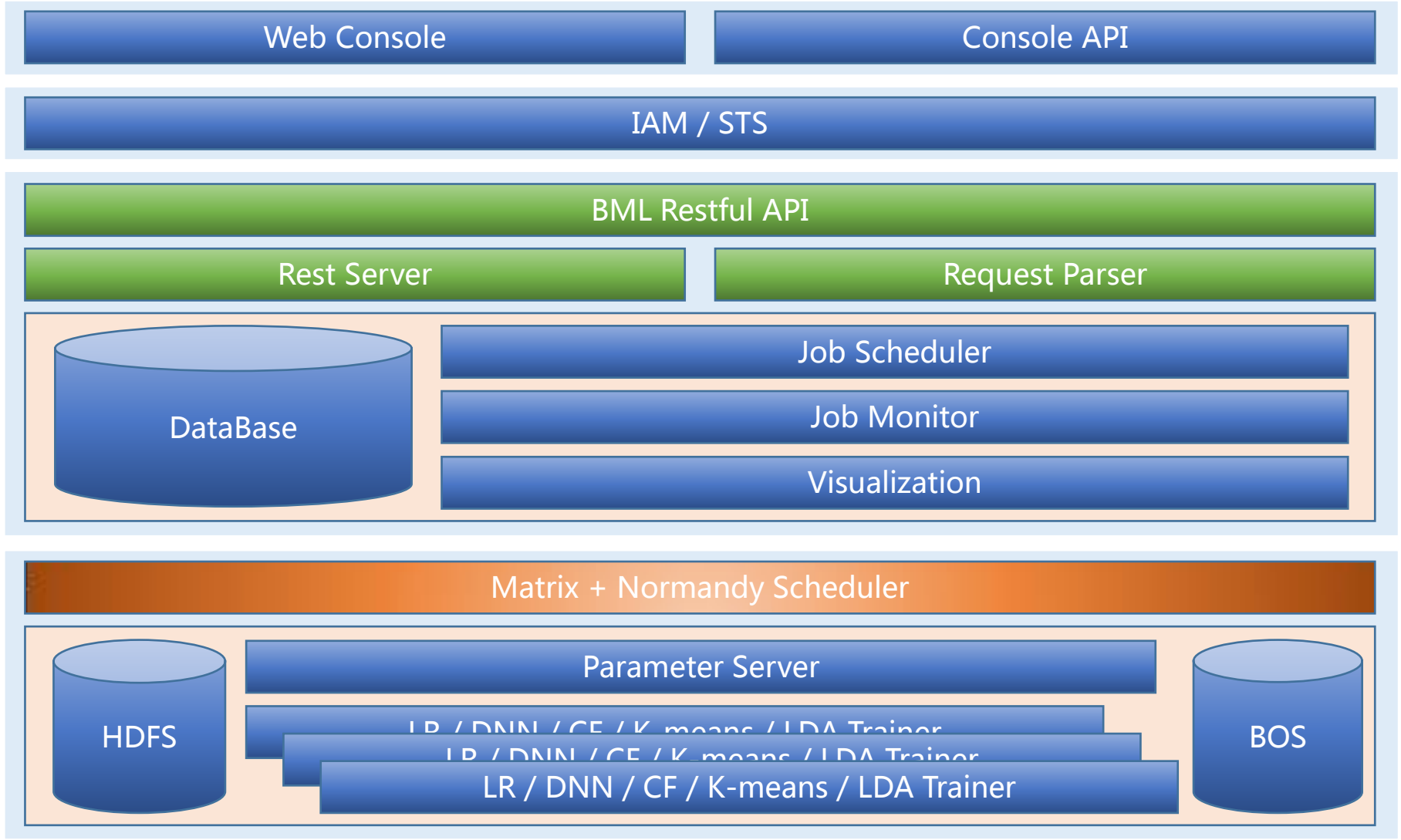
目录

- 百度机器学习开放平台历程
- 云端上的BML
- 大规模推荐系统实践

百度机器学习开放平台发展



平台整体架构


Web Console
Console API
IAM / STS
BML Restful API
Rest Server
Request Parser
DataBase
Job Scheduler
Job Monitor
Visualization
Matrix + Normandy Scheduler
HDFS
Parameter Server
LR / DNN / CF / K-means / LDA Trainer
BOS

