2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)

# Accurate delivery analysis of distributed e-commerce based on Word2vector

Lin Long
Shanxi Finance and Economics University
School of Information Management
Taiyuan, China
410420589@qq.com

***Abstract*—The e-commerce market is fiercely competitive nowadays. How to achieve accurate customer delivery has become a key issue in e-commerce research. The purpose of this paper is to optimize the e-commerce advertising and push delivery based on the current customer search text big data. In this paper, the Word2vector algorithm is used to calculate the big data from Alibaba customers. The customers' search preferences, search histories and other related text content are used and analyzed. The research results show that the Word2vector algorithm is faster and more effective than the current mainstream CTR algorithm.**

***Keywords-Target; Distributed E-commerce; Word2vector***

## I. Introduction

In the age of the Internet, to identify your own customer base, the first thing to do is to find your customers. How to find customers? Of course, this is a network era, you might say, it is very simple to find customers [1]. But simplicity does not mean high efficiency. In the past, everyone used big data to find customers. It not only wastes time, but also consumes a lot of manpower and financial resources, and the results are always unsatisfactory.

How to get a bigger return with less investment has been a major issue for advertisers. Recently, "accurate delivery" has become a major solution to this problem. As the name implies, accurate delivery means ‘accurate advertising to the people who are most likely to buy [2][3]. It is better for users to click on the order when they see the ad in the web. Under this expectation, advertisers can hardly be tempted when someone tells him: ‘We can advertise more accurately, who you want to vote for, and who you can vote for’.

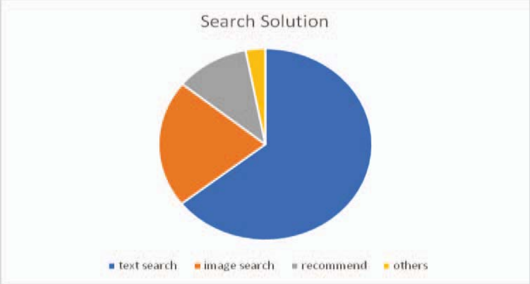


Fig. 1 User Search Behavior Distribution

The data in Fig.1 come from the statistics of Alibaba. From the data, it can be found that currently users are more likely to search information through text input. Therefore, the accurate delivery based on user text input is a more feasible and practical way. Inspired by this, This paper uses the Word2vector algorithm [4] to embed the words of user search information into vectors which can be used to get accurate delivery. Fig.2 is the framework of the research process.

Algorithm mathematical model
Feasibility Analysis
Instance implementation

Fig. 2 Research Process

## II. Algorithm

The Word2vector tool mainly consists of two models: a skip-gram and a continuous bag of words (CBOW), and they use two different training methods respectively: negative sampling and sequence softmax (Hierarchical Soft-max). It is worth mentioning that the vectors studies by Word2vector can better express similarities between different words.

Natural language is a complex system used to express meanings. In this system, words are the basic units of meanings. In machine learning, how to use vectors to represent words? As the name implies, a word vector is a vector used to represent a word, and is usually also considered as the feature vector of the word. In recent years, word vectors have gradually become a basic technique of natural language processing.

In NLP (Natural Language Processing), the most fine-grained semantic units are words. Words form sentences, and sentences form paragraphs, chapters, and documents. So, to deal with the problem of NLP [5], we must first deal with the words. To process words more effectively and semantically, words should be converted into numerical forms, *i.e.* embedded into a mathematical space, which is called word embedding. Word2vector is a famous method of word embedding. In simple terms, through word embedding a word is converted into an expression of the corresponding vector so as to make it easier for machine to read or process the word.

