4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2015)

A text classification model constructed by Latent Dirichlet Allocation and Deep Learning

Yu Liu[1, a], Zhengping Jin [2, b]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[a]yuliu_careers@163.com, [b]zhpjin@bupt.edu.cn

Keywords: text classification, latent Dirichlet allocation, deep learning, Gibbs sampling

Abstract. In this paper, we proposed a mixed model of text classification constructed by latent dirichlet allocation and deep learning. The model present that a text will be represent as a vector computing by latent dirichlet allocation algorithm, and this vector is probabilistic vector of corresponding topic words space. Then we input these topic vectors into a deep learning framework for computing nonlinear relationship of each vector. Finally, we constructed a text classification system. The proposed model achieves a higher accuracy when compared with other current popular algorithms, such as SVM, KNN and TFIDF.

Introduction

Text classification plays an important role in many applications, such as document retrieval, web search, spam filtering and recommender system. However, machine learning algorithm is the heart of these applications such as Native Bayes or KNN. These algorithms typically face the text input to be represented as words vector. And the most common input vector representation for texts is the bag-of-words or bag-of-n-grams. However, the bag-of-words has many disadvantages. First, the bag-of-words vector is an unstructured collection and even some different documents have exactly the same word representation. Second, the bag-of-words is not considering the semantics of the words. LDA[1] have emerged as a powerful new technique for finding useful structure and sensing semantic of text. In below section, we describe a topic model for uncovering the underlying semantic structure of a document collection, and this model based on a hierarchical Bayesian analysis of the original texts. However, if only having topic vectors we do not complete the text classification task. And we need to explore the complex nonlinearity interdependencies of input vectors, but some traditional algorithms cannot work it such as KNN. Deep learning[2-6] has been applied to work related representations for words[7-8]. And it is good at complex nonlinear relationship representation[9-10]. So, we adopt deep learning algorithm to representation for text topics. In the next section, we construct a deep semantic learning framework.

In this paper, we propose a deep learning framework with multiple layer neural network structure for considering the problem of modeling text corpora and other collections of discrete data. First, we use LDA algorithm to product input vectors by preprocessing the text corpus. Unlike the bag-of-words, LDA provides a more semantic document topic representation. Then, we import input vectors into deep learning framework which can explore the complex nonlinearity interdependencies of input vectors. In addition, we use a topic vector to represent a text than bag-of-words, because it usually with fewer words and it can reduce the calculated quantity of the deep learning algorithm.

The plan of this article is as follows. In the next section, we will describe the details of the algorithm of the model. In the third section, the experiment results which indicate the proposed model outperform many popular algorithms in this field will present to readers. In the last section, we have a conclusion about this paper.

Deep semantic text classification model

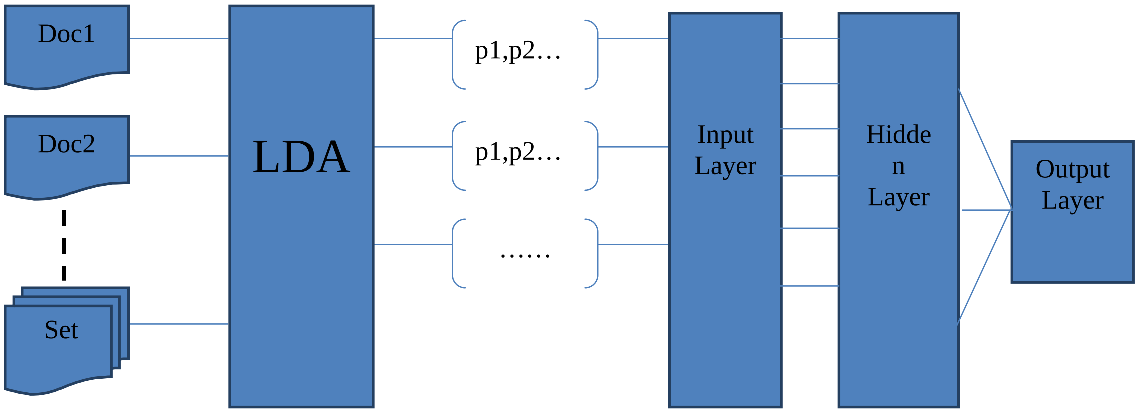

Fig. 1 Deep semantic text classification model

Fig. 1 shows the main structure of this text classification model. The first half part is LDA algorithm. LDA is an important topic model algorithm. LDA reduce a long text words into a short topic vector. The long text could be some news or fictions and so on. LDA makes the topics to represent text in a way suitable with content items on a meaning level. The latter half part is deep learning framework. Our proposed model explicitly input the topic vectors into specific neutral network to explore the complex nonlinearity interdependencies of topic vectors. Finally, we take the uniform training rule to train this classification model.

Topic probabilistic model. Each document $\vec{w}_m \in D$ is mapped to a K-dimensional topic vector $\vec{\vartheta}_m$, where the K is the topic amount. It is deriving from the below equation:

$$P(\vec{w}_m,\vec{z}_m,\vec{\vartheta}_m,\Phi\,|\,\vec{\alpha},\vec{\beta})=\prod_{n=1}^{N_m}P(w_{m,n}\,|\,\vec{\varphi}_{Z_{m,n}})\cdot P(Z_{m,n}\,|\,\vec{\vartheta}_m)\cdot P(\vec{\vartheta}_m\,|\,\vec{\alpha})\cdot P(\vec{\varphi}_{Z_{m,n}}\,|\,\vec{\beta}). \qquad (1)$$

Those $\vec{\alpha},\vec{\beta}$ are hyper-parameters for document-specific topic proportion and topic-specific term distribution respectively. The $w_{m,n}$ is stand for the n^{th} word of m^{th} document. The $Z_{m,n}$ is stand for the topic of the n^{th} word of m^{th} document. The $\vec{\varphi}$ is the topic-specific term distribution.

