Age prediction in social networks based on word embedding and Tensor learning

Zi-Yi Lin and Yan Wang
No.1 Middle School affiliated to Central China Normal University, Wuhan, China
Department of Systems Engineering and Engineering Management, City University of Hong, Kong, Hong Kong, China
Email: ywang462-c@my.cityu.edu.hk

Latent attribute prediction problem in social network provides a set of conditions for the construction of text classification models. The general framework of current latent attribute prediction problems is mapping the text in social network to vector space, along with a classification model to classify different categories. Unfortunately, as the vector space model ignores the similarity and relevance between different words, it fails to identify the semantic fuzziness in natural language and always performs badly on a long text. With the aim of finding a better framework for age prediction problem, in this paper we propose a word embedding based tensor space model which maps text to tensor feature space. The proposed method relies on supervised tensor learning algorithms which are well studied by many scholars, thus allowing for its easy application in text classification problems. Two experiments on different testing sets show the effectiveness and limitation of our approach.

Keywords: Tensor Space Model; Word Embedding; Age Prediction; Tensor Learning; Social Network.

1. Introduction

With the recent explosion of social network, studies have increasingly focus on predicting latent attributes of users from large amounts of freely available content. These latent attributes include user gender[1],{[3] age,[4[location,[5] political preferences,[6] etc. Most studies are conducted on a labeled data set, and thus formulated the latent attributes prediction as a text classification problem.

Current studies always adopt the vector space model (VSM)[7] to represent the text in social media as vectors of identifiers, along with a classification model to classify different categories. The VSM is a straightforward practical method in many text-related applications, such as text classification[8], information filtering[9] and in-formation retrieval[10]. Unfortunately, the VSM only records the frequency information of words, ignoring the similarity and relevance between words. It has following limitations:

●The large dimensionality always leads to the curse of dimensionality, and thus affects the performance of text classification and regression;

●Search keywords must precisely match document terms: word substrings might result in a \false positive match";

●Semantic sensitivity; documents with similar context but different term vocabulary won't be associated, resulting in a \false negative match";

●It theoretically assumes terms are statistically independent.

●To deal with these limitations, we propose the tensor space model (TSM) based on word embedding. Word embedding is a feature learning technique which learns a distributed representation model that mapping the words in vocabulary to vectors of real numbers in a low-dimensional space. The methods to generate this distributed representation model include neural network[11], dimensionality reduction on the word co-occurrence matrix[12], and explicit representation in terms of the context in which words appear[13]. On the basis of word embedding, the proposed TSM transforms each document to a 2-d tensor (matrix) whose rows correspond to the

●Words in documents and columns correspond to the vectors in distributed model.

●Compared to VSM, TSM has following advantages:

●It effciently reduces the dimensionality and the number of free parameters in VSM;

●It utlizes the similarity and relevance information between words, to better handle the semantic fuzziness in natural language;

●In this paper, the TSM is applied to an age group classification problem in Sina weibo. Sina weibo is one of the most popular social media in China, with more than 200 millions active users in January 2016. We collect a corpus containing more than 30 millions tweets from about 650 thousands users, and then select tweets of 2127 labeled users from this corpus as our labeled data set. The rest of this paper is organized as follows. In section 2, we describe how we collect data and embedding words. In section 3, we show the details of proposed TSM, along with the support tensor machine algorithm for classifying age groups. Section 4 presents three experiments to prove the effectiveness of proposed method. In section 5, we conclude with a summary.

2. Data and Pre-Training

2.1. *Data collection*

The data in this study are obtained from the official open API provided by Sina

weibo. We collected a corpus which contains the tweets and tag information of 650,000 users. The tag information, as a special personal information in Sina weibo, is a group of tags to describe the most typical characters of user, as shown in figure 1. User can select tags from a given keyword list or edit new ones. Through tag information, we can find that there are 3537 users have the tag \post-80s", and 2156 users have the tag \post-90s". In our study, we assume that these two tags

Fig. 1 Example of Tag Information

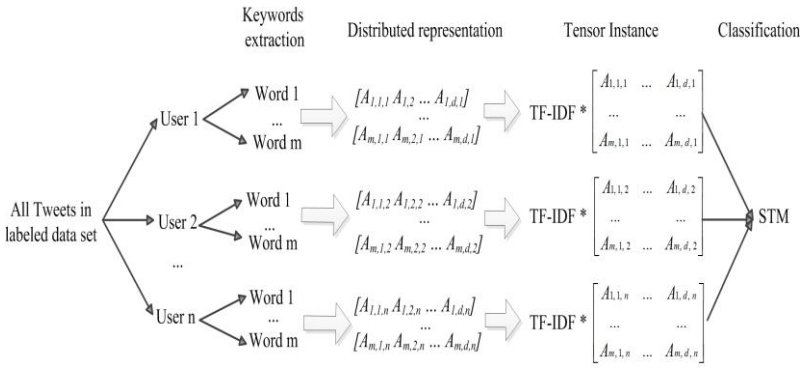

Fig. 2 Framework of Tensor Space Model

Indicate the birth year of users, and thus construct a labeled data set containing two age groups: post-80s group and post-90s group.