978-1-7281-5094-9/19/$31.00 ©2019 IEEE
DOI 10.1109/MLBDBI48998.2019.00020

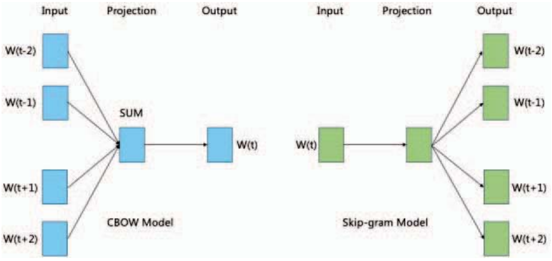
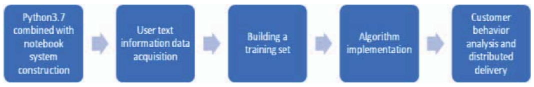


Fig. 3 Algorithm Structure

How could we calculate the probability that a text sequence will appear in the language [6]? It is a basic problem because it plays an important role in many NLP tasks. For example, in the problem of machine translation, if we know the probability of each sentence in the target language, we can pick the most reasonable sentence from the candidate set and return it as the translation result.

## III. Algorithm Implementation

In the calculation part of the example, the algorithm is implemented in Python. The process of the algorithm implementation is shown in the Fig.4.

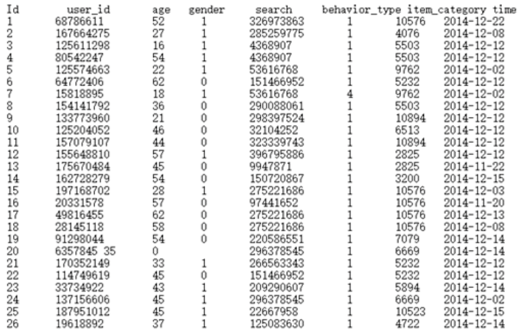


Python3.7 combined with notebook system construction → User text information data acquisition → Building a training set → Algorithm implementation → Customer behavior analysis and distributed delivery

Fig. 4 Instance Algorithm Implementation Flow

### A. Dataset

The data set of this paper comes from 1000 high search customers from Alibaba public data base. Each customer has 10 search histories, a total of 10000 data points. And each user has their own user ID varying from 1 to 1000. Gender, active time, and keyword vector are included, also with the basic information such as the time, location, and the age of the customer's search. The experimental samples are reliable and the experiment is reproducible.



| Id | user_id | age | gender | search | behavior_type | item_category | time |
|---|---|---|---|---|---|---|---|
| 1 | 68786611 | 52 | 1 | 326973863 | 1 | 10576 | 2014-12-22 |
| 2 | 167664275 | 27 | 1 | 285259775 | 1 | 4076 | 2014-12-08 |
| 3 | 125611298 | 16 | 1 | 4368907 | 1 | 5503 | 2014-12-12 |
| 4 | 80542247 | 54 | 1 | 4368907 | 1 | 5503 | 2014-12-12 |
| 5 | 125574663 | 22 | 1 | 53616768 | 1 | 9762 | 2014-12-02 |
| 6 | 64772406 | 62 | 0 | 151466952 | 1 | 5232 | 2014-12-12 |
| 7 | 15818895 | 18 | 1 | 53616768 | 4 | 9762 | 2014-12-02 |
| 8 | 154141792 | 36 | 0 | 290088061 | 1 | 5503 | 2014-12-12 |
| 9 | 133773960 | 21 | 0 | 298397524 | 1 | 10894 | 2014-12-12 |
| 10 | 125204052 | 46 | 0 | 32104252 | 1 | 6513 | 2014-12-12 |
| 11 | 157079107 | 44 | 0 | 323339743 | 1 | 10894 | 2014-12-12 |
| 12 | 155648810 | 57 | 1 | 396795886 | 1 | 2825 | 2014-12-12 |
| 13 | 175670484 | 45 | 0 | 9947871 | 1 | 2825 | 2014-11-22 |
| 14 | 162728279 | 54 | 0 | 150720867 | 1 | 3200 | 2014-12-15 |
| 15 | 197168702 | 28 | 1 | 275221686 | 1 | 10576 | 2014-12-03 |
| 16 | 20331578 | 57 | 0 | 97441652 | 1 | 10576 | 2014-11-20 |
| 17 | 49816455 | 62 | 0 | 275221686 | 1 | 10576 | 2014-12-13 |
| 18 | 28145118 | 58 | 0 | 275221686 | 1 | 10576 | 2014-12-08 |
| 19 | 91298044 | 54 | 0 | 220586551 | 1 | 7079 | 2014-12-14 |
| 20 | 6357845 35 | 0 |  | 296378545 | 1 | 6669 | 2014-12-14 |
| 21 | 170352149 | 33 | 1 | 266563343 | 1 | 5232 | 2014-12-12 |
| 22 | 114749619 | 45 | 0 | 151466952 | 1 | 5232 | 2014-12-12 |
| 23 | 33734922 | 43 | 1 | 209290607 | 1 | 5894 | 2014-12-14 |
| 24 | 137156606 | 45 | 1 | 296378545 | 1 | 6669 | 2014-12-02 |
| 25 | 187951012 | 45 | 1 | 22667958 | 1 | 10523 | 2014-12-15 |
| 26 | 19618892 | 37 | 1 | 125083630 | 1 | 4722 | 2014-12-14 |