Next

- 百度机器学习开放平台历程
- 云端上的BML
- 大规模推荐系统实践

云端上的BML

百度云

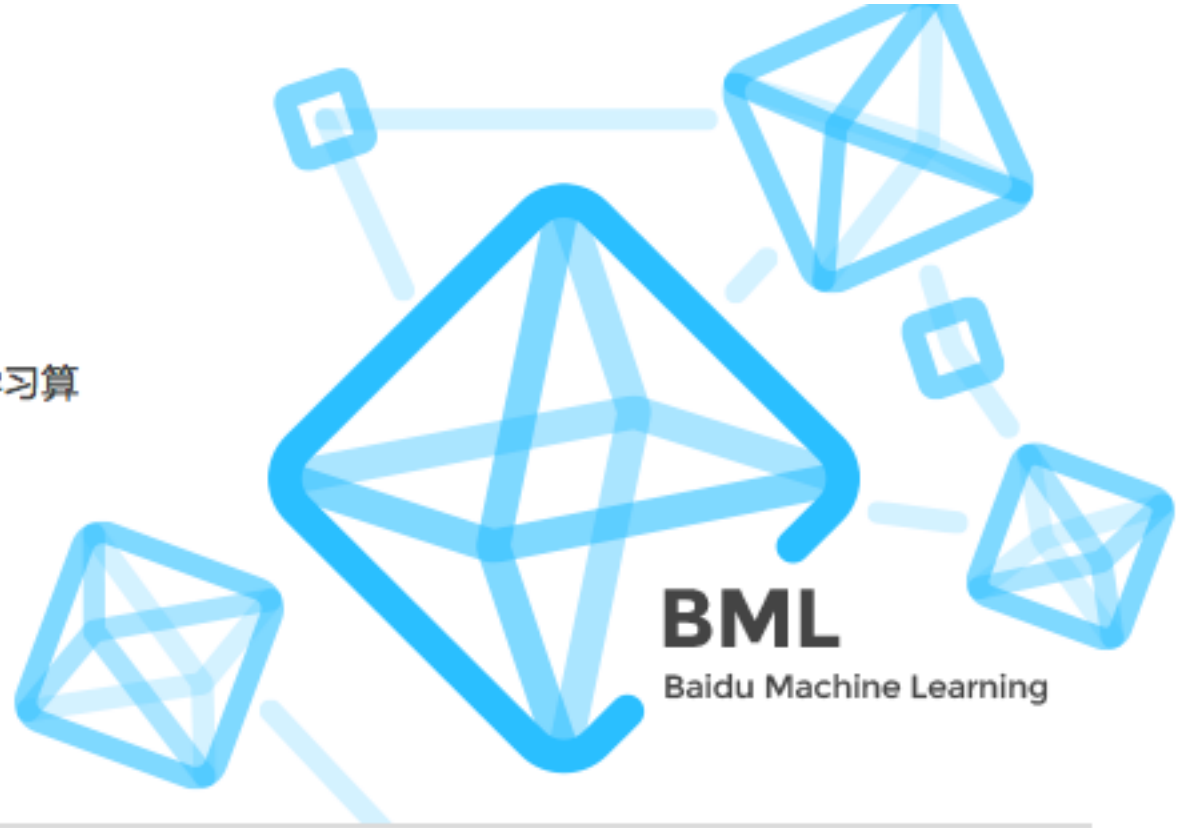
百度机器学习 BML

Baidu Machine Learning

百度自主研发的新一代机器学习平台，基于百度内部应用多年的机器学习算法库，提供实用的行业大数据解决方案

立即申请公测

帮助文档 >



https://cloud.baidu.com/

云端上的BML

百度云

预处理

生成数据集

分类算法

推荐算法

聚类算法

主题模型

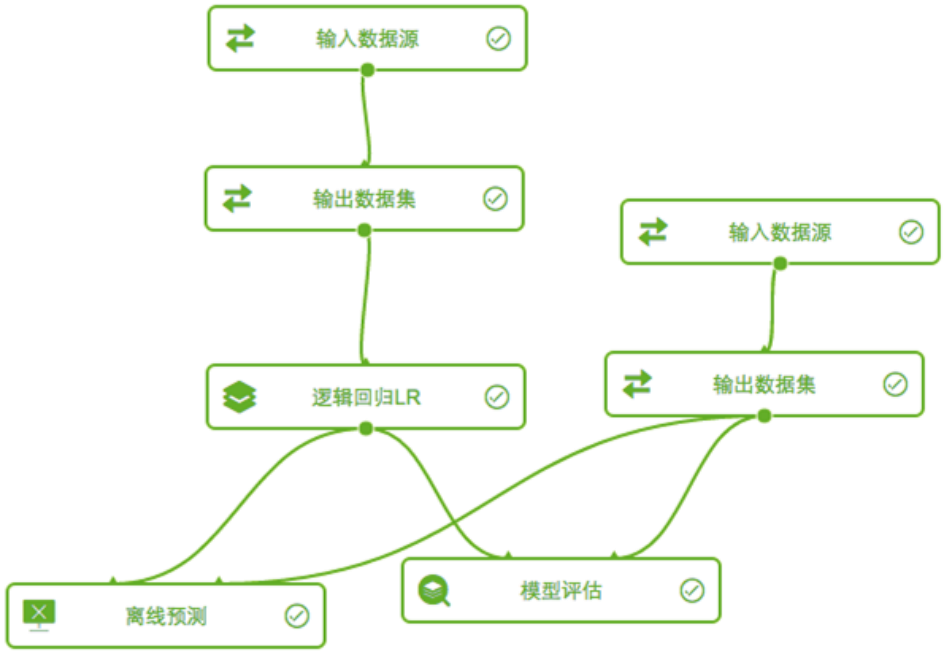
神经网络

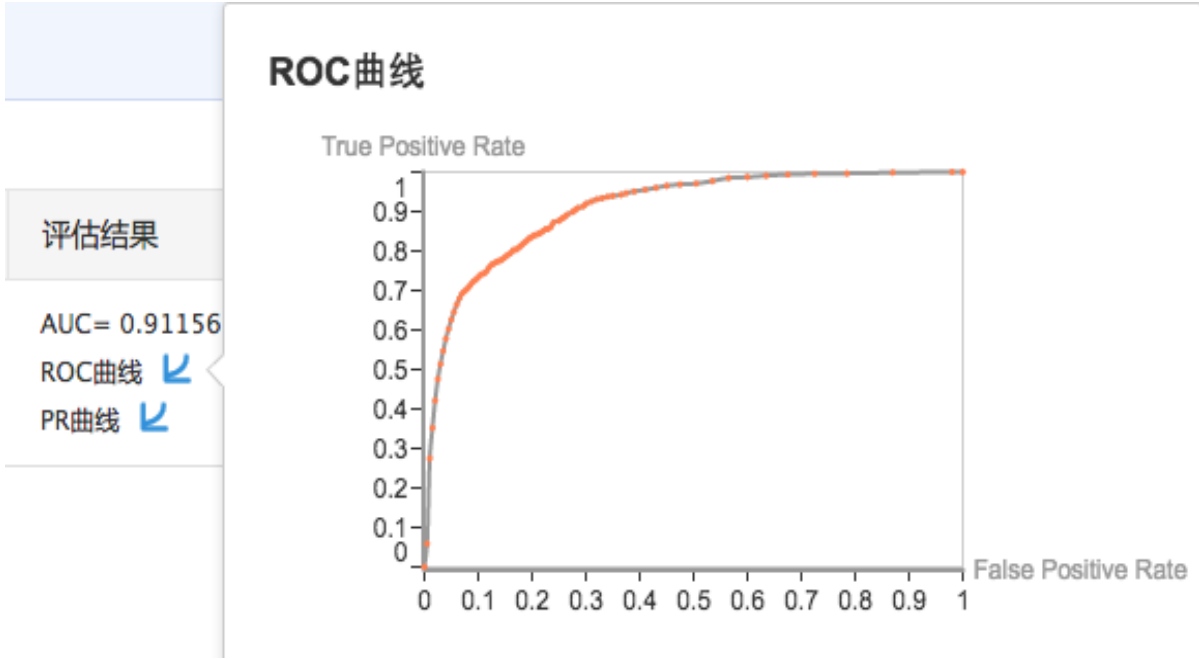
在线学习

