Solutions to Discount Optimisation

Yu Pengqian

1 Background and Objectives

E-commerce platform often provide vouchers to buyers who have been "inactive" for a long time. "Inactive" buyers are those who have not purchased anything from the platform for some specified time period. The purpose of sending vouchers is to motivate these buyers to start to buy things from the platform again and hopefully they will continue the purchases subsequently.

The goal in this mini project is to design an effective discounts and rebates management process, and to reactive as many as possible potential buyers by giving them right amount of discount under budget constraint.

With the background and objectives in mind, we break this problem into two main tasks:

Task 1. Classifying the "inactive" buyers into several groups according to their identifications, historical behaviors and vouchers received. In particular, we wish to know

Subtask 1. whether a buyer will use the voucher;

Subtask 2. whether a buyer will make a repurchase within a certain period.

Task 2. Identifying the groups of potential buyers and deciding the voucher strategy such that the "reactivation" (adoption rate) is maximized under the fixed budget.

We organize this report as follows: In Section 2, we introduce methods used for Task 1. In particular, we discuss methodologies used for two subtasks in Section 2.1 and Section 2.2, respectively. We investigate the approaches for Task 2 in Section 3. We provide implementation details for both tasks in Section 4. Finally, the results and discussions are offered in Section 5.

2 Methodology for Classification

In this section, we describe the techniques used for the classification in discount optimisation problem. There are two subtasks for the classification task, namely,

- to predict the voucher is used or not;
- to predict the time period of a repurchase given the usage of voucher.

We discuss these subtasks separately in below.

2.1 Predicting the Voucher Usage

We treat this subtask as a binary classification problem and use logistic regression technique to solve the problem.

Logistic regression is an important machine learning algorithm, and the goal is to model the probability of a random variable Y being 1 or 0 given experimental data. In our setting, the

random variable Y is interpreted as the voucher usage, i.e., $y = 1$ if the voucher is used and $y = 0$ if the voucher is unused. In addition, we consider a generalized linear model function parametrized by θ

$$h_\theta(\mathbf{x}) \triangleq \frac{1}{1 + e^{-\theta^\top \mathbf{x}}}$$

where $\mathbf{x} \in \mathbb{R}^{27}$ is the feature vector

$$\mathbf{x} \triangleq [\quad 1, \quad \underbrace{\mathbf{p}, \quad \mathbf{b}, \quad \mathbf{w},}_{\text{from user's transaction history}} \quad \underbrace{\mathbf{c}, \quad \mathbf{t}, \quad \mathbf{a}, \quad \mathbf{g}}_{\text{from user's id}} \quad]. \tag{1}$$

Here $\mathbf{p} \in \mathbb{R}^6$ is the promotion type and

$$\mathbf{p} = [\mathbf{e}_3(i) \quad \mathbf{e}_3(j)] \quad \text{if the discount type is i and max\_value is of type j,}$$

where $\mathbf{e}_3(i)$ denotes the unit vector in $\mathbb{R}^3$ with i-th element being 1. The discount and max_value types are designed according to *voucher_mechanics.csv*, and they are shown in Table 1 and Table 2, respectively.

Discount type	Amount
1	20
2	30
3	50

Table 1: Types of discounts.

Max_value type	Amount
1	1000000
2	1500000
3	2000000

Table 2: Types of max_value.

The feature vector $\mathbf{b} \in \mathbb{R}^2$ comes from the *voucher_distribution_active_date.csv*, and it records the total number of active sessions for each user 31 days before the voucher receiving date. Specifically, it represents the user's activeness before receiving a certain voucher

$$\mathbf{b} = \begin{cases} \mathbf{e}_2(1) & \text{if the user's active sessions in last 31 days are below } 23^1, \\ \mathbf{e}_2(2) & \text{otherwise.} \end{cases}$$

According to *transactions_MY.csv*, the vector $\mathbf{w} \in \mathbb{R}^3$ represents the total amount of money spent on the website, and it has the form

$$\mathbf{w} = \begin{cases} e_3(1) & \text{the total amount of money spent is between 0 and } 6 \times 10^7, \\ e_3(2) & \text{the total amount of money spent is above } 6 \times 10^7, \\ e_3(3) & \text{no record}^2. \end{cases}$$

[1]Here the threshold 23 is the mean of users' active sessions.
[2]Here the threshold 6×10^7 is the mean of users' total amount of spent money.

