Personalized Marketing

IMA Math-to-Industry Boot-Camp

Mentor: Richard Sharp

Nicole Bridgland, Yiying Cheng, Tianyi Mao
Hieu Nguyen, Kelly O'Connell, Anna Srapionyan

July 27, 2017

1 Introduction

Our goal for this project was to design an algorithm to make offer decisions for marketing. A standard practice is to send out offers somewhat randomly and see what has the desired effect. That takes a long time, and wastes money sending ineffective ads during the testing phase. Since offers change frequently, information gained from such a slow method cannot be put to use for very long, if at all. Our algorithm starts learning what different people are likely to respond to within days of sending the offers out, and continues to improve as time goes on. We accomplished this by segmenting the (simulated) population into groups based on their demographic and historical spending data, determining a short-term metric for success (KPIs), and then using a probabilistic experiment called a contextual bandit to learn which types of people prefer which types of offers. Running our algorithm on our small simulated population for 15 days gave a revenue increase of ten thousand dollars compared with distributing offers randomly. Scaling to ten million people and a year, this corresponds to a revenue gain of 243 million dollars.

Disclaimer: This project is a hypothetical approach to personalization and does not make use of proprietary data or methods owned by Starbucks. For more information about the methods used in this project or personalization at Starbucks please see Contextual Bandit Algorithms with Supervised Learning Guarantees or A.I. Informs Personalization for Starbucks.

2 Key Performance Indicators

A Key Performance Indicator (KPI) is a measurable value that demonstrates how effectively a company is achieving key business objectives. These metrics are used to determine a company's progress in achieving its strategic and operational goals, and also to compare a company's finances and performance against other businesses within its industry. High-level KPIs may focus on the overall performance of the enterprise, while low-level KPIs may focus on processes in departments such as sales, marketing or a call center.

Some examples of KPIs in marketing and sales are:

- New customer acquisition.
- Demographic analysis of individuals (potential customers) applying to become customers, and the levels of approval, rejections, and pending numbers.
- Status of existing customers.
- Customer attrition.
- Turnover (i.e., revenue) generated by segments of the customer population.
- Outstanding balances held by segments of customers and terms of payment.
- Collection of bad debts within customer relationships.
- Profitability of customers by demographic segments and segmentation of customers by profitability.

For our project we are using the following five KPIs:

1. KPI_1 shows how much more money a person in a target group spends than a person in a control group (measured for the first three days of the validity period of each offer):

$$KPI_1 = \frac{\$Mean(Spent_{Target})}{\$Mean(Spent_{Control})}.$$

2. KPI_3 shows what fraction of offers are being completed after being viewed by the target group (measured for the first three days of the validity period of each offer):

$$KPI_3 = \frac{\#Completed}{\#Viewed}.$$

3. KPI_4 shows the fraction of number of transactions for people who received an offer over number of transactions who did not (measured for the first three days of the validity period of each offer):

$$KPI_4 = \frac{\#Transactions_{Target}}{\#Transactions_{Control}}.$$

4. KPI_5 shows the amount of money per transaction after viewing the offer over the amount of money per transaction before viewing the offer. This is calculated only for a subset of the target group - those who had transactions before and after viewing the offer:

$$KPI_5 = \frac{\text{\$ per transaction before viewing the offer}}{\text{\$ per transaction after viewing the offer}}.$$

5. KPI_6 shows how much more transactions per time unit people in target group make after and before viewing the offer. This is only calculated for a subset of the target group - those who had transactions before and after viewing the offer:

$$KPI_6 = \frac{\#\text{ transactions per time after viewing the offer}}{\#\text{ transactions per time before viewing the offer}}.$$

Ideally, we would like our short-time KPIs to be correlated with the long-times.

Next, since we want to pass a single KPI to the decision strategies' (bandit's) algorithm, we take the weighted average of the above five KPIs. The weights of the KPIs depend on the importance of our business goals and objectives:

$$KPI = 0.32 \times KPI_1 + 0.16 \times KPI_3 + 0.32 \times KPI_4 + 0.1 \times KPI_5 + 0.1 \times KPI_6.$$

3 Segmentation

Customers respond differently to different offers, hence to decide which offer to send to a customer, we need to know their behavior. From demographic and past preference data given by the customers, segmentation groups people with similar behavior and responsiveness. For instance, students should be more delighted to get a buy one get one promotion than a millionaire. Quality of the customer segmentation will influence on decision strategy of giving out offers.