效果评估

可视化

云端上的BML


输入数据源
输出数据集
逻辑回归LR
离线预测
输入数据源
输出数据集
模型评估


评估结果
AUC= 0.91156
ROC曲线
PR曲线
ROC曲线
True Positive Rate
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
False Positive Rate

Next

- 百度机器学习开放平台历程
- 云端上的BML
- 大规模推荐系统实践

推荐系统必备

百度云

干净的数据

高效的算法

完善的系统

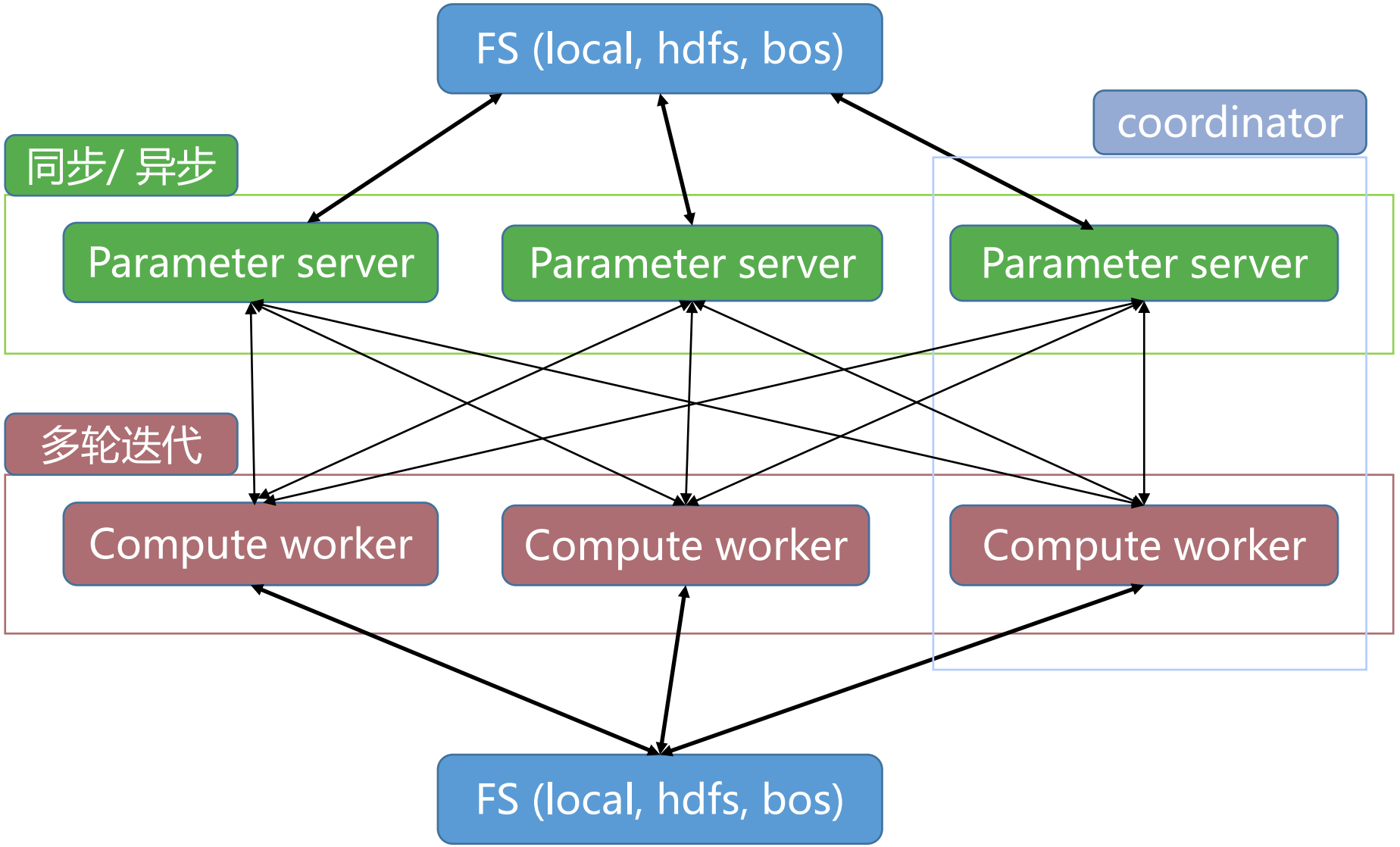
业务场景

菜品推荐

新闻推荐

搜索推荐

ELF (Essential Learning Framework)

