Regret Analysis of Causal Bandit Problems

Yangyi Lu
Department of Statistics, University of Michigan
yylu@umich.edu

Amirhossein Meisami
Adobe Inc.
meisami@adobe.com

Ambuj Tewari
Department of Statistics, University of Michigan
tewaria@umich.edu

Zhenyu Yan
Adobe Inc.
wyan@adobe.com

【强化学习 96】Causal Bandit

张楚珩
清华大学 交叉信息院博士在读

提出一种 bandit problem + causal inference 相结合的问题，给出相应的算法。

原文传送门

Lattimore, Finnian, Tor Lattimore, and Mark D. Reid. "Causal bandits: Learning good interventions via causal inference." Advances in Neural Information Processing Systems. 2016.（提出 causal bandit problem，提出解决 best arm identification 问题的算法，分析了 simple regret）

Yangyi Lu, Amirhossein Meisami, Ambuj Tewari, and Zhenyu Yan. "Regret Analysis of Causal Bandit Problems." Arxiv preprint 1910.04938（提出 UCB/TS 方法来解决 causal bandit 里面的探索-利用问题）

特色

十月份的新 paper，在之前提出了一种比较新的问题 causal bandit problem 的基础上，设计了基于 upper confidence bound (UCB) 和 Thompson sampling (TS) 的方法。由于使用了 causal information，其在理论上和实验上都有更好的效果（更小的 regret）。

过程

1. 问题举例

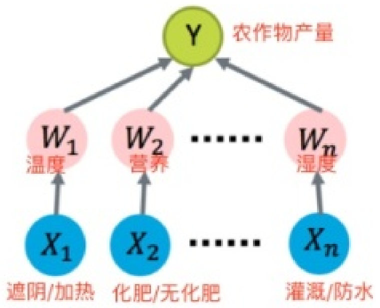
先来看几个现实生活里面可能遇到的问题，来作为 causal bandit problem 的例子。

农夫种地

这个例子来自于第一篇 paper。农民种地，目标是增加农作物的产量。跟农作物产量直接相关的有如下因素：1）温度；2）土壤中的某种营养物质含量；3）土壤湿度。但是农夫只能够通过一些操作来间接控制这些变量，比如：A）增加大棚遮阴或者用电热灯加热；B）使用或者不使用某种特殊的化肥；C）灌溉或者使用一种防水膜。

这个问题其实可以被看做是一个标准的 bandit problem，不同的 A-C 的组合看做不同的 arm，农作

物的产量看做是 reward。但是现实中，我们可以知道一些额外的信息，比如各个变量之间的因果关系（known causal structure）、虽然不能控制但是能够测量直接的决定因素（比如 1-3）。Causal bandit problem 就是新增了已知的 causal graph 和可以观察到的决定性因素。在这个问题里面，causal graph 可以被画成下面的形式。



在 causal bandit problem 中当然也可以和 bandit problem 类似解决不同的问题，比如 best arm identification 和 regret minimization。

Email Champaign

第二篇 paper 提供了另一个例子。Adobe 公司希望给用户发广告邮件，目标是希望增大用户把邮件点开的概率。蓝色节点表示可以控制的变量，有产品的种类（比如是 PS 还是 Acrobat）、邮件的目的（比如是福利还是营销）、发送的时间（比如是早上还是晚上）等。公司有一个邮件库，指定了蓝色节点对应的数值之后，就从邮件库里面选一个邮件给用户发，根据用户的不同反馈得到不同的奖励。但是最后决定用户会不会点击的是 z_1, z_2, z_3 三个因素，其相互关系如下图所示。（图中的数值表示每种随机变量有几种可能的取值）。注意到对于蓝色的节点来说，也可以不指定一个数值，这样就从其本身的一个分布中采样。

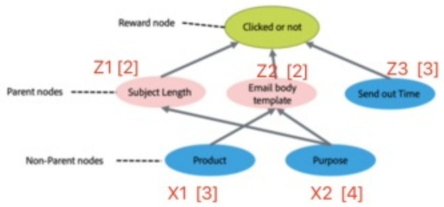


Figure 5: Causal Graph for Email Campaign: only blue nodes are under control.

