Group Project: Dynamic Pricing under Competition

Xinzhuo Jiang, Xiao Lei, Ting Liu, Wenyu Ma, Shixiang Zhou

April 29, 2017

1 Binary Multinomial Logit Demand

In the first case, we assume our demand follows a Bernoulli random variable with parameter

$$\frac{e^{\alpha-\beta\cdot p_{1t}}}{1+\sum_{i=1}^{n} e^{\alpha-\beta\cdot p_{it}}},$$

where p_{1t} is our price and $p_{it}, i \neq 1$ is others price at time t.

The starting price is initialized as 1. From 2 to 100 period, we follow the price of other competitors by choosing the lowest price except ourselves in the last period. After collecting 100 prices from all teams and 100 demand of our own, the MLE method is applied to estimate α and β. The initial starting point is 0 and 1. Later on, we use the parameters from last period as the starting points in current period. The log-likelihood function for time τ is

$$LLH = \sum_{t=1}^{\tau} \log\left((\frac{e^{\alpha-\beta\cdot p_{1t}}}{1+\sum_{i=1}^{n} e^{\alpha-\beta\cdot p_{it}}})^{y_t} (1 - \frac{e^{\alpha-\beta\cdot p_{1t}}}{1+\sum_{i=1}^{n} e^{\alpha-\beta\cdot p_{it}}})^{1-y_t} \right),$$

where y_t is our own demand.

After estimating the demand model, we need to predict competitors' prices. We assume the *sorted prices* of competitors follows the multivariate normal distribution, i.e.

$$p = [p_{(1)}, \ldots, p_{(n-1)}] \sim Normal(\mu, \Sigma),$$

where $p_{(1)}$ is the smallest price and $p_{(n-1)}$ is the largest price for other competitors. To estimate μ and Σ, we use the last 100 prices. We sort every column of the price matrix as increasing order, and then calculate the mean and covariance matrix of each row. The sorting procedure is to exclude the effect of symmetry. For example, if a team use (0,100,0,100,...) and another team use (100,0,100,0,...), before sorting we will use 50 as expected price from both competitors, but after sorting we will recognize that the competitors will use 0 and 100 as their prices, and we don't care who bid 0 and who bid 100.

Finally, we use sample average approximation (SAA) to choose the optimal prices. We sample 1000 price vectors by the multivariate normal distribution. The average revenue for a price p_1 is simply given by

$$E(R) = \frac{1}{1000}\sum_{j=1}^{1000} p_1 \cdot \frac{e^{\alpha-\beta p_1}}{1+e^{\alpha-\beta p_1}+\sum_{i=1}^{n-1} e^{\alpha-\beta p_{(i)}^j}},$$

where $p_{(i)}^j$ is the (i)-th smallest price in sample j. To optimize over p_1, we simply try $p_1 = 0.1, 0.2, \ldots, 99.9$ one by one and choose the one with largest mean revenue.

2 General Demand

When there is no specific demand functions, we assume the total demand equals to D_k with probability θ_k. For each D_k, our demand follows a binomial distribution with parameter D_k and possibility

$$\frac{e^{\alpha_k-\beta_k\cdot p_{1t}}}{1+\sum_{i=1}^{n} e^{\alpha_k-\beta_k\cdot p_{it}}},$$

i.e., there are D_k customers and each one make their choice following a MNL model.

The starting price will be 0. The reason of doing so is that we want to receive as many demand as we can to estimate the total demand.

For $t = 2$ to 100, we follow the lowest price of other competitors in the last period. Meanwhile, we construct the set possible D_k as $D = \{2^i : \max y_t \le 2^i \le (n+1)\max y_t\}$, where y_t is our own demand. The motivation is: D_k is at least the most demand we receive, and considering a no-purchase situation, the expectation of no-purchase will be less than the expected demand when price is 0. So using $(n+1)\max y_t$ as the upper bound is enough for estimating. We construct the set at time 2, and update it in the first 100 periods.

After the first 100 period, we use EM algorithm to estimate $\theta_k, \alpha_k, \beta_k$. In expectation step we estimate θ_k and in maximization step we estimate α_k

and β_k. Each time we use the parameter from last time as starting point. We then still use the last 100 sorted prices to estimate multivariate normal distribution, and sample 1000 price for the other competitors. However, the expected revenue for p_1 becomes

$$E(R) = \sum_{k=1}^{K} \theta_k D_k \frac{e^{\alpha_k - \beta_k \cdot p_{1t}}}{1 + \sum_{i=1}^{n} e^{\alpha_k - \beta_k \cdot p_{it}}}.$$

To optimize over p_1, we simply try $p_1 = 0.1, 0.2, \ldots, 99.9$ one by one and choose the one with largest mean revenue.