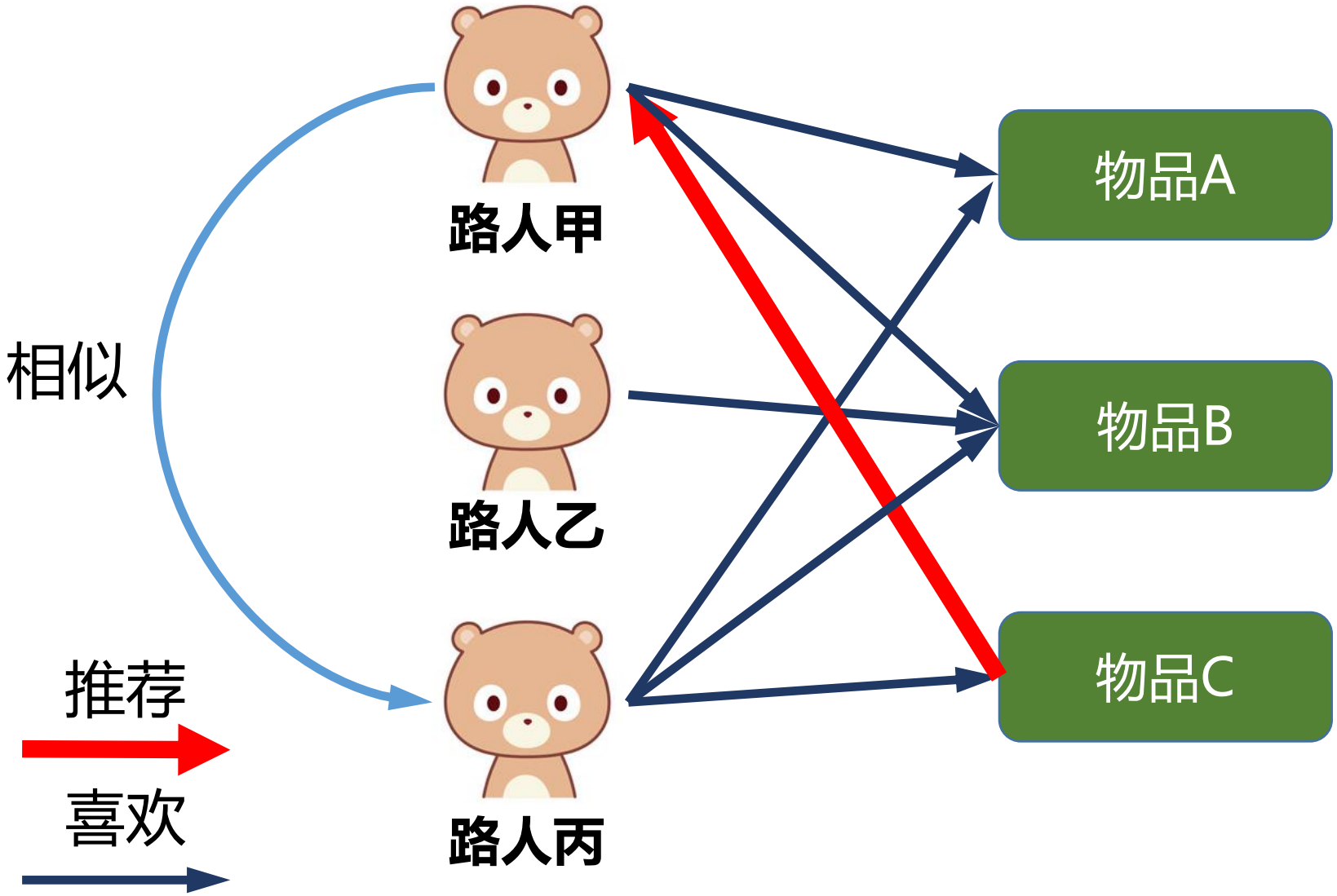


ELF (Essential Learning Framework)