2. 数学描述

Causal structure 给定，用一个 directed acyclic graph (DAG) $\mathcal{G}$ 表示。图中有 N 个随机变量，

$\mathcal{X}=\{X_1,\cdots,X_N\}$ ，每个节点可能的取值有 k 种（离散的）。某个节点 X_i 的父节点记为 Pa_{X_i} ，随机变量点 X_i 的分布由其所有父节点决定。存在一个在这些随机变量上的联合概率分布 P。行动（也叫 intervention）能够控制的节点为 $X=\{X_1,\cdots,X_m\}\subset\mathcal{X}$ ，一个行动记为 $a=\mathrm{do}(X=x)$ ，其中 $x=\{x_1,\cdots,x_m\}$ 为相应可以控制随机变量选定的值，也可以不对相应随机变量加以干涉，不妨记做 $x_i=0$ 。目标是最大化奖励随机变量 Y 的取值，记 $\mathrm{Pa}_Y=Z$ 为直接决定 Y 取值的随机变量集合。

对于某个行动，其对应的奖励期望可以被写作 $\mu_a:=\mathbb{E}[Y|a=\mathrm{do}(X=x)]$ 。Regret minimization 的目标就是通过每一回合选择合适的行动，使得 regret $\mathbb{E}[R_T]=T\mu^*-\sum_{t=1}^{T}\mathbb{E}[\mu_{a_t}]$ 最小，其中 $\mu^*=\max_a \mu_a$ 。

3. C-UCB

为了解决上述问题，文中提出 causal UCB (C-UCB) 方法。

Algorithm 1 C-UCB

Input: Horizon T, action set $\mathcal{A}$, δ, causal graph $\mathcal{G}$, number of parent variables n, number of values each parent variable can take on: k.

Initialization: Values assignment to parent variables: $\mathbf{Z}_j$, $\hat{\mu}_{\mathbf{Z}_j}(0)=0$, $T_{\mathbf{Z}_j}(0)=0$, for $j=1,\ldots,k^n$.

for $t=1,\ldots,T$ **do**

  for $j=1,\ldots,k^n$ **do**

    $\mathrm{UCB}_{\mathbf{Z}_j}(t-1)=\hat{\mu}_{\mathbf{Z}_j}(t-1)+\sqrt{\frac{2\log(1/\delta)}{1\vee T_{\mathbf{Z}_j}(t-1)}}$.

  end for

  $a_t=\mathrm{argmax}_{a\in\mathcal{A}}\sum_{j=1}^{k^n}\mathrm{UCB}_{\mathbf{Z}_j}(t-1)\boxed{P(\mathrm{Pa}_Y=\mathbf{Z}_j|a)}$

  Pull arm a_t and observe reward Y_t and its parent nodes' values $\mathbf{Z}_{(t)}$.

  Update $T_{\mathbf{Z}_j}(t)=\sum_{s=1}^{t}\mathbb{1}_{\{\mathbf{Z}_{(s)}=\mathbf{Z}_j\}}$ and $\hat{\mu}_{\mathbf{Z}_j}(t)=\frac{1}{T_{\mathbf{Z}_j}(t)}\sum_{s=1}^{t}Y_s\mathbb{1}_{\{\mathbf{Z}_{(s)}=\mathbf{Z}_j\}}$, for $j=1,\ldots,k^n$.

end for

其实比较直接。直接的 UCB 方法对于每一个 action 维护一个 UCB 数值，这里转而对每个不同的 Z 的取值维护 UCB 数值，由于 $|Z|=n$ 并且每个随机变量有 k 个可能的数值，因此需要维护 k^n 个 UCB 的值。由于已知了 causal graph，action 对随机变量 Z 产生的影响可以被计算出来，从而能够知道不同的 action 对应 UCB 的期望值。

该算法能够得到如下的 regret bound。

Theorem 1 (Regret Bound for C-UCB). *Let* $Y|_{Pa_Y=\mathbf{Z}_j}=\mathbb{E}[Y|Pa_Y=\mathbf{Z}_j]+\epsilon$, *for* $j=1,\ldots,k^n$, *where* ϵ *is a mean zero, 1-subgaussian distributed random error. If* $\delta=1/T^2$, *the regret of policy defined in Algorithm 1 is bounded by*

$$\mathbb{E}[R_T]=\tilde{O}\left(\sqrt{k^nT}\right). \quad (1)$$

4. C-TS

类似。（TS 不太熟，没仔细看了）

Algorithm 2 C-TS with Beta Prior (If $Y \in [0,1]$)

Input: Horizon T, action set $\mathcal{A}$, causal graph $\mathcal{G}$, all $P(Pa_Y|a)$, number of parent variables n, number of values each parent variable can take on: k.