Moreover, $\mathbf{c} \in \mathbb{R}^2$ in the feature vector represents the completeness of users' profile in *user_profiles_MY.csv*. That is,

$$\mathbf{c} = \begin{cases} \mathbf{e}_2(1) & \text{if the user's profile is complete,} \\ \mathbf{e}_2(2) & \text{otherwise.} \end{cases}$$

$\mathbf{t} \in \mathbb{R}^4$ represents the (rounded) year since registration. According to *user_profiles_MY.csv*, $\mathbf{t}$ has the form

$$\mathbf{t} = \begin{cases} \mathbf{e}_4(1) & \text{if the user has registrated for 0 year,} \\ \mathbf{e}_4(2) & \text{if the user has registrated for 1 year,} \\ \mathbf{e}_4(3) & \text{if the user has registrated for 2 years,} \\ \mathbf{e}_4(4) & \text{if the user has registrated more than 3 years.} \end{cases}$$

$\mathbf{a} \in \mathbb{R}^6$ represents the range of ages for each user from *user_profiles_MY.csv*. In particular,

$$\mathbf{a} = \begin{cases} \mathbf{e}_6(1) & \text{if the user's age is between 0 and 20,} \\ \mathbf{e}_6(2) & \text{if the user's age is between 20 and 30,} \\ \mathbf{e}_6(3) & \text{if the user's age is between 30 and 40,} \\ \mathbf{e}_6(4) & \text{if the user's age is between 40 and 50,} \\ \mathbf{e}_6(5) & \text{if the user's age is between 50 and 60,} \\ \mathbf{e}_6(6) & \text{otherwise.} \end{cases}$$

$\mathbf{g} \in \mathbb{R}^3$ represents the gender of each user, and it has the form

$$\mathbf{g} = \begin{cases} \mathbf{e}_3(1) & \text{if the user is a male or predicted male,} \\ \mathbf{e}_3(2) & \text{if the user is a female or a predicted female,} \\ \mathbf{e}_3(3) & \text{if the user's gender is unknown.} \end{cases}$$

We model the probability that y is 1 and 0 with the function

$$\mathbb{P}(y|\mathbf{x};\theta) = h_\theta(\mathbf{x})^y(1 - h_\theta(\mathbf{x}))^{1-y}.$$

Assuming that all samples are independent, we further take our likelihood function of the form

$$\begin{aligned} L(\theta|\mathbf{x}) \triangleq & \mathbb{P}(Y|X;\theta) \\ = & \prod_i \mathbb{P}(y_i|\mathbf{x}_i;\theta) \\ = & \prod_i h_\theta(\mathbf{x}_i)^{y_i}(1 - h_\theta(\mathbf{x}_i))^{1-y_i}. \end{aligned}$$

We are maximizing the log likelihood $N^{-1}\log L(\theta|\mathbf{x})$ which yields the optimization problem

$$\min_\theta -\frac{1}{N}\left[\sum_{i=1}^{N} y_i \log h_\theta(\mathbf{x}_i) + (1 - y_i)\log(1 - h_\theta(\mathbf{x}_i))\right].$$

Here N is the sample size.

Group	Description
1	The buyer will repurchase in 15 days.
2	The buyer will repurchase in 30 days.
3	The buyer will repurchase in 60 days.
4	The buyer will repurchase in 90 days.
5	The buyer will not repurchase.

Table 3: Groups of buyers.

2.2 Predicting the Repurchase

Recall that the goal is to maximize the "reactivation" rate. We divide the "inactive" buyers into five groups according to the time duration they come back for a repurchase. In particular, we have the groups of buyers described below in Table 3.

For each group $i \in I \triangleq \{1, 2, 3, 4, 5\}$, we denote $\mathbf{y} = \mathbf{e}_5(i)$ as the label vector for each buyer. The repurchase is related to the received voucher, and we further define the feature vector $\mathbf{z}$ for each buyer as

$$\mathbf{z} \triangleq [\mathbf{p}, \quad \mathbf{b}, \quad \mathbf{w}, \quad \mathbf{c}, \quad \mathbf{t}, \quad \mathbf{a}, \quad \mathbf{g}, \quad \mathbf{u}] \tag{2}$$

Here vectors $\mathbf{p}, \mathbf{b}, \mathbf{w}, \mathbf{c}, \mathbf{t}, \mathbf{a}, \mathbf{g}$ are defined similarly as the ones in (1), and $\mathbf{u}$ is the indicator of the voucher's usage in *training.csv*, i.e.,

$$\mathbf{u} = \begin{cases} \mathbf{e}_2(1) & \text{if the user used the voucher,} \\ \mathbf{e}_2(2) & \text{otherwise.} \end{cases}$$

We will use a deep neural network (DNN) for this classification problem.

3 Maximization of Reactivation Rate

In the previous section, we have identified/predicted the voucher usage and the user's expected duration of return. Let M be the total number of potential"inactive" users. Our strategy is listed as follows

- We will not send the voucher to users who are unlikely to use it. We define the total number of such users is M_1;
- For users who are expected to use the voucher, we will decide the voucher type (discount value and minimum amount of money spent) to send based on the predicted user's duration of return. For a user who is likely to return in t days, we will send him/her a voucher with t-days discount duration.

