A Least Square based Model for Rating Aspects and Identifying Important Aspects on Review Text Data

Pham Duc Hong, Le Anh Cuong*

K20 - Computer Science - University of Engineering and Technology, VNU, Hanoi, Vietnam

*Ton Duc Thang University, HoChiMinh city, Vietnam

Abstract

In recent years, opinion mining and sentiment analysis has been one of the attracting topics of knowledge mining and natural language processing. The problem of rating aspects from textual reviews is an important task in this field. In this paper we propose a new method for rating product aspects as well as for identifying important aspects in general. Our proposed model is based on the least square method and the QR decomposition technique. In our experiment, we use a dataset of 594810 reviews of 3775 hotels collected from the very famous website in tourism tripadvisor.com with five common aspects including *cleanliness, location, service, room* and *value*. Experimental result shows that our proposed method outperforms some well known studies for the same problem.

Main Objectives

Let $D = \{d_1, d_2, ..., d_{|D|}\}$ be a set of review text documents for a considering entity (i.e. hotel), where each review document *d* is associated with an overall rating (i.e denote O_d). We assume all aspects using the same dictionary, named *V*. This dictionary contains *n* words which express opinions. Suppose that the set of reviews *D* have *k*-aspects. Denote $\{A_1, A_2, ..., A_k\}$ is a set of aspects, where an aspect A_i is a set of words that characterize a rating factor in the reviews.

For each review document *d*, we define the following symbols:

$r_d = (r_{d1}, r_{d2}, ..., r_{dk})$ is a *k* - dimensional vector of *k* aspect rating, where the *i*-th dimension is a numerical measure, indicating the degree of satisfaction demonstrated in the review *d* toward the aspect A_i.

$\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)$ is a *k* - dimensional vector of overall aspect weights for all reviews, (i.e. this notation is the overall aspect weights in general), where the *i*-th dimension is a numerical measure, indicating the degree of importance corresponding to aspect A_i, where we require $0 \le \alpha_i \le 1$ and $\sum_{i=1}^{k} \alpha_i = 1$.

$x_{di} = (x_{di1}, x_{di2}, ..., x_{din})$ is a feature vector presentation for aspect A_i in review d.

$w_i = (w_{i1}, w_{i2}, ..., w_{in})$ is a vector indicates the word sentiment polarities on aspect A_i.

Methods

We propose a model based on least square to identify important aspects as well as aspects rating (or ranking aspects) from consumer reviews. This model includes the training phase and testing phase.

For the training phase, using both known aspect ratings and overall rating of each review in data set D to learn the overall aspect weights. We have the squared residuals function for overall aspect ratings as follows:

$$E(\alpha) = \sum_{d \in D} s_d(\alpha)^2 \quad (1)$$

where $s_d(\alpha) = O_d - \sum_{i=1}^{k} \alpha_i r_{di}$

Suppose $\sum_{i=1}^{k} \alpha_i = 1$ and $0 \le \alpha_i \le 1$ instead of variable α_i as follows:

$$\alpha_i = \frac{\exp(\hat{\alpha}_i)}{\sum_{l=1}^{k} \exp(\hat{\alpha}_l)} \quad (2)$$

The function $E(\alpha)$ with α is unknown variable becomes $E(\hat{\alpha})$ with $\hat{\alpha}$ is unknown variable as follows:

$$E(\hat{\alpha}) = \sum_{d \in D} s_d(\hat{\alpha})^2 \quad (3)$$

where $s_d(\hat{\alpha}) = O_d - \sum_{i=1}^{k} \frac{\exp(\hat{\alpha}_i)}{\sum_{l=1}^{k} \exp(\hat{\alpha}_l)} . r_{di}$

The goal is to determine the variable values $\hat{\alpha}$ to the function $E(\hat{\alpha})$ reaches the minimum value, this is the problem nonlinear square optimization, it has no closed-form solution and is solved by iterative algorithm, at each iteration algorithm is approximated by a linear one. Specifically, at iteration *t*-th we rewrite the error function $E(\hat{\alpha})$ from (3) as follows:

$E(\hat{\alpha}^{(t)}) \approx \left\| A^{(t)} \hat{\alpha} - b^{(t)} \right\|^2$ where

$$A^{(t)} = \begin{bmatrix} \Delta s_1(\hat{\alpha}^{(t)})^T \\ \Delta s_2(\hat{\alpha}^{(t)})^T \\ \vdots \\ \Delta s_{|D|}(\hat{\alpha}^{(t)})^T \end{bmatrix} \quad (4)$$

and

$$b^{(t)} = \begin{bmatrix} \Delta s_1(\hat{\alpha}^{(t)})^T \hat{\alpha}^{(t)} - s_1(\hat{\alpha}^{(t)}) \\ \Delta s_2(\hat{\alpha}^{(t)})^T \hat{\alpha}^{(t)} - s_2(\hat{\alpha}^{(t)}) \\ \vdots \\ \Delta s_{|D|}(\hat{\alpha}^{(t)})^T \hat{\alpha}^{(t)} - s_{|D|}(\hat{\alpha}^{(t)}) \end{bmatrix} \quad (5)$$

$\hat{\alpha}^{(t+1)}$ obtained at iteration *t*-th,

$$\hat{\alpha}^{(t+1)} = \arg min_{\hat{\alpha}} ||A^{(t)} \hat{\alpha} - b^{(t)}||_2 \quad (6)$$

To (6) address above constraint linear square optimization problem. We apply the QR Decomposition [1] to determine $\hat{\alpha}^{(t+1)}$. We denote $R = [r]_{|D|xk}$ as a review − aspect rating matrix which describes the rating of aspects in reviews; $O = \left(O_1, O_2, ..., O_{|D|}\right)$ is a vector of the overall rating of *m* reviews. $\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)$ is the overall aspect weights which is unknown variable.

