**Thinking1 ALS都有哪些应用场景**

ALS属于数据挖掘, 可以做推荐系统, 比如电影推荐, 商品推荐, 广告推荐等。

原理就是给各个指标,判定等加权重,然后将这些训练集输入ALS,包括其他的参数,内部进行矩阵相乘,根据这些权重,给用户对未知,未点击的商品也给一个分数,就是喜好程度. 然后把喜好程度高的商品推荐给用户,假如用户不喜欢,从线上观察效果不好,那这个模型就有问题,需要修改参数,修改权重,或者添加权重,使之达到一个理想的效果。

推荐系统的两大应用场景：

评分预测（Rating Prediction）

主要用于评价网站，比如用户给自己看过的电影评多少分（MovieLens），或者用户给自己看过的书籍评价多少分（Douban）。矩阵分解技术主要应用于评分预测问题。

Top-N推荐（Item Ranking）

常用于购物网站，拿不到显式评分，通过用户的隐式反馈为用户提供一个可能感兴趣的Item列表。排序任务，需要排序模型进行建模。

**Thinking2 ALS进行矩阵分解的时候，为什么可以并行化处理**

ALS, Alternative Least Square, ALS，交替最小二乘法

Step1，固定Y 优化X

Step2，固定X 优化Y

重复Step1和2，直到X 和Y 收敛。每次固定一个矩阵，优化另一个矩阵，都是最小二乘问题。

当固定Y后，迭代更新X时每个X只依赖自己，不依赖于其他的标的物的特征向量，所以可以将不同的X的更新放到不同的服务器上执行。同理，当X固定后，迭代更新Y时每个Y只依赖自己，不依赖于其他用户的特征向量，一样可以将不同用户的更新公式放到不同的服务器上执行。Spark的ALS算法就是采用这样的方式做到并行化的。

**Thinking3 梯度下降法中的批量梯度下降（BGD），随机梯度下降（SGD），和小批量梯度下降有什么区别（MBGD）**

* **批量梯度下降：**
* **在每次更新时用所有样本**
* **稳定，收敛慢**
* **随机梯度下降**
* **每次更新时用1个样本，用1个样本来近似所有的样本**
* **更快收敛，最终解在全局最优解附近**
* **mini-batch梯度下降**
* **每次更新时用b个样本，折中方法**
* **速度较快**

**Thinking4 你阅读过和推荐系统/计算广告/预测相关的论文么？有哪些论文是你比较推荐的，可以分享到微信群中**

**推荐这篇论文：**This paper proposed a meta-learning-based approach to recommendation, which aims to select the best algorithm for each user-item pair.

<http://aiai.ucd.ie/aics2018/papers/aics_20.pdf>