Let d, w and v denote the time difference of last purchase up to now, discount value and minimum spend requirement, respectively. According to the preliminary study, the voucher adoption rate γ for each user is given by

$$0.098 + 0.003w - 0.002v - 0.00042d + 0.005t, \tag{3}$$

and the cost associated with this voucher is w. Note that the decision variables $w \in \{20, 30, 50\}$ and $v \in \{1000000, 1500000, 2000000\}$ are taking discrete values.

We can then formulate the objective in this report as the following mixed integer programming problem

$$
\begin{aligned}
&\max_{w_i,v_i} \frac{1}{M_2} \sum_{i=1}^{M_2} [0.098 + 0.03(2w_{i1} + 3w_{i2} + 5w_{i3}) - 2(1000v_{i1} + 1500v_{i2} + 2000v_{i3}) - 0.00042d_i + 0.005t_i] \\
&\text{s.t.} \sum_{j=1}^{3} w_{ij} = 1, \quad \forall i \\
&\quad \sum_{j=1}^{3} v_{ij} = 1, \quad \forall i \\
&\quad \sum_{i=1}^{M} [20w_{i1} + 30w_{i2} + 50w_{i3}] \leq B \\
&\quad w_{ij} \in \{0,1\}, \quad \forall i,j \\
&\quad v_{ij} \in \{0,1\}, \quad \forall i,j.
\end{aligned} \tag{4}
$$

Here B is the given budget amount, $M_2 = M - M_1$ is the total number of target customers who are predicted to use the voucher, and d_i and t_i are known from the user's profile and the predicted duration of return.

After the Problem (4) is solved with the optimum (w_{ij}^*, v_{ij}^*), the optimal strategy for the user i is then to send a voucher with discount value $20w_{i1}^* + 30w_{i2}^* + 50w_{i3}^*$ and minimum spend requirement $1000000v_{i1}^* + 1500000v_{i2}^* + 2000000v_{i3}^*$.

4 Implementation Details

In this section, we report the implementation details for our two main tasks. All results were generated on desktop with Intel Core i5-4570 CPU of 3.20 GHz clock speed and 8 GB RAM. For data base manipulation and feature selection, we use open-source relational database management system MySQL with the administration tool HeidiSQL and Matlab, respectively. For classification and optimization tasks, we use Matlab and Python jointly.

We list the details of the implementation codes in Table 4.

Code name	Description
pre-process-sqls.sql	Extract the raw data for the training.
pre-process-sqls-predict.sql	Extract the raw data for the prediction.
last-purchase-sqls.sql	Extract the raw data for the reactivation optmisiation.
pre_process.m	Generate features for training data.
pre_process_predict.m	Generate features for predict data.
logistic_regression.m	Perform logistic regression.
integer_programming.m	Optimizing voucher strategy.
neural_network.py	Deep neural network prediction.

Table 4: Codes description.

We list all data resulting from our implementations in Table 5.

Data name	Description	Generating code/logging files
raw-data.csv	Raw data for training.	pre-process-sqls.sql
raw-data-predict.csv	Raw data for predicting.	pre-process-sqls-predict.sql
raw-data-last-purchase.csv	Raw data for reactivation optimisation.	last-purchase-sqls.sql
x_logistic.mat	Features for logistic regression.	pre_process.m
y_logistic.mat	Labels for logistic regression.	pre_process.m
x.mat	Features for neural network.	pre_process.m
y.mat	Labels for neural network.	pre_process.m
x_logistic_predict.mat	Features for logistic regression predication.	pre_process_predict.m
y_logistic_predict.csv(.m)	Predicted labels by logistic regression.	logistic_regression.m
x_nn_predict.m	Features for neural network prediction.	logistic_regression.m
y_nn_predict.csv	Predicted labels by neural network.	neural_network.py
voucher_strategy_result.csv	Voucher strategy.	integer_programming.m and Excel
predict_result.csv	Results for *predict.csv*.	Excel manipulation

Table 5: Data description.

4.1 Logistic Regression for Voucher Usage

We use Matlab and the build-in function *fminunc* to perform the logistic regression. We note that there is data imbalance in the training data (there are only 23453 users have used voucher). To handle this issue, we randomly select the training data such that the ratio between number of users who haven't used voucher and the number of users who have used voucher is $2:1$.