We propose the algorithm for computing the overall aspect weights (i.e. it also aims to identify important aspects) in the Algorithm 1:

Algorithm 1 Identify the important aspects

Input: Matrix $R = [r]_{|D|xk}$, vector $O = \left(O_1, O_2, ..., O_{|D|}\right)$; Error threshold ε, Iterative threshold *I*

1. Initialization $\hat{\alpha}^{(0)}$
2. for *t*=1 to *I* -*1* do
   2.1. Compute $\alpha^{(t)}$ according to Eq. (2);
   2.2. Update $A^{(t)}$ according to Eq. (4);
   2.3. Update $b^{(t)}$ according to Eq. (5) ;
   2.4. Compute $E(\hat{\alpha})^{(t)}$ according to Eq. (3);
   2.5. Update $\hat{\alpha}^{(t+1)}$ according to Eq. (6);
3. For offline learning, the step 2 may be repeated until the iteration error $E(\hat{\alpha})$ is less than the error threshold ε or the predetermined number of iterations have been completed.

Output: α

We denote $X_i = [x]_{|D|xn}$ as a aspect review − term matrix of aspect A_i which describes the occurrences of terms of aspect A_i on reviews, it is the sparse matrix whose columns correspond to terms and whose rows correspond to aspect A_i on reviews. $r_i = (r_{i1}, r_{i2}, ..., r_{i|D|})$ denotes a vector of the aspect rating of aspect A_i on $|D|$ reviews. $w_i = (w_{i1}, w_{i2}, ..., w_{in})$ is a vector indicating the word sentiment polarities on aspect A_i which is unknown variable. The algorithm determines the vector w_i indicates the word sentiment polarities on aspect A_i are as follows:

Algorithm 2 Identify the vector w_i indicates the word sentiment polarities on aspect A_i

Input: Matrix $X_i = [x]_{|D|xn}$, vector $r_i = (r_{i1}, r_{i2}, ..., r_{i|D|})$;

$$w_i = \arg min_{w_i} ||X_i w_i - r_i||_2 \quad (7)$$

Output: w_i

To (7) address above constraint linear square optimization problem. We apply the QR Decomposition [1] to determine $W_i = (w_{i1}, w_{i2}, ..., w_{in})$ on aspect A_i .

Results

All the algorithms are evaluated on the same data set, we use the whole data set for both training and testing. We perform 4-fold cross validation and report the mean value of performance.

0.0.1 Evaluation on aspect rating prediction

We use three different measures [4] to quantitatively evaluate different methods as follows: (1) Mean square error on aspect rating prediction (Δ^2_{aspect}, lower is better); (2) Aspect correlation inside reviews (P_{aspect}, higher is better); (3) Aspect correlation across all reviews (P_{review}, higher is better). In Table 1, we show the results to compare with PRank algorithm [2].

Algorithm	Δ^2_{aspect}	P_{aspect}	P_{review}
PRank	0.571	0.526	0.687
Our algorithm 2	**0.214**	**0.652**	**0.752**

Table 1: Compare the predicted results of aspect rating

0.0.2 Evaluation on weighting aspects prediction

The combination of the predicted results of aspect rating and the inference results of aspect weight help us to predict overall rating. We use mean square error on overall aspect rating prediction ($\Delta^2_{Overall}$, lower is better) to evaluate the results of inference aspect weight through four cases following: (1) PRank algorithm+Our algorithm 1, (2) PRank algorithm+PRR algorithm [4], (3) Our algorithm 2+PRR algorithm and (4) Our algorithm 1+Our algorithm 2. In Table 2, we show the comparison results between some methods.

Algorithm	$\Delta^2_{Overall}$
PRank algorithm+Our algorithm 1	0.402
PRank algorithm+PRR algorithm	0.425
Our algorithm 2+PRR algorithm	0.409
Our algorithm 1+Our algorithm 2	**0.362**

Table 2: The differences of overall ratings predicted with ground-true ratings

Through Table 2, we can see that the combination of our proposed algorithms 1 and 2 gives the slightly better result. These results indicate the overall aspect weights (i.e. important aspects) are determined by the PRR algorithm is not good as our algorithm 1.

Conclusions

In this paper, we have proposed a new method based on Least Squares model using both known aspect ratings and the overall rating of reviews to identify the overall aspect weights directly from numerous consumer reviews. Through experimental results, we have demonstrated that our proposed algorithm 1 for determining important aspects (i.e. determining aspect weights) is better than the probabilistic regression algorithm (i.e. PRR algorithm). We also have proposed the algorithm 2 for aspect rating, and the obtained experimental results show that it is better the PRank algorithm.

References

[1] The Math Works, Inc. and the National Institute of Standards and Technology. Software available at http://www.codeproject.com/Articles/5835/DotNetMatrix-Simple-Matrix-Library-for-NET.

[2] K.Crammer and Y.Singer, Pranking with ranking, In Proceedings of NIPS, 2001, pp.641-647.

[3] Duc-Hong Pham, Anh-Cuong Le and Thi-Kim-Chung Le, "A least square based model for rating aspects and identifying important aspects on review text data", Conference on Information and Computer Science (NICS), 2nd National Foundation for Science and Technology Development, 2015.

[4] H.Wang, Y.Lu and Ch.Zhai, Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach, In Proceedings of SIGKDD, 2010, pp.168-176.