Educational and Non-educational Text Classification Based on Deep Gaussian Processes

Huijuan Wang*, Jing Zhao*$^{(\boxtimes)1}$, Zeheng Tang, and Shiliang Sun$^{(\boxtimes)2}$

Department of Computer Science and Technology, East China Normal University
3663 Zhongshan Road, Shanghai 200241, P. R. China
jzhao@cs.ecnu.edu.cn[1] slsun@cs.ecnu.edu.cn[2]

Abstract. With the development of the society, more and more people are concerned about education, such as preschool education, primary and secondary education and adult education. These people want to retrieve educational contents from large amount of information through the Internet. From the technical view, this requires identifying educational and non-educational data. This paper focuses on solving the educational and non-educational text classification problem based on deep Gaussian processes (DGPs). Before training the DGP, word2vec is adopted to construct the vector representation of text data. Then we use the DGP regression model to model the processed data. Experiments on real-world text data are conducted to demonstrate the feasibility of the DGP for the text classification problem. The promising results show the validity and superiority of the proposed method over other related methods, such as GP and Sparse GP.

Keywords: Deep Gaussian processes · Text classification · Word2vec · Machine learning

1 Introduction

Education has been a significant topic for a long time. It provides us with knowledge about the world, paves the way for a good career, leads to enlightenment and lays the foundation of a stronger nation. Various of work on educational data analysis has been done to offer some instructive guides for educational institutions or administrators [1–4]. Nowadays, lots of people are concerned about education, desiring to obtain education related information. While massive text data about education can be found on the Internet, most of them should be recognized from the large and diverse data which include educational data and non-educational data. To solve this problem, we focus on the educational and non-educational text classification task. Recently, many text classification methods have been proposed [5, 6]. Among them, the probabilistic models have the advantages of modeling the uncertainty and achieving competitive performance

* The authors contributed equally to this work.

with less data. So we resort to the Deep Gaussian Process (DGP), which is a deep extension of Gaussian processes (GPs).

GPs are one of the most famous probabilistic models and have a long history in the statistics community [7]. GPs are introduced into the machine learning domain as an effective Bayesian method for nonlinear regression and classification problems [8–10]. Under certain conditions, a GP can be seen as an MLP with infinite units in the latent layer.

Recently, deep learning has attracted sustained attention, which empirically seems to have structural advantages that can improve the quality of learning complicated data structures. Meanwhile, a kind of deep probabilistic model named DGP arises, which can include as many GP latent layers as possible, and thus is more powerful in data prediction and data structure analysis. In addition, various inference methods for the DGP have been developed [11–14]. Among them, the stochastic Expectation Propagation (EP) combined with the probabilistic back-propagation algorithm gives a computationally efficient, scalable and easy to implement algorithm [14]. Also, it is an algorithm designed for supervised learning tasks. Therefore, we adopt the DGP with stochastic EP approximation inference for our classification problem.

We can tell from the task that the text data are discrete, while the model we adopted needs continuous features, which requires changing discrete features into continuous features without losing much information. We adopt word2vec [15] to represent the words in our text data as continuous vectors. Some work has shown the effectiveness of using word2vec to construct the continuous vector representation. For example, word2vec was successfully used in the Sina Twitter data analysis [16], the Indonesian news articles analysis [17] and educational data analysis [18]. Therefore, the adoption of word2vec to represent text data is reasonable and will help to the classification task.

In this paper, we first use word2vec to represent the discrete words in continuous vectors on the basis of some necessary text processing. Then we apply the DGP to the resulting vectors. We record and analyze the experimental results by considering different factors such as inducing point number, latent layer number and training point number. We also analyze the characteristics of GPs and DGPs to find out their different applying scopes in the application level, which may give us good clues to the hyper-parameters setting. The experimental results show that DGPs work well in the text classification problem and do achieve a competitive accuracy with high efficiency.

