**原理介绍：结合上下文信息的Bandit算法—LinUCB算法**

备注：该强化学习的算法在推荐系统上已经被人实践过，故也进行了研究。

# 一、背景

强化学习关于Bandit相关的算法，如：Naive、Epsilon-Greedy（ε-Greedy）、Thompson Sampling（汤普森采样）、UCB（Upper Confidence Bound）等，其中LinUCB方法中考虑了臂的特征，故称为基于上下文的Bandit算法。

# 二、实现原理

算法参考使用的是岭回归的原理。

* 岭回归原理如下：

首先，传统的回归问题优化目标：



其中，i表示每个样本，yi表示每个样本对应目标真实值，xiTθ表示每个样本对应目标的估计值。

通过最小二乘法，可以解决求出θ的问题。最小二乘法的标准解如下：



当x维度很高时，或者样本数小于特征数时，使用岭回归能得到更好的结果,使左边的矩阵的秩为满秩（可求逆矩阵），岭回归的优化目标是：



岭回归的标准解如下：



其中I表示单位矩阵。

* LinUCB原理如下：

LinUCB 算法假设一个物品推送给用户之后，获得的收益与相关特征呈线性关系，这里的相关特征就是指上下文信息。LinUCB有两个版本：Disjoint和 Hybrid，Disjoint表示不同臂之间的不相关，也就是说参数不共享，Hybrid表示臂之间共享一些参数。这里只介绍 Disjoint模型。

为了方便说明，假设每个臂包含一个物品，我们在每一次选择时，用户与物品的的特征构成了上下文信息，表示为 x，维度为d，每个臂维护了一个d 维的表示特征系数的向量θ，使用c表示本次选择的收益，如果用户点击了就为1，否则为0。假定：

我们的目标就是求出每个臂对应的特征系数向量θ。如果知道多次选择时的上下文信息x及其对应的收益c，那么就可通过岭回归的方式来求出每个臂对应的最优θ。

以求解某个臂对应的θ为例（其他臂的方式一样），针对某个臂，假定我们收集了m次选择时的上下文信息x及其对应的收益c（也就是m条训练样本），分别使用D和C来表示，D 是一个m\*d的矩阵，表示这m次的上下文信息，C是一个m\*1的向量，表示这m次对应的收益。那么：

然后通过岭回归去求解出每个臂对应的最优的θ估计值。

求解出臂对应的θ的估计值之后，结合某次选择的上下文信息 x，就可以预测出收益 r：

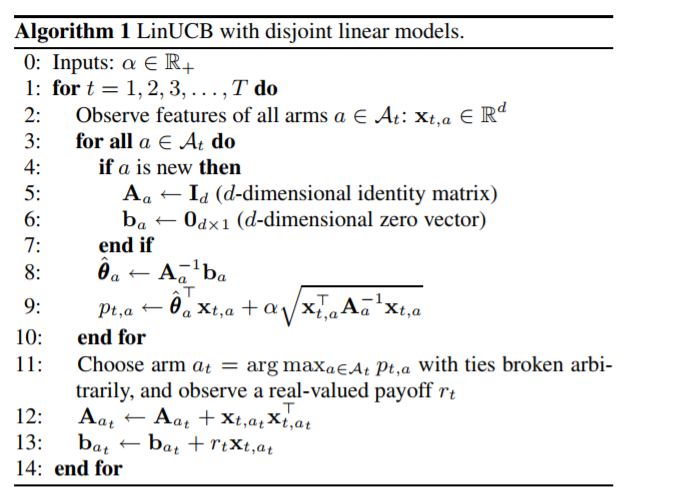
置信区间上边界为：

将预期收益与置信区间上边界相加得到 p：

这样，在每次进行选择时，计算每个臂对应的p，选择最大的p对应的臂。

# 三、算法流程

整个算法的流程如下：



详细步骤如下：

1.设定一个参数α，这个参数决定了我们 Explore的程度

2.开始试验迭代

3.获取选择每一个臂时的上下文信息（也就是特征向量）x(t,a)

4.开始计算每一个臂的预估收益及其置信区间

5.如果这个臂还从没有被试验过，那么：

6.用d\*d单位矩阵初始化作为Aa

7.用d\*1的零向量初始化作为ba

8.处理完没被试验过的臂

9.使用Aa和ba来计算该臂对应的预估θ

10.使用上一步计算出的预估θ和特征向量x(a,t)计算预估收益，同时加上置信区间宽度来计算p(t,a)

11.对每一个臂进行4-10步骤的过程

12.选择第10步计算出的最大的的p(t,a)的值对应的臂，并观察其真实收益rt

13.更新Aat

14.更新bat