Big Data and Data Analysis Hw1

B00902042 詹舜傑
B00902048 吳瑞洋
B00902064 宋昊恩
R02922164 邵　飛

April 17, 2015

1 Proposed solution

We try to use two significantly different methods to model this problem. They can be categorized as Machine Learning based algorithms ans Statistics & Regression based algorithms. We will briefly introduct our motivation and implementation details of these methods, respecitively.

1.1 Machine Learning based algorithms

There are several ways to derive models to this problem, such as using *Support Vector Machine(SVM)*[1], *Random Forest(RF)*[2], *Gradient Boosting Machine(GBM)*[3], and other feature-based model. However, we can soon find out that it is hard to model the law of *diminishing marginal utility*, since these feature models use determined weights to identify the importance of certain feature.

Instead, we would like to use some collaborative models, such as *Matrix Factorization(MF)*[4], *Tensor Decomposition(TD)*[5] to solve these problems. The assumption of matrix factorization is fulfilled by: similar invested media with the same amount of money can get similar feedback money; the difference of invested money with the same invested media can also be revealed by the difference of feedback money.

We will discuss the problems with time and without time separately. Thus, the first problem is about:

- money, and invested media
- invested money, invested media, and invested day.nd invested day

First, we would like to give the definition of different notations as follows.
S_t: sum of invested money before and including day t
F_t: sum of feedback money before and including day t
S_i: latent vector for discrete invested money
M_i: latent vector for invested media
D_i: latent vector for invested day
L_S: slots number of discrete invested money
L_M: types number of invested media
L_D: days number of invested day (maybe referred to 365 here)
K: length of latent features

1.1.1 MF model with invested approach and invested money

In this model, there is no difference in terms of locations and days. In other words, we sum up the invested money for each invested approach throughout a whole year. To be more specific, if it feedbacks v_i revenue after investing a sum of s_i on an invested media m_i on some day, we will append (s_i, m_i, v_i) to a list of tuples L at training stage. After the model learning, we can apply a dynamic programming (DP) algorithm to find out the best distribution of invested money with time-complexity of $\mathcal{O}(K \times L_S^2 \times L_M)$.

Algorithm 1 MF Learning and DP referring

function MF(L)
    // MF learning steps are standard, and thus omitted
    return M, S ▷ $M^{|L_M| \times K}$, $S^{|L_S| \times K}$ are latent vectors with size K

function DP_Reference(m, r) ▷ now is considering media m with r money remains
    // If it has been visited, return calculated value directly
    if $m == L_M - 1$ **then**
        return 0
    $ret = 0$
    for $i = 0$ to r **do**
        $ret = max(ret, \text{DP\_Reference}(m+1, r-i) + M_m^T S_i)$
    return ret

1.1.2 TD model with invested approach, invested money and invested day

This approach is similar to our previous model, however, we utilize more time information. To be clear, in previous model, we sum up the invested money and use it to locate one of the latent vectors in Q. That is to say, for a certain approach a_i, we cannot differentiate the length of time interval between different instances but only observe the increasing trend for both invested money and feedback revenue. To solve this problem, we separate the time factor into a new dimension, and thus we model this problem with a $3D$ *MF* model, which usually called *TD* problem.

After the learning process of *TD*, we can apply a similar DP algorithm to find out the best allocation with time-complexity $\mathcal{O}(K \times L_S^2 \times L_M \times L_D)$.

Algorithm 2 TD Learning and DP referring

function TD(L)
    // TD learning steps are standard, and thus omitted
    return M, S, D ▷ $M^{|L_M| \times K}$, $S^{|L_S| \times K}$, $D^{|D_S| \times K}$ are latent vectors with size K

function DP_Reference(m, r, d) ▷ now is considering media m with r money remains
    // If has been visited, return calculated value directly
    if $m == L_M - 1$ **then**
        return 0
    $ret = 0$
    if $m == L_M - 1$ **then**
        for $i = 0$ to r **do**
            $ret = max(ret, \text{DP\_Reference}(0, r-i, d+1) + M_m^T S_i + S_i^T D_d + D_d^T M_m)$
    else
        for $i = 0$ to r **do**
            $ret = max(ret, \text{DP\_Reference}(m+1, r-i, d) + M_m^T S_i + S_i^T D_d + D_d^T M_m)$
    return ret

