bandit algorithm による LP の最適化

開発チーム　上原隆寛

1　はじめに

web 広告において、より効果的な広告を表示するための代表的な方法として、AB テストがある。優れた効果を持つものを A, 劣っているものを B とした時に、事前情報がない状態で AB テストを行うと、統計的に十分な試行回数まで A も B も平等に扱うことになる。統計的に有意差が出るまでの期間、B も A と同じだけ選択し続けるのは無駄が大きく、大きな機会損失であると言える。

bandit algorithm は k 回目の選択を、k-1 回目までの情報を元に判断するアルゴリズムである。k 回目までで A が優れている場合一定の確率で A を選択し、残りの確率で B, C, D の効果を探索しに行くことを選択する。これは言い換えると、「できるだけその時点での最適解を選択しつつ、他の選択肢の効果を知るための探索的選択も行う」ということである。

任意の k 回目の選択を最適化することはつまり、累積報酬を最大化することであり、この点において AB テストに比べて優れていることがわかる。今回はこの bandit algorithm を LP に適用し、効果を最大化したいと考えている。

2　ベイズ推定の説明

ベイズ推定は大まかにいうと、起きた事象に合わせて仮定していた確率分布を更新していくという手法です。例えば、表と裏がどれくらいの確率で出るかわからないコインを投げるという事象を考えます。情報がないので、一般的なコインに従って表と裏の出る確率をそれぞれ 0.5, 0.5 とします。コインを投げて表が出たら、表が出やすいコインである確率が高いため、0.5 → 0.6 と更新します。そのあと裏が出たら 0.6 →　0.5 に更新し直します。(この数値の更新はかなり単純化しています。) このような更新を繰り返すことでその都度確率分布を更新していくのがベイズ推定です。ここで、更新する前の確率を事前確率、更新した後の確率を事後確率と呼びます。ベイズ推定を一言でまとめると、観測された事象を元に事前確率を更新し事後確率を導くということであると言えます。ベイズ推定の優れている点は二つあります。一つ目は 3 回連続で表が出たとしても、コインは大体の場合 50 与えているため、早計に表の確率を 100 二つ目は試行回数の少ない主観的な事象の結果を、本来十分な試行回数が必要である確率に落とし込めるところです。今回では各広告表示の結果をその都度ベイズ的に反映していく事でどの広告がより優れているかということを判定していきます。

3　尤度

簡単のために、web 広告では目的のページに遷移したら報酬 1 が得られ、そうでない場合報酬が 0 であると考えます。また、任意の選択肢 i の報酬を x_i とし、報酬が得られる確率を μ_i と更新します。ここで、$x_i = 1$ である確率は μ_i、$x_i = 0$ である確率は $1 - \mu_i$ と書くことができます。これらはそれぞれ

$$p(x_i = 1|\mu_i) = \mu_i$$

$$p(x_i = 0|\mu_i) = 1 - \mu_i$$

と書くことができ、これをまとめると

$$p(x_i|\mu_i) = \mu_i^{x_i}(1 - \mu_i)^{1-x_i}$$

と書くことができます。この $p(x_i|\mu_i)$ をベルヌーイ分布と呼びます。真の期待値が μ_i が求まっている場合、この μ_i がもっとも高い選択肢を選び続けることで web 広告の効果を最大化することができます。しかし、実際には真の μ_i はもとまらないので、いかにして μ_i が高いであろう選択肢を選択するかということが重要となってきます。

4　事前確率と事後確率

前項から真の μ_i の値がもとまらない中でうまく μ_i の値を予測していくことが必要になります。ここで、観測結果 x_i が与えられた時の μ_i の分布 $p(\mu_i|x_i)$ を考えることで確率分布の形で μ_i を与えたいと思います。それまでの更新で $p(\mu_i)$ は求まっているので、ベイズの定理を用いることで、

$$p(\mu_i|x_i) = \frac{p(\mu_i)p(x_i|\mu_i)}{\int p(\mu_i)p(x_i|\mu_i)d\mu_i}$$

と求めることができます。新しい x_i が与えられる度にこのように事後確率を更新し、その事後確率を次の更新での事前確率として用いることで、その都度 $p(\mu_i)$ を求めていきます。

5　Thompson Sampling の説明

今回はバンディットモデルの中で最も効率的であるアルゴリズムの一つである Thompson Sampling を使います。Thompson Sampling は各選択において、「アームの報酬の期待値を元にアームを選択します。アームの報酬の期待値は確率分布の形で与えられるため、直接比較することが難しいです。Thompson Sampling では各アームの期待値の事後確率 ($p(\mu_i)$) から乱数を生成し、その

乱数が一番大きいものを選択する手法です。bandit アルゴリズムにおける活用と探索のバランスを事後確率分布からの乱数生成によって行なっています。(これは二つの正規分布から乱数を生成するとして、グラフに重なりが存在すれば、山の部分がいくら離れていようが小さい方が選択される可能性が存在することを考えれば直感的に理解ができます。)

6　小まとめ

ここで、これまでのまとめを行います。まず、広告効果を測る上で知りたいのは各パターンの期待値 μ_i です。仮に μ_i を定めた場合に x_i はベルヌーイ分布の形で与えられます。ただ、直接 μ_i を知ることはできないので、それまでの事象から $p(\mu_i)$ つまり期待値の確率分布を求めたいと思います。新しく x_i が結果として得られた時、それまでの $p(\mu_i)$ を用いて新しく $p(\mu_i|x_i)$ を求めることができます。次の更新では $p(\mu_i|x_i)$ を $p(\mu_i)$ として用いることで更新を続けていきます。ここまでは、bandit と関係なく、ベイズ推定の話です。banditAlgorithm では複数の選択肢について上記のように確率の更新を行います。求まった $p(\mu_i), i = 1..n$ から乱数を生成して、一番乱数が大きかったものを選択するのが ThompsonSampling です。

7　確率の更新

実際に ThompsonSampling ではどのように確率分布の更新を行なっているかを説明していきます。ベイズの定理から $p(\mu_i|x_i)$ を求める時、$p(\mu_i), p(x_i|\mu_i)$ を適当に選ぶと $p(\mu_i|x_i)$ の確率分布は未知のものになってしまい、一般的には求めることができません。よって $p(\mu_i|x_i)$ が既知の分布になるように $p(\mu_i), p(x_i|\mu_i)$ をうまく選ぶ必要があります。また、$p(\mu_i|x_i)$ の更新は何度も行うため、できれば事前分布 $(p(\mu_i))$ と事後分布 $(p(\mu_i|x_i))$ の分布の種類を揃え任意の回数の更新で行う作業を同じにしたいです。尤度 $p(x_i|\mu_i)$ に対応して、上の条件を満たす確率分布を共役事前分布といいます。今回の $p(x_i|\mu_i)$ はベルヌーイ分布であるので対応する共役事前分布はベータ分布になります。事前分布と事後分布が同じ種類の確率分布であるため、各回における事前確率の更新は以下のようにまとめることができます。

$$a = a + x$$

$$b = b + 1 - x$$

8　まとめ

今回行なったことは主に二つあります。一つ目は、ベイズ推定によって各パターンの広告の期待値予測を行う。二つ目は、期待値予測に基づいて探索と活用のバランスがよくなるように選択肢を選ぶということです。この方法は尤度とそれに対応した共役事前分布を選択することによって、複数段階の報酬に対応したり、広告途中追加に対応していたりなど非常に汎用性の高い方法です。この論文では、web 広告を例に、ThompsonSampling を用いた banditAlgorithm の紹介を行いました。