Fig. 5 Part of the data set screenshot

### B. Input layer

First, the user will enter the text information that he or she wants to search. On the e-commerce website, each user has his or her own text database. The database forms a basic data input layer, for example if a user input "fridge", information such as "home appliances" will form a data input layer of a fixed user ID [7]. When using Word2vector, each word is always represented by a vector, for example,

[0.31343242, 0.65464122, 0.12343425, ..., -1.324344]

If every word is expressed as a vector, the entire dictionary corresponding to a user ID will be a matrix, which is the final result of the training of the projection matrix and the iteration. If we set the matrix dimension D (hyper-parameter), the dimension of the projection matrix is $D*V$, $C*$ word vector (one-hot means that the nth column is taken in the projection matrix to represent the corresponding word), dimension Indicates that

$$D * V * V * 1 = D * 1$$

### C. Hidden layer

There is only one hidden layer in our algorithm. The results of the word vector multiplied by the projection matrix in the upper layer are input into this layer. The BP process needs to update the projection matrix C and the projection matrix H, here we use the sigmoid as activation function then go to the next layer.

### D. Output layer

Here is a soft-max layer, there are 30,000 classes in total. The probability of "home appliances" is predicted from the dictionary. This leads to a problem, if the dictionary is particularly large, then the parameters of this layer will explode [8] directly. In 2003, it was only used by the CPU, which caused the model to be not good, but today's computers make this process possible.

### E. Training process

The content required by the user can be obtained through part of the training process, and the data is verified through the training process result. After the verification, an output database is created for the client, which also corresponds to a customer ID. The following is the pseudo-code of our proposed algorithm.

1.The process of building different customer ID lexicons: the structure of words;

2. Network model initialization: negative sampling initialization, generate negative sampling probability table;

3. Start reading every word in the file;

4. Return a word hash value;

5. Find the position of the word in the thesaurus and find out if the word exists. Returns -1 if it does not exist, otherwise returns the index of the word in the thesaurus;

6. Sort by word frequency, key structure body comparison function;

7. The most frequently vocabulary;

8. Processing low frequency words;

9. Create a huffman tree: Create Binary Tree;

10. Return the user's vocabulary high frequency results and output, get the common lexicon.

### *F. User distributed and accurate database formation*

Through the output database of the user ID, the advertisement, content and other preferential information are built into a data warehouse [9], and feedback to the customer through the data warehouse form a distributed and accurate delivery. We compare the Word2vector algorithm with the traditional CTR algorithm commonly used in the industry. The CTR prediction solution is a generalized linear model LR (logistic regression) + artificial feature engineering. LR uses the Logit transform to map the function values to the 0~1 interval. The mapped function value is the estimated value of CTR [10].