1.1.3 Answers to questions

1. What is your definition of best allocation?
   With the DP algorithm, we can guarantee the best allocation for both model.
2. What is the dependent variable in your model ?
   - invested media: dependent variable in both *MF* and *TD*
   - invested money: dependent variable in both *MF* and *TD*
   - invested day: dependent varaible in *TD*
3. What are the independent variables ?
   - invested day: independent varaible in *MF*

1.2 Statistics & Regression based algorithms

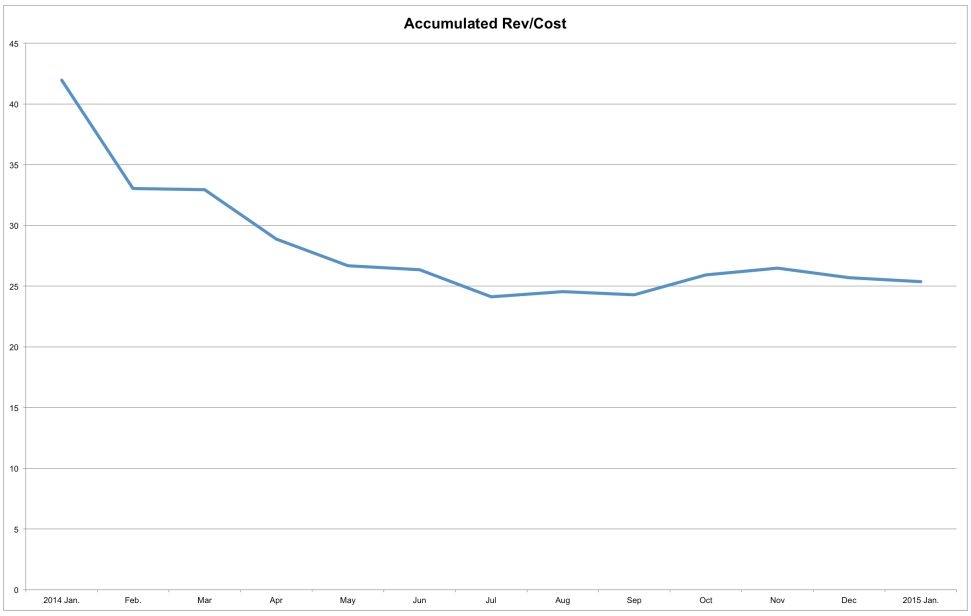

Figure 1.1: Income / Expense

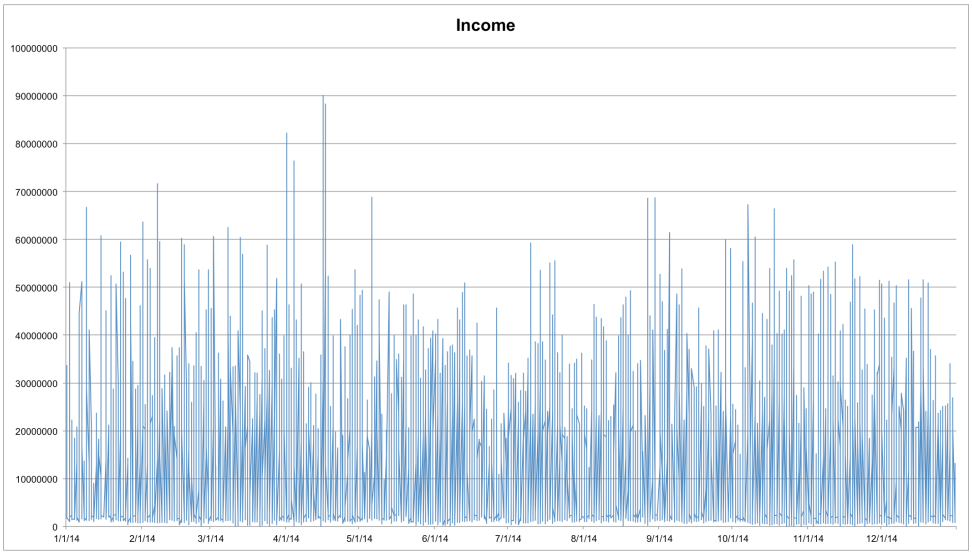

Figure 1.2: Income

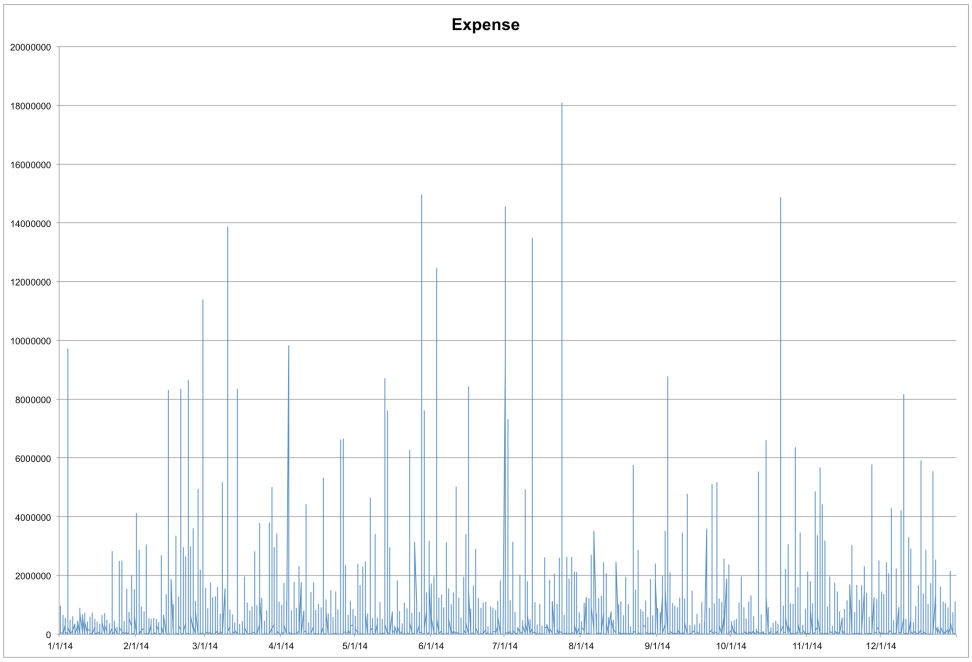

Figure 1.3: Expense

1.2.1 Basic Method (ROI Model)

Using *Return on Investment (ROI)* of each media as the benchmark to rearrange budget.

We first give several notation definitions: S_0 and S_1 are the total budget this year and next year; m_i means the i-th media, and $F(i)$ calculates the budget addition on m_i next year. Furthermore, we define income and expense of m_i last year to I_i^y and E_i^y, respectively. Then, we can get the formula:

$$F(t) = \frac{1}{\sum_i \frac{I_i^y}{E_i^y}} (\frac{I_t}{E_t})(S_1 - S_0) \tag{1.1}$$

1.2.2 Advanced Method (Revenue/Cost model)

Using max ratio of accumulated income and expenses in a year as benchmark to rearrange budget.

We define the income and expense of m_i on month m last year to I_i^m and E_i^m, respectively. Thus, for each media i, we would like to find the month M, such that $\sum_{m=1}^{M} \frac{I_i^m}{E_i^m}$ has the maximum value, said v_i^{MAX}. Then, we have $F(t) = \frac{v_t^{MAX}}{\sum_i v_i^{MAX}} v_t^{MAX}(S_1 - S_0)$

1.2.3 Answers to questions

1. What is your definition of best allocation?
   We use a utility function (1.1) to model the income next year, which provides the best allocation.
2. What is the dependent variable in your model ?
   - media: dependent variable in our *ROI model*
   - time: dependent variable in our *Revenue/Cost model*, i.e. we sort revenue by month interval
3. What are the independent variables ?
   - location: independent variable in both our *ROI model* and *Revenue/Cost model*

2 Reference

[1]: Gunn, Steve R. "Support vector machines for classification and regression." ISIS technical report 14 (1998).
[2]: Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[3]: Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.
[4]: Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 8 (2009): 30-37.
[5]: Kolda, Tamara G., and Brett W. Bader. "Tensor decompositions and applications." SIAM review 51.3 (2009): 455-500.