The training time is 2.9 seconds and the accuracy is 67.56%. The resulting optimal parameter $\theta \in \mathbb{R}^{27}$ is

$$\begin{aligned}\theta =&[-0.0863\ -0.1280\ -0.1408\,0.0450\ -0.3020\,0.5729\,0.0969\ -0.2394\,0.3169\ -0.2435\,...\\ &0.2346\,2.1223\ -0.1842\,0.0168\ -0.4191\ -0.1171\,0.1175\ -0.0235\ -0.0541\,0.0609\,...\\ &0.1761\,0.2808\,0.3601\ -0.2533\ -0.1664\ -0.0919\ -0.0046].\end{aligned}$$

We then use the trained logistic model to predict the voucher usage for the predict data, and construct the feature vector $\mathbf{z}$ defined in (2).

4.2 Neural Network for Repurchase

A deep neural network (DNN) is an artificial neural network with multiple hidden layers between the input and output layers (see Figure 1). In this subsection, we train a DNN in Python using TensorFlow open-source library for the repurchase classification task.

We design a DNN containing 5 hidden layers with activation function *relu*: $\max(0, x)$. The numbers of neurons for each layer are 128, 256, 512, 256 and 128, respectively. The output layer contains neurons with *softmax* activation function: $e^{z_j} / \sum_{k=1}^{K} e^{z_k}$ for $j = 1, \ldots, K$. In our case, $K = 5$.

The training time is 305 seconds, and the cross-validation accuracy is 49.5%.

4.3 Reactivation Rate Maximization

We solve the problem using Matlab mixed-integer linear programming solver *intlinprog*. We let the fixed budget $B = 100000$. The running time is 0.96 seconds.

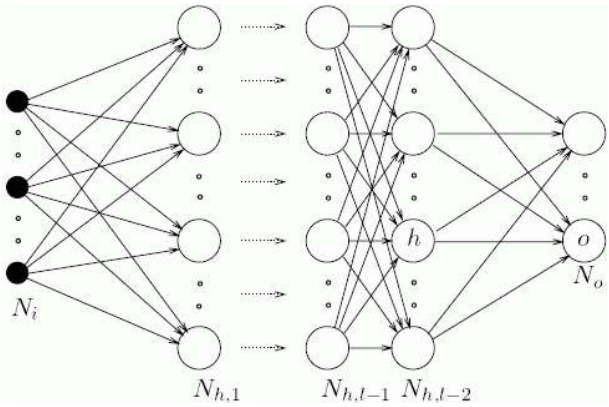

Figure 1: Illustration of a deep neural network.

5 Results and Discussions

The results for *predict.csv* can be found in *predict_result.csv* by manipulating *predict.csv*, *y_logistic_predict.csv* and *y_nn_predict.csv* in Excel. The voucher strategy for *predict.csv* can be found in *voucher_strategy_result.csv* by manipulating *predict.csv* and *voucher_strategy.csv* in Excel.

5.1 Results and Discussions for Logistic Regression

We have predicted that there are 1479 people likely to use the voucher, a 1.87% of the total population. We illustrate the results with an example. The users with id 4327100 and 4681858 have same number of active sessions previously (both are 2), but the first user has a complete profile and has spent 6 times more money on shopping (67181000 compared with 11482000). As expected, we predict the first user will use the voucher while the other does not as the former one is more "loyal" to the platform.

We remark that the training accuracy can be improved if we use the nonlinear model, e.g., wavelet-based approximations, SVM in its kernel form or multi-layer neural network. The linear parameterization $\theta^\top \mathbf{x}$ we used here may lead to poor performance.

5.2 Results and Discussions for Neural Network

The cross-validation accuracy for the trained neural network is 49.5%, which indicates a low degree of overfitting in comparison with the training accuracy 50.1%. We have obtained some desired results. For example, the users with id 8062318 and 8209876 have a similar profile: ages are 28.2 and 23.7 in a same range, registration durations up to now are both 1.3 years and the genders are both (predicted) female. However, the former one has spent more money on the website (359761000 compared to 10973000) and has more active sessions than the later one (678 compared to 39). As expected, the neural network approach predicts the former one is likely to come back and repurchase. In fact, we predict the former one will repurchase within 15 days while the other dose not.

While the present approach achieves some good performance, we may improve its accuracy by adjusting the hyper-parameters such as the learning rate.

5.3 Results and Discussions for Optimisation

The algorithm turns out to be time-efficient, and it yields three different voucher strategies: discount 20 with max_value 2×10^6, discount 50 with 1×10^6 and no discount.

In practice, we may not be able to send as many vouchers as possible to users who are willing to use them. Moreover, we also need to perform fraud detection on the group of users. For example, we should not give a voucher to the one who is actually the seller of a certain product. It is also worthwhile to identify those users who are suspicious identity thefts. This fraud detection requires supervised machine learning algorithms and a labeling process, and will be beneficial to our discount optimisation.

After performing fraud detection, we can rank the potential users according to their voucher adoption rates. In particular, we will make use of *raw-data-last-purchase.csv*, calculate the adoption rate for each user according to (3), and send n vouchers to the top n users.