In this way, we formulate the age prediction problem as a text classification problem based on a labeled data set. To ensure the effectiveness of the labeled data set, we delete those users with less than 10 tweets from our data set. Finally, the data set in our study contains about 535 thousands tweets from 2127 users (1380 in post-80s group, and 747 in post-90s group).

2.2. *Word embedding*

In our study, corpus we obtained from Sina weibo is used to train a word distributed representation model. Before training the model, we need to segment the Chinese tweets into words, so that we can identify them in word embedding process. We adopt an open source Chinese word segmentation tool NLPIR[14] to segment the Chinese tweets in corpus.

In addition to the Chinese word segmentation, another open source tool word2vec from Google is adopted to train a word distributed representation model[15]. In this process, all words in the labeled data set will be successfully mapped to distributed vectors. Finally, we have trained a 40-d distributed representation model which contains vectors of 161427 words.

3. Tensor Space Model

In this section, the tensor space mode (TSM) is presented for addressing our text classification problem. To reduce the feature dimensionality and utilize similarity and relevance information between words, the proposed TSM shown in Fig. 2, in its essential, is a collection of words' distributed vectors. To ensure each sample in TSM has a uniform dimensionality, the first step is extracting a fixed number of keywords of each users. Then, tweets of each user is represented as a matrix instance (2-d tensor) which contains the distributed vectors of keywords. In figure 2, m denotes the number of keywords of each user, d denotes the dimensions of the distributed representation model, and n represents the number of samples in labeled data set. After tensor representation, the support tensor machine16 is adopted as our classifier.

3.1. *Keywords extraction*

After word embedding, as the number of words in tweets of different users are different, we cannot just simply combine the distributed vectors of all words together, or it will lead to a diverse dimensionality in feature space. To ensure a uniform dimensionality, we only extract a fixed number of representative keywords for each user. In this paper, the term frequency - inverse document frequency (tf-idf)[17] method is adopted to extract the keywords, which is briefly introduced as follows: The term frequency (tf) of term i in document d is calculated as:

$$tf(i,d) = \frac{n_{i,d}}{\sum_k n_{k,d}} \quad (1)$$

Where $n_{i,j}$ is the times of term i occurred in document d, and $\sum_k n_{k,j}$ is the number of words in document d.

The inverse document frequency (idf) is the measure of one term's importance in the corpus. The idf of term *i* in corpus D is calculated as:

$$\mathrm{tfidf(i,d,D)} = \log \frac{|\mathrm{N}|}{1+|\{\mathrm{d}\in \mathrm{D}:\mathrm{i}\in \mathrm{d}|} \quad (2)$$

Where N denotes the total number of documents in the corpus, $|\{d \in D: i \in d|$ denotes the number of documents where the term i appears (tfi;d > 0).

Then tf-idf of term i in document d of corpus D is calculated as:

$$tfidf(i, d, D) = tf(i, d) * idf(i, D) \tag{3}$$

3.2. *Tensor feature space*

To utilize the similarity and relevance information, we will use the collection of distributed word vectors to construct a new tensor feature space for text classification. However, it is not wise to simply collect the vectors of keywords together, because the importance of different keywords are not equal. To retain importance, the keywords of user U will be mapped to vectors as follows:

$$\begin{aligned} term_1 &\rightarrow tfidf(1, U, D) * [A_{1,1}, A_{1,2}, \ldots, A_{1,d}] \\ term_2 &\rightarrow tfidf(2, U, D) * [A_{2,1}, A_{2,2}, \ldots, A_{2,d}] \\ &\ldots \\ term_m &\rightarrow tfidf(m, U, D) * [A_{m,1}, A_{m,2}, \ldots, A_{m,d}] \end{aligned} \tag{4}$$

Then, a matrix instance (2-d tensor) for each user is contructed as follows:

$$X_U = \begin{bmatrix} tfidf(1,U,D) & 0 & \ldots & 0 \\ 0 & tfidf(2,U,D) & \ldots & 0 \\ \ldots & & & \\ 0 & 0 & \ldots & tfidf(m,U,D) \end{bmatrix} * \begin{bmatrix} A_{1,1}, \ldots, A_{1,d} \\ A_{2,1}, \ldots, A_{2,d} \\ \ldots \\ A_{m,1}, \ldots, A_{m,d} \end{bmatrix} \tag{5}$$

Finally, the tweets of all users will be transformed to a 3-d tensor feature space $X \in R^{m*d*n}$, where m denotes the number of keywords, d denotes the dimension of distributed representation, and n denotes the number of samples (users).