2 Data Collection and Reconstruction

We collect the raw text data from the Internet and label them through human labor. We first conduct some conventional text processing like word stemmer, stop word removal and phases segments on the text data set. The data stored in the form of discrete words are not applicable for further processing when continuous data are required. The word2vec[1] is employed to represent the words in a continuous vector form.

Nowadays word2vec has been a useful tool in lots of natural language processing related work, such as clustering, looking for synonyms, part of speech analysis and so on. The core technology of word2vec is a word frequency Huffman coding and a three-layer neural network structure. By constructing a Huffman tree with the word frequency, the words are encoded in a distributed form, and the similarity of text semantics is encoded as the similarity in the K-dimensional vector space, which is convenient for the further training and predicting processes.

3 Model Introduction

After data processing, we get N data point pairs $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$. We will then train a model which can capture the essential characteristics of data set and generalize easily to a new point. DGPs can be seen as a deep extension of GPs by adding more latent layers. We will first introduce the GP and then present the DGP model as well as its stochastic EP inference.

3.1 Gaussian Processes

The GP supposes that any finite educational text data subset subjects to a Gaussian distribution [7]. It models the mapping from input to output as a certain Gaussian process. We suppose each data point y_n is generated from the corresponding latent function $f(\mathbf{x}_n)$ with an independent Gaussian noise, i.e.

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \epsilon \sim \mathcal{N}(\mathbf{0}, \delta_\epsilon^2 \mathbf{I}), \tag{1}$$

where $f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$ captures the dependency and characteristics of data. We adopt the automatic relevance determination (ARD) covariance kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 e^{-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{i,q} - x_{j,q})^2}$ with kernel parameters $\theta = \{\sigma_f^2, w_1, ...w_Q\}$ to construct the covariance matrix. In the Bayesian scenario, the parameters are learned by maximizing the marginal likelihood,

$$p(\mathbf{y}|\mathbf{X}) = \prod\nolimits_{n=1}^{N} \int p(y_n|f) p(f|\mathbf{x}_n) df = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{NN} + \delta_\epsilon^2 \mathbf{I}). \tag{2}$$

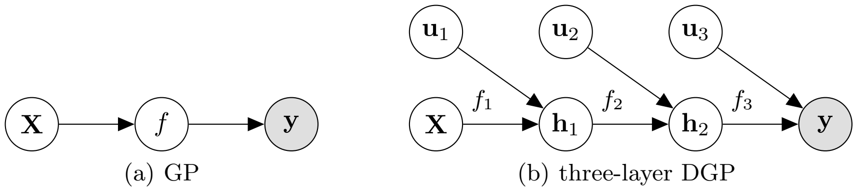
Then the prediction distribution of a new point $\mathbf{x}^*$ can be derived through conditional Gaussian distributions, as

$$\begin{aligned} p(f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*) &= \mathcal{N}(\mu^*, \mathbf{\Sigma}_f^*), \qquad (3)\\ \mu^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{X})^T (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \delta_n^2 \mathbf{I})^{-1} \mathbf{y},\\ \mathbf{\Sigma}_f^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \delta_n^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{X}, \mathbf{x}^*). \end{aligned}$$

The procedure of training costs $\mathcal{O}(N^3)$ time and can be reduced to $\mathcal{O}(NM^2)$ with M inducing points [19].

[1] Word2vec is an efficient tool for Google to represent the words as real value vectors. The python program can be achieved using the gensim toolkit.

3.2 Deep Gaussian Processes



(a) GP

(b) three-layer DGP

Fig. 1. (a) shows the GP model. (b) shows the three-layer DGP model, where $\{f_l\}_{l=1}^{3}$ are the GP mappings and $\{\mathbf{u}_i\}_{i=1}^{3}$ are the inducing points used for inference.