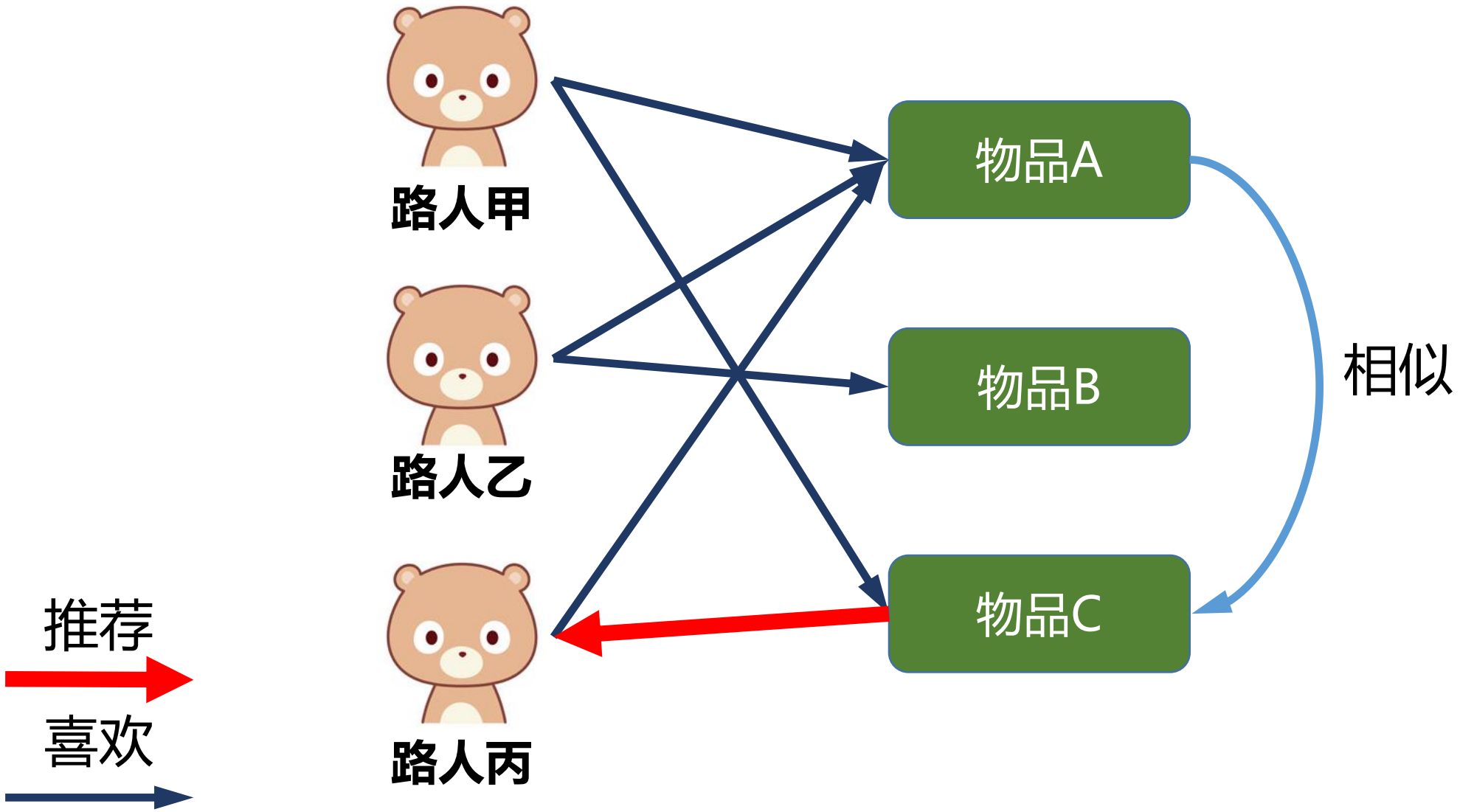
百度云

- 简单易用, 编写Async SGD LR仅需要200行代码
- 组件分布式多线程实现, 支持细粒度的线程控制
- 节点间通信依赖高效的baidu-rpc
- 深度优化hashtable, 专用于Parameter Server

协同过滤-基于user的CF



协同过滤-基于item的CF



协同过滤

- 收集用户偏好 ← 隐式/显式反馈
- 找到相似的用户或物品 ←

Cosine相似度

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i}| \times |\vec{j}|}$$

欧几里得距离

$$d(i,j) = |\vec{i} - \vec{j}|$$

$$sim(i,j) = \frac{1}{1+d(i,j)}$$

- 计算推荐 ←

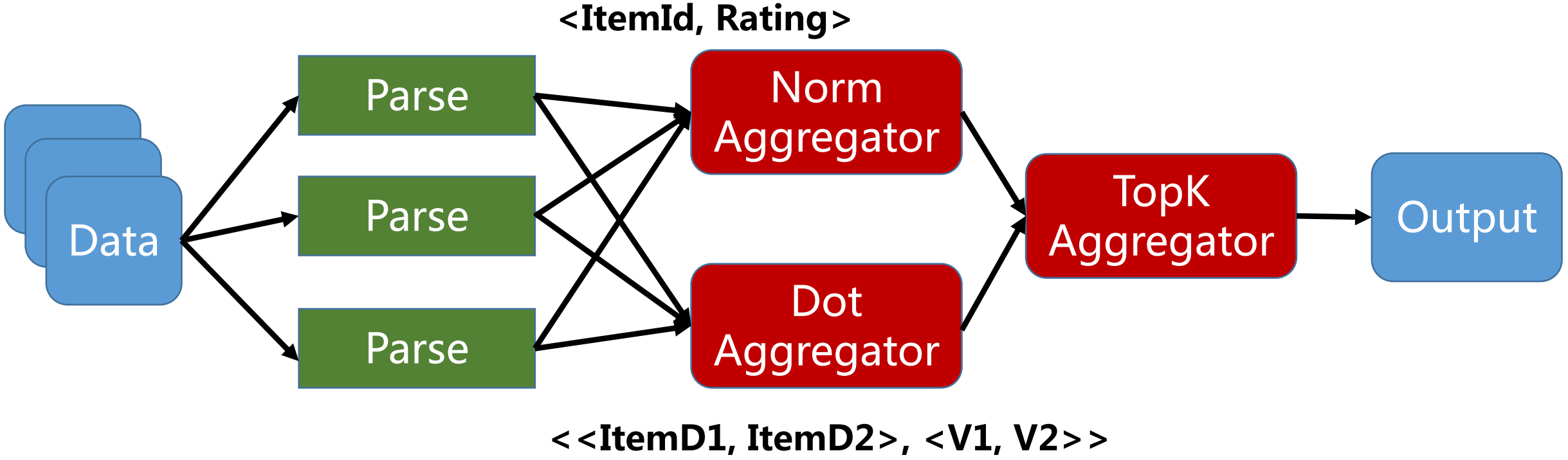
$$P_{u,i} = \frac{\sum_{all\ similar\ items,\ N}(S_{i,N} * R_{u,N})}{\sum_{all\ similar\ items,\ N}(|S_{i,N}|)}$$