Initialization: Value assignments to parent variables: $\mathbf{Z}_j$, $S^0_{\mathbf{Z}_j} = F^0_{\mathbf{Z}_j} = 1$, for $j = 1, \ldots, k^n$.

for $t \in \{1, \ldots, T\}$ **do**

  Sample $\hat{\theta}_j(t)$ from beta distn with parameters $(S^{t-1}_{\mathbf{Z}_j}, F^{t-1}_{\mathbf{Z}_j})$, for $j = 1, \ldots, k^n$.

  for action $a \in \mathcal{A}$ **do**

    $\hat{\mu}_a = \sum_{j=1}^{k^n} \hat{\theta}_j(t) P(\mathrm{Pa}_Y = \mathbf{Z}_j|a)$

  end for

  $a_t = \mathrm{argmax}_a\, \hat{\mu}_a$

  Pull arm a_t and observe reward $\tilde{Y}_t$ and its parent nodes values of $\mathbf{Z}_{(t)}$. Perform a Bernoulli trial with success probability Y_t and observe the output Y_t.

  if $Y_t = 1$ **then**

    $S^t_{\mathbf{Z}_{(t)}} = S^{t-1}_{\mathbf{Z}_{(t)}} + 1$

  else

    $F^t_{\mathbf{Z}_{(t)}} = F^{t-1}_{\mathbf{Z}_{(t)}} + 1$

  end if

end for

Theorem 2 (Bayesian Regret Bound for C-TS). *Let $Y|_{Pa_Y=\mathbf{Z}_j} = \mathbb{E}[Y|Pa_Y = \mathbf{Z}_j] + \epsilon$, for $j = 1, \ldots, k^n$, where ϵ is a mean zero, 1-subgaussian distributed random error. Then the Bayesian regret of policies in Algorithm 2 and Algorithm 3 are both be bounded by:*

$$BR_T = \tilde{O}\left(\sqrt{k^n T}\right). \tag{2}$$

5. 线性模型

假设奖励 Y 可以被写成 Z 的线性函数

$$Y|_{\mathrm{Pa}_Y=\mathbf{Z}} = f(\mathbf{Z})^T\theta + \epsilon, \tag{3}$$

where f denotes the feature function applied on the parent nodes of Y, θ denotes the linear coefficient and ϵ is a zero mean, 1-subgaussian distributed random error.

对于某个行动的奖励期望可以被写作

$$\mu_a = \sum_{j=1}^{k^n} \mathbb{E}\left[Y | \mathrm{Pa}_Y = \mathbf{Z}_j\right] P(\mathrm{Pa}_Y = \mathbf{Z}_j | a) \tag{4}$$

$$= \langle \sum_{j=1}^{k^n} f(\mathbf{Z}_j) P(\mathrm{Pa}_Y = \mathbf{Z}_j | a), \theta \rangle. \tag{5}$$

类似地，能够得到相应的线性算法，CL-UCB 和 CL-TS。

6. 和普通 bandit 问题解法比较

文章想说明在特定的情形下，使用 causal information 对应的算法，相比于不使用这些信息得到的算法，性能上有提升。

考虑如下问题：一共有 N 个随机变量，每个随机变量有两种可能的取值，最后奖励只和第一个随机变量的取值有关，即 $Y = \Delta \cdot X_1 + \epsilon, \quad \epsilon \sim N(0,1)$ 。

根据前面的分析 C-UCB 算法 regret 为 $O(\sqrt{k^n T}) = O(\sqrt{2^1 T})$ ；而在该问题中，如果不使用 causal information，使用 UCB 算法 regret 为 $O(\sqrt{3^N T})$ ，会随着整体随机变量数目的增长而增长。

直观上来讲，由于知道了不同 action 如何影响 Z，即可以间接控制 Z；因此，对 Z 做 UCB 肯定会更直接；当 Z 的维度小的时候，用 causal information 当然效果会更明显。

感觉实现需要知道 causal graph 就已经比较 bug 了，更 bug 的是，还能准确定量地得到不同 action 产生的不同随机变量 Z 的分布。

发布于 2019-10-17