3.3. *Tensor learning*

One question remained in section B is that why we transform tweets to a tensor space $X_i \in R^{m*d}$, instead of a high-dimensional vector space $x_i \in R^{1*md}$. That is because tensor representation helps to reduce the number of free parameters in learning process. In this paper, we adopt the support tensor machine (STM)[16] as our classifier. For a training sample $X_i \in R^{m*d}$, the decision function of STM will be

$$f(X_i) = sign(\omega_1^T X_i \omega_2 + b) \tag{6}$$

where the projection vectors $w_1 \in R^m$, $w_2 \in R^d$and bias b are obtained from

$$\begin{aligned} \min_{\vec{\omega_1},\vec{\omega_2},b} \quad & \frac{1}{2}\|\vec{\omega_1} \otimes \vec{\omega_2}\|_{Fro}^2 + c\sum_{i=1}^{n}\xi_i \\ s.t. \quad & (\vec{\omega_1}^T * X_U * \vec{\omega_2} + b) \geq 1 - \xi_i, \quad 1 \leq i \leq n \\ & \vec{\xi} \geq 0 \end{aligned} \tag{7}$$

where $\otimes$ denotes the outer product, $\|*\|_{\text{Fro}}^2$ denotes the squared Frobenius norm, and $\vec{\xi} = [\xi_1, \xi_2, \ldots, \xi_n]$ is the vector of all slake vaiables to deal with the linearinseparable problem.

As the tensor learning model is not the main focus of this paper (but will be the focus of our further studies), the optimization of STM will not be introduced in this paper[18].

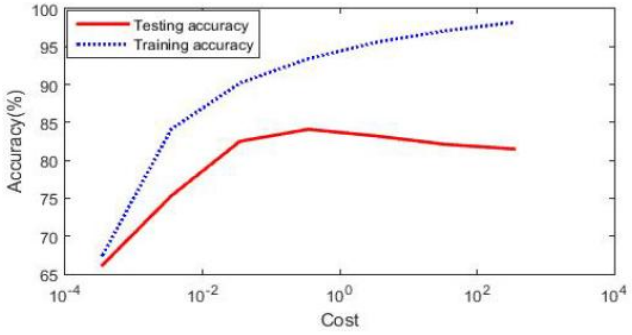

Fig. 3 Performance of STM under different cost

4. Experiment

In this section, we compares the performance of TSM with that of VSM. Linear Support Vector Machine (C-SVM), Naive Bayes and k-nearest neighbor (kNN) in vector space are adopted as the baselines. To make the result more convincing, we conduct three experiments using different testing set. In experiment 1, the labeled samples are randomly splitted into two parts: 80% for training and 20% for testing. In experiment 2, to avoid the noise resulted from the ambiguous definition of post-80s and post-90s, we find 175 users whose age information are publicly available in our corpus to construct a new testing set.

4.1. *Experiment on Current Data Set*

As mentioned in section 3, our data set contains 2127 labeled users (1380 were born in 1980s and 747 were born in 1990s). The dimension of distributed representation model is 40, and the number of keywords for each user is 100. The rank and cost of support tensor machine (STM) is set to `6' and `0.35'. In baselines, we apply x^2 test to select the most correlated 20000 features. Then, in

C-SVM, we set the cost to `64', and use the default setting in libsvm for other parameters. In kNN, the number of neighbors is set to `5'. All the experiments are conducted on a computer with Inter(R) Core(TM) i5-2500 3.30GHZ cpu and 8GB RAM memory.

Tab. 1 Accuracy (%) and standard deviation of tensor method and vector methods

	STM	C-SVM	Niave Bayes	kNN
Training	93.2(1.7)	97.5(0.9)	90.2(1.9)	83.1(2.8)
Testing	82.2(3.7)	78.1(4.1)	78.2(4.4)	69.0(6.2)

Tab. 2 Training time of one experiment

	STM	C-SVM	Niave Bayes	kNN
Training time (s)	172	274	102	88

The averaged result of 50 random experiments is given in table 1. The training time for one experiment is given in Tab. 2. The sensitivity of the cost of STM is analyzed in Fig. 3. With similar training time, the STM obviously outperforms the vector-based methods.