协同过滤

	user1	user2	user3	user4	user5	user6
item1	5.0	2.0	2.0	5.0	4.0	0
item2	0	0	5.0	0	0	4.0
item3	3	2.5	0	0	3.0	0
item4	2.5	0	0	3.0	2.0	2.0
item5	0	2.0	4.0	4.5	0	0
item6	0	0	0	0	3.5	3.5
item7	0	0	5.0	0	0	4.0

- 当item数量很大时，计算量会大到难以计算
- 没有利用到矩阵稀疏的性质

协同过滤

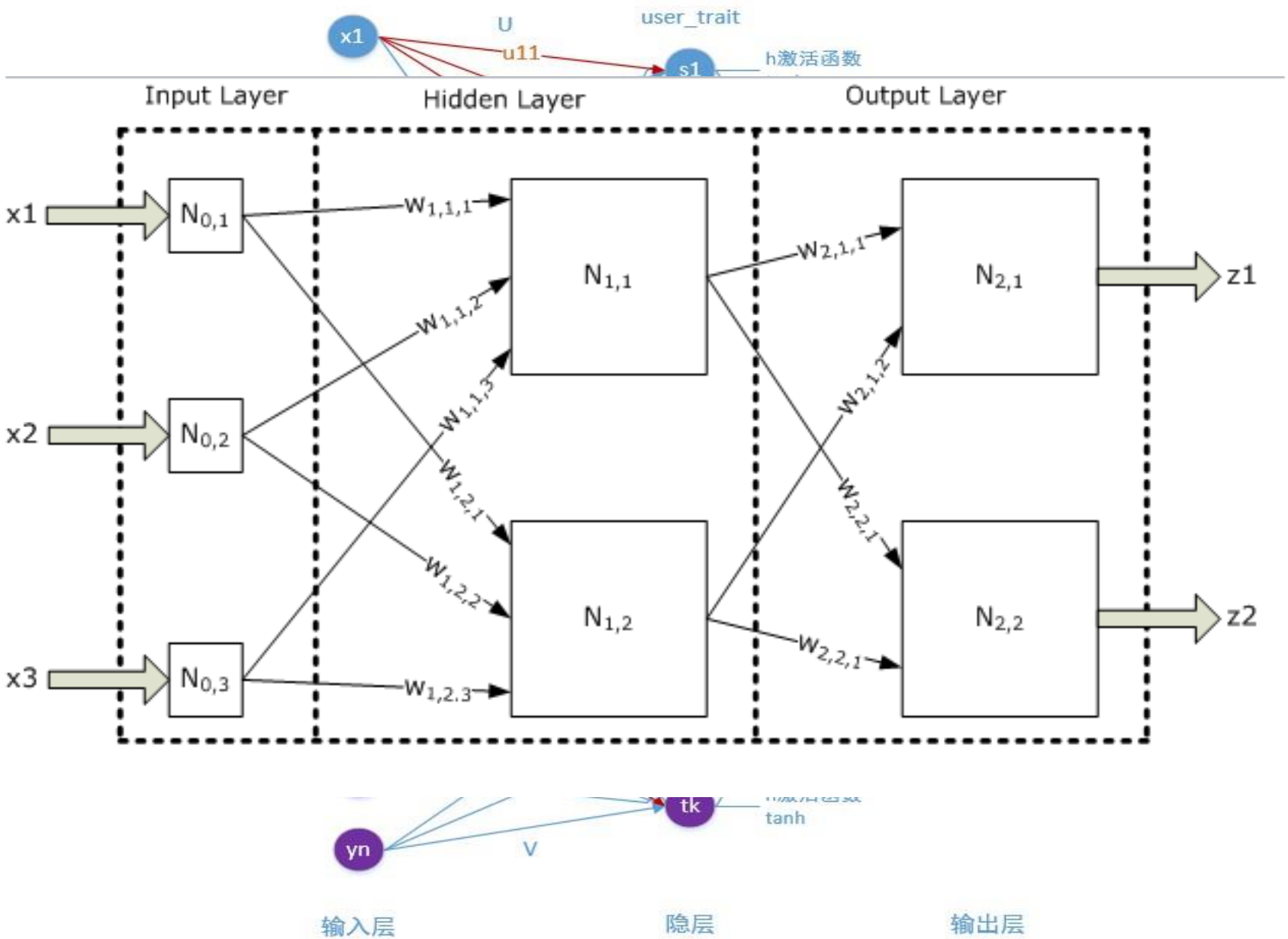


$$\cos(\vec{i},\vec{j}) = \frac{\text{rating}_{u_1 i} \times \text{rating}_{u_1 j} + \cdots + \text{rating}_{u_n i} \times \text{rating}_{u_n j}}{\sqrt{\text{rating}_{u_1 i}{}^2 + \cdots + \text{rating}_{u_n i}{}^2}\sqrt{\text{rating}_{u_1 j}{}^2 + \cdots + \text{rating}_{u_n j}{}^2}}$$