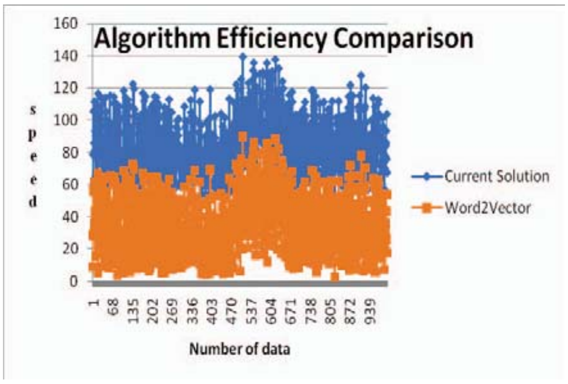


Fig. 6 Processing Speed Comparison

Figure 6 is a comparison of the calculation speeds of two algorithms, in which the yellow part represents the results of the algorithm used in the article and the blue part is the results of the CTR algorithm. By comparison, it can be seen that the calculation speed and delivery speed of the vector algorithm used in this paper are much faster.

At the same time [11], the algorithm studied in this paper uses a special customer database to predict customer behaviors, and it is also a kind of optimization about the products or advertisements while the previous research can only push customers.

## IV. Summary

This paper establishes a accurate delivery process through Word2vector's research on customer search text. The research results show that the text vector algorithm can not only process text information, but also realize distributed computation through the joint function of customer database, which speeds up the e-commerce website with faster marketing speed.

For the future development direction, the text recognition application scope will be smaller, and the precision delivery based on image and voice will be increased. At the same time, from the perspective of e-commerce, the content of the products, advertisements, etc. will be more diversified, and based on the customer's unique ID and database, the prediction of customer preferences will become a research hot spot, which can guarantee more accurate delivery to the real customers.

## References

[1] Nose, M., Maki, S., Yamane, N., Morikawa, Y.. N-best vector quantization for isolated word speech recognition[P]. SICE, 2007 Annual Conference,2007.

[2] Rong X. Word2vector Parameter Learning Explained[J]. Computer Science, 2014.

[3] Pei J, Li J. A corpus-based investigation of modal verbs in Chinese civil-commercial legislation and its English versions[J]. International Journal of Legal Discourse, 2018, 3(1):77-102.

[4] MatthewR. Banghart, Bernardo L. Sabatini. Photoactivatable Neuropeptides for Spatiotemporally Accurate Delivery of Opioids in Neural Tissue[J]. Neuron, 2012, 73(2):249-259.

[5] Hu J. E-commerce big data computing platform system based on distributed computing logistics information[J]. Cluster Computing, 2018(99):1-10.

[6] Wen S, Wei H, Yang Y, et al. Memristive LSTM Network for Sentiment Analysis[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, PP (99):1-11.

[7] Bakhache B, Ghazal J M, Assad S E. Improvement of the Security of ZigBee by a New Chaotic Algorithm[J]. IEEE Systems Journal, 2014, 8(4):1024-1033.

[8] Suguna S, Vithya M, Eunaicy J I C. Big data analysis in e-commerce system using HadoopMapReduce[C]// International Conference on Inventive Computation Technologies. 2016.

[9] Aboutorabi S H, Rezapour M, Moradi M, et al. Performance evaluation of SQL and MongoDB databases for big e-commerce data[C]// International Symposium on Computer Science & Software Engineering. 2016.

[10] Lilleberg J, Yun Z, Zhang Y. Support vector machines and Word2vector for text classification with semantic features[C]// IEEE International Conference on Cognitive Informatics & Cognitive Computing. 2015.

[11] Xue B, Chen F, Zhan S. A Study on Sentiment Computing and Classification of Sina Weibo with Word2vector[C]// IEEE International Congress on Big Data. 2014.