The $\vec{\vartheta}_m$ be show as $(p_1, p_2, \cdots, p_K)^T$, which p_i is m^{th} document sampling i-topic probability. Finally, every document's topic vector $\left\{\vec{\vartheta}_i, i=1\cdots M\right\}$ has been computed as input of deep learning framework.

Deep semantic neural network. Our deep neural network (DNN) framework has multiple hidden layers between the input and output layers. As mentioned above, the input layer is the topic vector $\left\{\vec{\vartheta}_i, i=1\cdots M\right\}$. And the higher layers enable composition of features from lower layers, giving the potential of modeling complex data with fewer units than a similarly performing shallow network.

The units of hidden layer have a full connection with the units input layer. And it transforms the features encoded in the topic vector into real-value number by nonlinearity function. The sigmoid function is used as nonlinearity function as following:

$$h(z)=\frac{1}{1+e^{-z}}. \qquad (2)$$

The nonlinearity function is also called activation function, and its output range is [0, 1]. Here z is the liner weighting function of input vector $\left\{\vec{\vartheta}_i, i=1\cdots M\right\}$ and the edge weight parameter W_K^M which connect the units between the input layer and hidden layer. The output layer is used softmax regression classier which is multiple classifications classier.

Training rule. For training LDA, we can use Gibbs sampling algorithm:

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t}. \tag{3}$$

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + a_k}. \tag{4}$$

That $n_k^{(t)}$ indicates word t is observed times allocating topic k, and $n_m^{(k)}$ indicates times which topic k is allocated to document m. The training algorithm will stop when parameters are convergence or iterations come to maximum value.

DNNs are typically feedforward networks. It can be trained with the standard backpropagation algorithm. The loss function is below:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h(\theta) - y)^2 + \lambda\|\theta\|_2^2. \tag{5}$$

That θ is training parameters, and $h(\theta)$ is the predicted function for text classification model. We also use standard L2 regularization of all the parameters, weighted by the hyper-parameter λ.

Experiments

Datasets. Our experiment data are from 20 newsgroups[11] and Reuters 21578[12]. 20 newsgroups data set consists of 20000 messages taken from 20 newsgroups, and Reuters 21578 dataset contains 21,578 documents. For all experiments, we will take ninety percentages of data for training task and the rest of data for testing task.

Results. As the result compared, we choose the Naive Bayes which is the simplest text classifier model and the Probabilistic Analysis of the Rocchio Algorithm with TFIDF[13] which also use the newsgroups data as the train data. Besides, SVM, KNN, and CART Decision Tree algorithm are considered as compared result. In the result, we take the accuracy as the evaluation index. Table 1 shows the result of the experiment, and the number stands for accuracy of each method. On the experiment data LDA+DNN performs significantly better than other algorithms.

Table 1. Experiment result compared

Method	ACC (20 newsgroups)	ACC (Reuters 21578)
LDA+DNN	**0.9150**	**0.9511**
Naive Bayes	0.7950	0.8341
PrTFIDF	0.9100	0.9500
SVM	0.9120	0.9510
KNN	0.9010	0.9415
CART	0.8125	0.8510

Conclusion

We have presented in this paper an efficient text classification model that combining topic model with deep learning. The advantage of LDA is that it represents text in a way suitable with content items on a meaning level and remedies the deep learning weakness of computing elapsed much time. The advantage of deep learning is that it explores the nonlinearity interdependencies of topic vectors and complements LDA topic model classification function. The experiment result shows that it outperforms current popular classification algorithms which contain SVM, KNN, TFIDF, CART Decision Tree and Naive Bayes.

Acknowledgements

This work is supported by NSFC (Grant Nos. 61300181, 61502044), the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

References

[1] Blei, D. M., Ng, A. Y., Jordan, M. I., & Lafferty, J. (2003). Latent dirichlet allocation. Journal of Machine Learning Research (Vol.3, pp.2003).

[2] Schölkopf, B., Platt, J., & Hofmann, T. (2007). Greedy layer-wise training of deep networks. In NIPS (Vol.19, pp.153-160).

[3] Ranzato, M., Poultney, C., Chopra, S., & Lecun, Y. (2006). Efficient learning of sparse representations with an energy-based model. Advances in Neural Information Processing Systems (NIPS 2006, 1137 - 1144.

[4] Deng, L., & Yu, D. (2013). Deep learning: methods and applications. Foundations & Trends® in Signal Processing, 7(3).

[5] Bengio, Y. (2009). Learning deep architectures for ai. Foundations & Trends in Machine Learning, 2, 1-55.

[6] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), 1527 - 1554.

[7] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin and Jean-Luc Gauvain. (2006). Neural probabilistic language models. Studies in Fuzziness & Soft Computing, 194, 137-186.

[8] Zhao, Y., Zhao, Y., Huang, S., Huang, S., Chen, H., & Chen, H., et al. (2014). An Investigation on Statistical Machine Translation with Neural Language Models. Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer International Publishing.

[9] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp.1701-1708). IEEE Computer Society.

[10] Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp.1653-1660). IEEE Computer Society.

[11] 20 newsgroup datasets https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

[12] Reuters 21578 datasets https://archive.ics.uci.edu/ml/datasets/Reuters-21578

[13] Thorsten Joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, Proceedings of the Fourteenth International Conference on Machine Learning, p.143-151, July 08-12, 1997