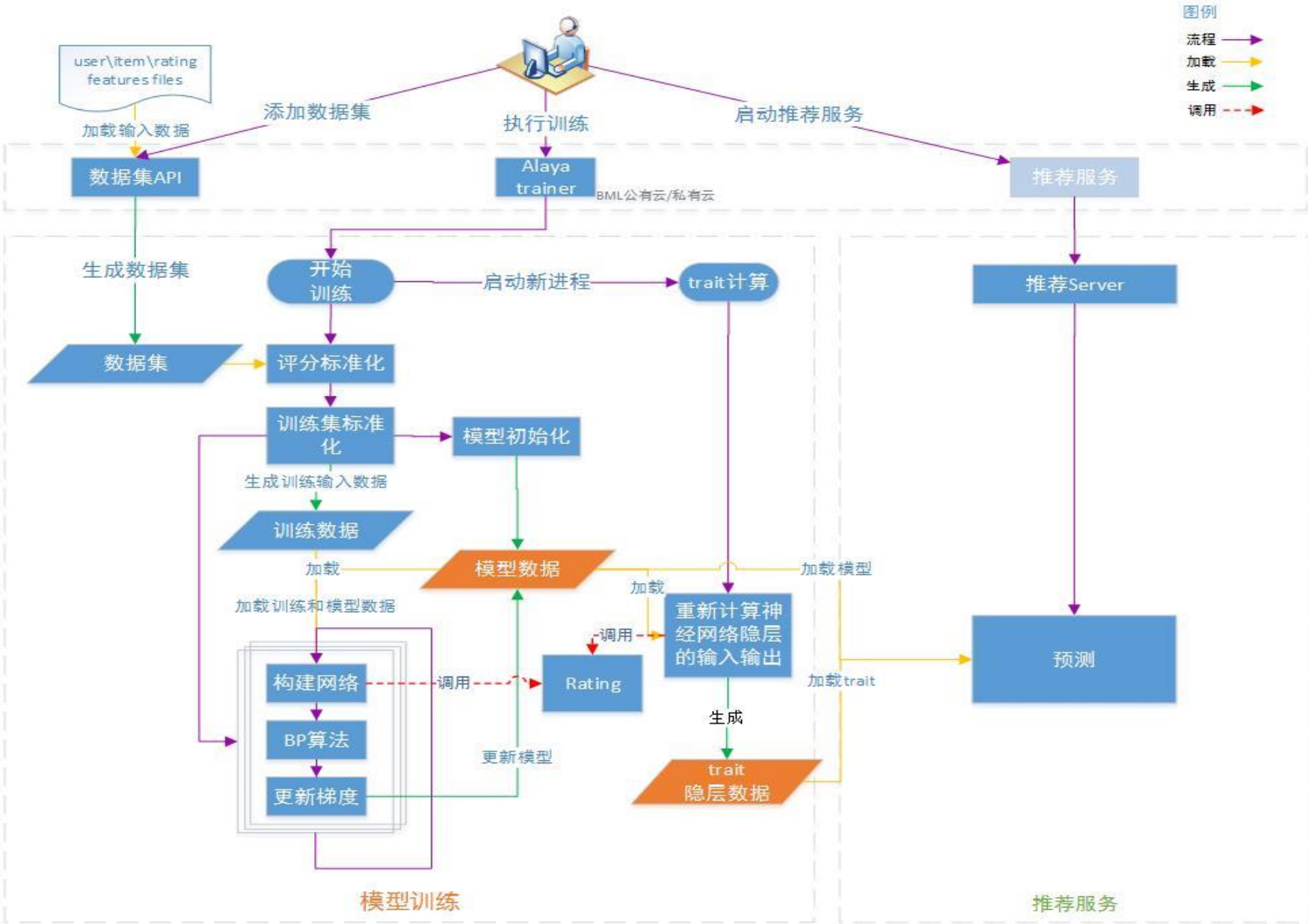
Alaya推荐

- 基于深度学习的推荐系统
- 支持基于用户和商品的协同过滤
- 可以使用上更多的用户信息
- 并行分布式训练

Alaya训练


x1
U
user_trait
u11
s1
h激活函数
Input Layer
Hidden Layer
Output Layer
x1
x2
x3
N0,1
N0,2
N0,3
w1,1,1
w1,1,2
w1,1,3
w1,2,1
w1,2,2
w1,2,3
N1,1
N1,2
w2,1,1
w2,1,2
w2,2,1
w2,2,1
N2,1
N2,2
z1
z2
yn
V
tk
tanh
输入层
隐层
输出层