One clustering technique is k-means, which is used in this study because of its robustness. Some features taken into account are customer age, length of membership, spending before and after offer, etc. The default number of clusters is 4, which we have observed to work well. For future work, consensus clustering can be used to estimate the number of clusters.

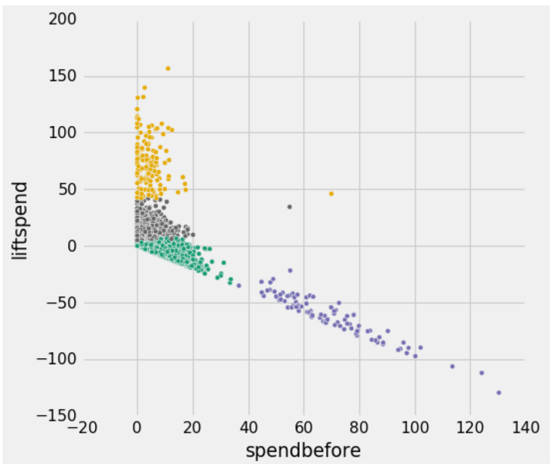

Figure 1: Customer segmentation.

4 Decision Strategy

Consider offers being sent to customers in each group. We want to send the offer with the largest payoff for each group. Let us represent each expected payoff by μ_i- i.e. μ_i is the true payoff that the specific group use the $i-th$ offer. We assume that μ_i do not change over time. It is important to note that we do not know what μ_i is, if we know, then we could just choose the offer i for which μ_i was largest. So we modeled our problem as Multi-arm Bandit problem and used Bayesian Bandit Algorithm to solve this problem.

4.1 Muti-armed Bandit

A multi-armed bandit is an experiment with the goal of achieving the largest possible reward from a payoff distribution with unknown parameters that are to be learned sequentially. It models an agent that simultaneously attempts to acquire new knowledge (called "exploration") and optimize decisions based on existing knowledge (called "exploitation"). An algorithm of the MAB problem must decide which offer to send at each time step t, based on the payoffs of the previous $t-1$ plays. Suppose we have K offers to send to the customers in each group. Let μ_i denote the (unknown) expected payoff for offer i. Our goal is to maximize the expected total payoff in time T, i.e., $\mathbb{E}[\sum_{t=1}^{T} \mu_{i(t)}]$, where $i(t)$ is the offer sent in step t, and the expectation is over the choices of $i(t)$ made by the algorithm. Or, equivalently, to minimize the expected total regret: the amount we lose because of not playing optimal offer in each step. The expected total regret in time T is given by

$$\mathbb{E}[\mathcal{R}(T)] = \mathbb{E}[\sum_{t=1}^{T}(\mu^* - \mu_{i(t)})]$$

where $\mu^* = \max\{\mu_j : 1 \leq j \leq K\}$

4.2 Bayesian Bandit Algorithm

In the model described above, we have 3 possible offers and 7 groups of people. We assume that the distribution of payoff of the offer i for different groups are different and the same for the people in the same group. So we can consider a specific group and we just follow the same idea for other groups.

4.2.1 Algorithm

The Algorithm: Suppose that for each offer, payoff is generated from some parametric distribution ν_i. Then the overall structure of the algorithm is as follows:

- For every offer, start with a prior belief on parameters of the distribution
- On making observations from an offer, update to posterior belief.
- At time t, send every offer with its posterior probability of being the best offer

We define the payoff r_t for sending offer to be the lift revenue(determined by KPI). We randomly selected a fixed fraction of people as our control group(no offer), and we also have several target groups T_i who received offer i. Suppose the average revenue at time step t is $r_C(t)$ for control group, $r_{T_i}(t)$ for target group T_i. Then our payoff of sending offer i is $r_t = r_{T_i}(t) - r_C(t)$.

