Bandit算法

Bandit算法的起源是多臂赌博机问题（Multi-armed bandit problem, K-armed bandit problem, MAB），其描述是有K台赌博机，选择哪台赌博机收益最大，即决策问题。常用于推荐算法的两个问题上，第一个是EE问题，即exploit-explore问题，用于权衡保守策略与探索策略；第二个是用户冷启动问题，用于为新用户推荐物品。

常用算法

Thompson sampling算法

Thompson sampling算法简单实用，它假设每个赌博机的背后有一个概率分布，产生收益的概率为p。我们需要不断地试验，去估计出一个置信度较高的“概率p的概率分布”就能近似解决这个问题了。

进一步，假设概率p的概率分布符合beta(wins, lose)分布，它有两个参数: wins和lose。每个臂都维护一个beta分布的参数。每次试验后，选中一个臂，摇一下，有收益则该臂的wins增加1，否则该臂的lose增加1。每次选择臂的方式是：用每个臂现有的beta分布产生一个随机数b，选择所有臂产生的随机数中最大的那个臂去摇。

其思想与贝叶斯分析类似，具体地，针对用户冷启动或者EE问题，可以设置win为点击数，lose为除点击外的次数。

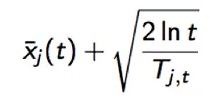
Epsilon-Greedy算法

朴素Bandit算法，首先选一个（0,1）之间较小的数作为epsilon；然后每次以概率epsilon从所有臂中随机选一个；以概率1-epsilon 选择截止到当前，平均收益最大的那个臂。

有点类似模拟退火的思想，具体地，收益可以是点击率或者其他业务指标。

UCB算法

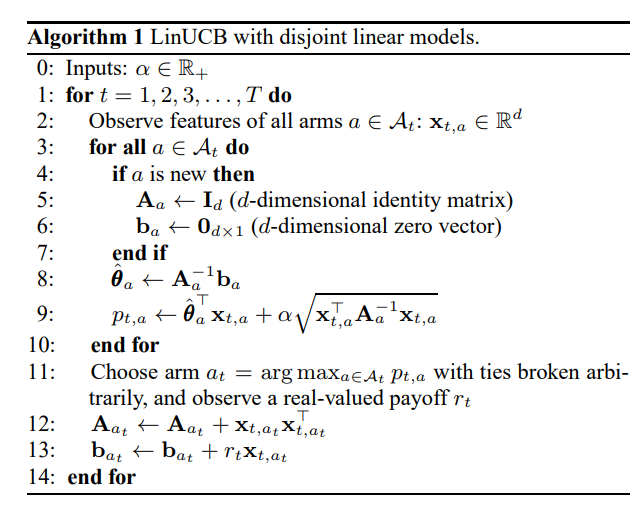
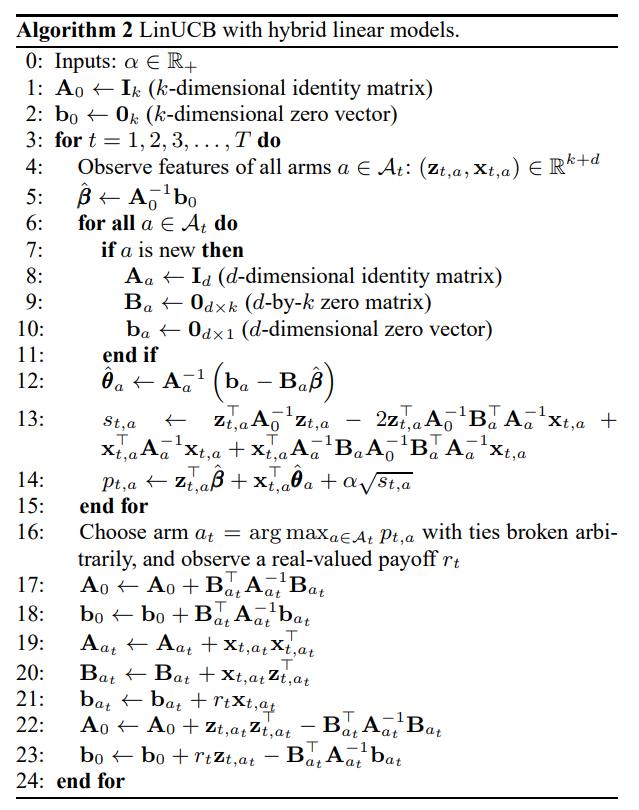
UCB算法全称是Upper Confidence Bound（置信区间上界），先对每一个臂都试一遍；按照如下公式计算每个臂的分数，然后选择分数最大的臂作为选择；观察选择结果，更新t和Tj,t。



其中加号前面是这个臂到目前的收益均值，相当于点击率，后面的叫做bonus，本质上是均值的标准差，t是目前的试验次数，即推荐总次数，Tj,t是这个臂被试次数，即该物品推荐次数。这个公式反映一个特点：均值越大，收益越大，但同时被选次数较少的臂的bonus也会比较大，达到EE的效果。

结合特征的UCB算法

该算法来自2012年的《A Contextual-Bandit Approach to Personalized News Article Recommendation》，里面提到了融合物品特征以及用户特征的UCB算法，有两种实现方式：互斥模型以及混合模型。

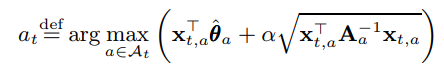
 

互斥的UCB模型的思想是利用多个过程下物品特征x与点击分数c来进行ridge回归，得到

，

对应地在算法1的流程图上，A相当于特征的外积D，然后再利用假设检验，得到



因此UCB公式为。

但在混合模型中，包含了新的特征z，是指用户与物品交互的特征，在实验中设置为两个特征的外积，故特征维度升高。