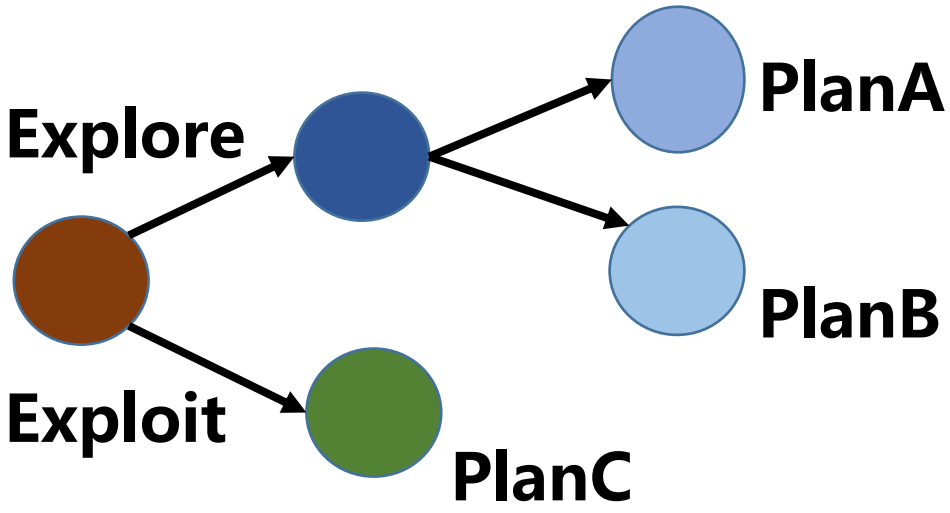
Alaya整体架构


图例
流程
加载
生成
调用
user\item\rating features files
加载输入数据
添加数据集
执行训练
启动推荐服务
数据集API
Alaya trainer
BML公有云/私有云
推荐服务
生成数据集
开始训练
启动新进程
trait计算
推荐Server
数据集
评分标准化
训练集标准化
模型初始化
生成训练输入数据
训练数据
加载
模型数据
加载模型
加载
加载训练和模型数据
重新计算神经网络隐层的输入输出
调用
构建网络
调用
Rating
预测
加载trait
BP算法
生成
更新模型
trait 隐层数据
更新梯度
模型训练
推荐服务

Exploration and Exploitation

假设有一个广告资源池，每个广告的收益都服从一个未知的分布。怎么知道该给每个用户展示哪个广告，从而获得最大的点击收益？

- **Epsilon-Greedy方法**



- **UCB方法**
  - 先在前K轮每个Plan都执行一次
  - 在后面的N轮，选值最大的那个

$$\bar{x}_j + \sqrt{\frac{2 \ln n}{n_j}}$$

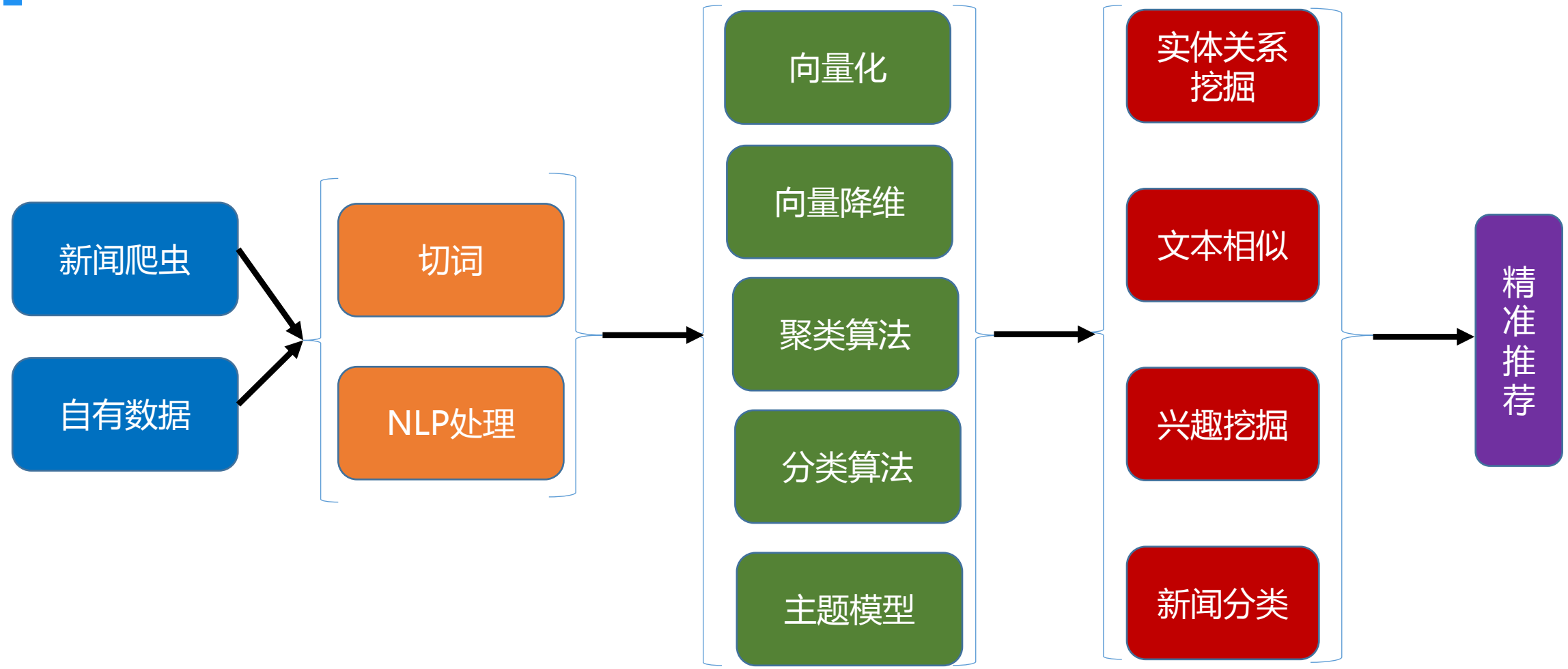
Exploration and Exploitation

- **LinUCB**

$$a_t \stackrel{\text{def}}{=} \arg\max_{a \in \mathcal{A}_t} \left(\mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}} \right)$$

- **Thompson sampling**

新闻推荐


新闻爬虫
自有数据
切词
NLP处理
向量化
向量降维
聚类算法
分类算法
主题模型
实体关系挖掘
文本相似
兴趣挖掘
新闻分类
精准推荐

THANK YOU

cloud.baidu.com

email:qinduohao@baidu.com