4.2. *Experiment on Examined Users*

In experiment 1, most of the labels in our data set are accurate, but noise still exists because of the ambiguous boundary between post-80s and post-90s. Some Chinese argue that post-80s are people born in 1980-1989, while others insist that post-80s should be people born in 1981-1990, which leads to some errors in the labels. In order to obtain a testing set without noise, we artificially check the personal information in Sina (which cannot be obtained from open APIs) of 175 users in our corpus to know their actual age.
In experiment 2, a new testing set with examined labels contains 175 users is constructed, with the exact definition that post-80s are people born in 1980-1989. In addition, the training set in this experiment is the combination of training set and testing set in experiment 1. The parameters are set to the same settings of experiment 1. Table 3 shows the result of proposed method and baselines.

Tab. 3 Testing Accuracy (%) on examined testing set

	STM	C-SVM	Niave Bayes	kNN
Accuracy	77.1	73.7	73.7	68.5

5. Conclusion

In this paper, we study the problem of age prediction in Sina weibo and our ma-jor contribution lies in the text representation model. To utilize the similarity and relevance information between words and reduce the dimensionality in text classification, a wording embedding based tensor space model is proposed. Moreover, support tensor machine is adopted as the classifier in tensor space. In experiments, performance of the proposed method outperforms the baselines on different data sets.

The key findings in our study include: 1) Since the tensor learning method can effectively decrease the number of free parameters, the proposed method works well in applications with limited training data. 2) Performance of the proposed method largely relies on the comprehensiveness of distributed representation model, which has been shown in experiment 3.

References

1. 1. S. Bergsma, M. Dredze, B. Van Durme, T. Wilson, and D. Yarowsky, \Broadly im-proving user classiffcation via communication-based name and location clustering on twitter." in HLT-NAACL, 2013, pp. 1010{1019.
2. 2. D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, \Classifying latent user attributes.
3. in twitter," in Proceedings of the 2nd international workshop on Search and mining.
4. user-generated contents. ACM, 2010, pp. 37{44.
5. 3. K. Filippova, \User demographics and language in an implicit social network," in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Compu-tational Linguistics, 2012, pp. 1478{1488.
6. 4. D.-P. Nguyen, R. Gravel, R. Trieschnigg, and T. Meder, \" how old do you think I am?" a study of language and age in twitter," 2013.
7. 5. Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie, \You are where you go: Inferring demographic attributes from location check-ins," in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. ACM, 2015, pp. 295{304.
8. 6. M. Conover, J. Ratkiewicz, M. R. Francisco, B. Gon_calves, F.

Menczer, and A. Flam-mini, \Political polarization on twitter." ICWSM, vol. 133, pp. 89{96, 2011.

9. 7. G. Salton, A. Wong, and C.-S. Yang, \A vector space model for automatic indexing," Communications of the ACM, vol. 18, no. 11, pp. 613{620, 1975.
10. 8. P. D. Turney, P. Pantel et al., \From frequency to meaning: Vector space models of semantics," Journal of arti_cial intelligence research, vol. 37, no. 1, pp. 141{188, 2010.
11. 9. T. W. Yan and H. Garcia-Molina, \Index structures for information _ltering under the vector space model," in Data Engineering, 1994. Proceedings. 10th International Conference. IEEE, 1994, pp. 337{347.
12. 10. T. Mikolov, K. Chen, G. Corrado, and J. Dean, \E_cient estimation of word repre-sentations in vector space," arXiv preprint arXiv:1301.3781, 2013.
13. 11. T. Luong, R. Socher, and C. D. Manning, \Better word representations with recursive neural networks for morphology." in CoNLL. Citeseer, 2013, pp. 104{113.
14. 12. M. Faruqui and C. Dyer, \Improving vector space word representations using multi-lingual correlation." Association for Computational Linguistics, 2014.
15. 13. O. Levy, Y. Goldberg, and I. Ramat-Gan, \Linguistic regularities in sparse and explicit word representations." in CoNLL, 2014, pp. 171{180.
16. 14. H. Zhang, \Nlpir: Natural language processing and information re trieval sharing platform," 2014.
17. 15. Y. Goldberg and O. Levy, \word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," arXiv preprint arXiv:1402.3722, 2014.
18. 16. Z. Hao, L. He, B. Chen, and X. Yang, \A linear support higher-order tensor machine for classi_cation," Image Processing, IEEE Transactions on, vol. 22, no. 7, pp. 2911{2920, 2013.
19. 17. A. Aizawa, \An information-theoretic perspective of tf{idf measures," Information Processing & Management, vol. 39, no. 1, pp.

45{65, 2003.

20. 18. D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, \Supervised tensor learning," in Data Mining, Fifth IEEE International Conference on. IEEE, 2005, pp. 8{pp.