The prior is something we believe to be true before we have any evidence - i.e., before we have sent the offer to any customers in group one. Here we choose the Gaussian distribution $\nu_i = \mathcal{N}(\mu_i, 1)$; μ_i is unknown. We only choose the Gaussian distribution specially for mathematically simplicity, we can compute a closed form of the posterior. If the prior is Gaussian $\mathcal{N}(0,1)$, the posterior after time t will be the Gaussian $\mathcal{N}(\hat{\mu}_{i,t}, \frac{1}{n_{i,t}+1})$. Here $n_{i,t}$ is the number of plays of offer i in the time $[1, \cdots, t]$ and $\hat{\mu}_{i,t} = \frac{1}{(n_{i,t}+1)} \sum\limits_{1 \leq \gamma \leq t, a_\gamma = i} r_\gamma$. We set $\hat{\mu}_{i,0} = 0$ for all i.

Suppose we define the payoff $r_{T_i}(t)$, $r_C(t)$ to be the revenue for target group and control group, then the distribution for target group is

$$\nu_{i,T_i} = \mathcal{N}(\mu_{T_i,t}, \frac{1}{n_{T_i,t}+1})$$

where $n_{T_i,t}$ is the number of people in the target group who received offer i, and the distribution for control group is

$$\nu_{i,C} = \mathcal{N}(\mu_{C_t,t}, \frac{1}{n_{C,t}+1})$$

where $n_{C,t}$ is the number of people in the control group.
Since they are independent, we can know that the posterior distribution for r_t is

$$\mathcal{N}(\hat{\mu}_{i,t}, \frac{1}{n_{T_i,t}+1} + \frac{1}{n_{C,t}+1})$$

Algorithm 1 Bayesian Bandit using Gaussian priors

for each $t = 1, 2, \cdots, T$ **do**
    for each offer $i = 1, \cdots, N$, sample μ_i independently from the distribution $\mathcal{N}(\hat{\mu}_{i,t}, \frac{1}{n_{T_i,t}+1} + \frac{1}{n_{C,t}+1})$
    Send Offer $a_t = \arg\max_i \mu_i$
    Observe payoff r_t

Noticed that, even though we assume that the people in the same group have the same behavior, we pick the sample from the distribution individually. We use an example to illustrate the reason. Suppose for offer one, our sample of μ_1 has worked out to be 2.2, while our sample of μ_2 is only 2. Since $\mu_1 > \mu_2$, we will send offer one. However, there was no guarantee that things worked out this way. It was possible that we randomly choose μ_1 be 1.8, and μ_2 be 2. Then actually we will send offer two. Since there is overlap between the distributions, we decided to choose sample individually to reduce the error and update the distribution of payoff for each offer simultaneously.

4.2.2 Result

We simulated this process on 10,000 customers for 15 days, and used random strategy as our baseline.

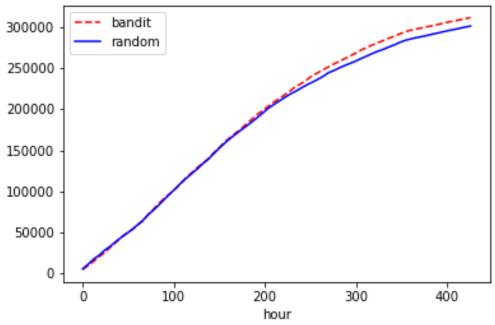

Figure 2: Revenue of random strategy and bandit algorithm.

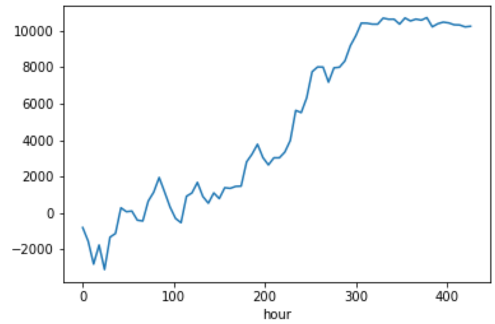

Figure 3: Lift revenue for bandit algorithm

From the above graphs, we can see the algorithm gave a revenue increase of ten thousand dollars over distributing offers randomly. So, we got a better strategy to send offers.