The DGP makes the GP deeper by adding more latent layers $\{\mathbf{h}_l\}_{l=1}^{L}$, each of which acts as the output of the above layer and the input of the next layer,

$$\begin{aligned} p(f_l|\Theta_l) &= \mathcal{GP}(f_l; \mathbf{0}, \mathbf{K}_l), l = 1, ..., L \qquad (4) \\ p(\mathbf{h}_l|\mathbf{h}_{l-1}, f_l) &= \prod\nolimits_n \mathcal{N}(h_{l,n}; f_l(h_{l-1,n}), \epsilon_l^2), h_{1,n} = \mathbf{x}_n, h_{L,n} = y_n. \end{aligned}$$

To release the burden of computation, inducing outputs $\mathbf{u}_l$ of input locations $\mathbf{z}_l$[2] in lth latent layer are introduced, as Figure 3.2 (b) shows. However, the model evidence is intractable since the latent variable is within the non-linear GP kernel mapping even after introducing the inducing point. Thus, the stochastic approximate EP method was developed to approximate the evidence [20]. It uses the following energy function as the objective likelihood,

$$\log p(\mathbf{y}|\mathbf{\Theta}) \approx \mathcal{F}(\mathbf{\Theta}) = (1 - N)\phi(\theta) + N\phi(\theta^{\backslash 1}) - \phi(\theta_{prior}) + \sum\nolimits_{n=1}^{N} \log \mathcal{Z}_n, \quad (5)$$

where $\mathbf{\Theta}$ denotes all the model parameters. ϕ is the log normalizer of a Gaussian distribution. $\theta, \theta^{\backslash 1}$ and θ_{prior} are the natural parameters of the distribution $q(\mathbf{u}), q^{\backslash 1}(\mathbf{u})$ and $p(\mathbf{u})$, respectively. $\mathcal{Z}_n = \int p(y_n|\mathbf{u}, \mathbf{X}_n) q^{\backslash 1}(\mathbf{u}) d\mathbf{u}$[3] is a approximation of $p(\mathbf{u}|\mathbf{X}, \mathbf{y})$.

When calculate the energy function, the first three terms are easy to compute, while the last difficult term is approximated by propagating a Gaussian through the next layer and projecting the non-Gaussian part back to a moment-matching process before propagating it to the next layer for each layer [14]. The parameters of the model and the approximation distribution can be derived with this process. The prediction distribution for a new point $\mathbf{x}^*$ can be expressed as

$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) \simeq \int p(y^*|\mathbf{x}^*, \mathbf{u}) q(\mathbf{u}|\mathbf{X}, \mathbf{y}) d\mathbf{u}, \quad (6)$$

which is also intractable and can be similarly dealt with. Specifically, a single forward pass is performed, in which each layer takes in a Gaussian distribution

[2] $\mathbf{z}_l$ will be omitted in our paper to simplify the notation.

[3] The $q^{\backslash 1}(\mathbf{u})$ is the variational cavity distribution of $\mathbf{u}$.

over the input, incorporates the approximate posterior of the inducing outputs and approximates the output distribution by a Gaussian [14]. We then use the sign of y^* as the classification result.

3.3 Remarks on DGPs

Here we give brief explain for why we choose DGPs rather than the standard GPs. On one hand, the DGP can learn the structure by extending the latent layers automatically, which will learn more effective features and thus making the prediction more accurate and powerful. Thus, the DGP is more flexible and suitable for complex data, like text data set. On the other hand, the propagation and moment-matching process costs $\mathcal{O}(NLM^2)$ time complexity for all data points. The data independency in the last term of the objective makes the stochastic optimization possible, which decreases the computational complexity substantially to $\mathcal{O}(\frac{NLM^2}{|B|})$, where $|B|$ denotes the mini-batch size.

4 Experiments

We first use word2vec to preprocess the text data and then adopt the DGP to classify the processed text data. We compare the DGP with other related methods like Sparse Gaussian Process (SGP) [19] and standard Gaussian Process Regression (GPR) [7] to show the advantages of the DGP. Experiments about DGPs with different latent layer numbers and inducing point numbers are also conducted.

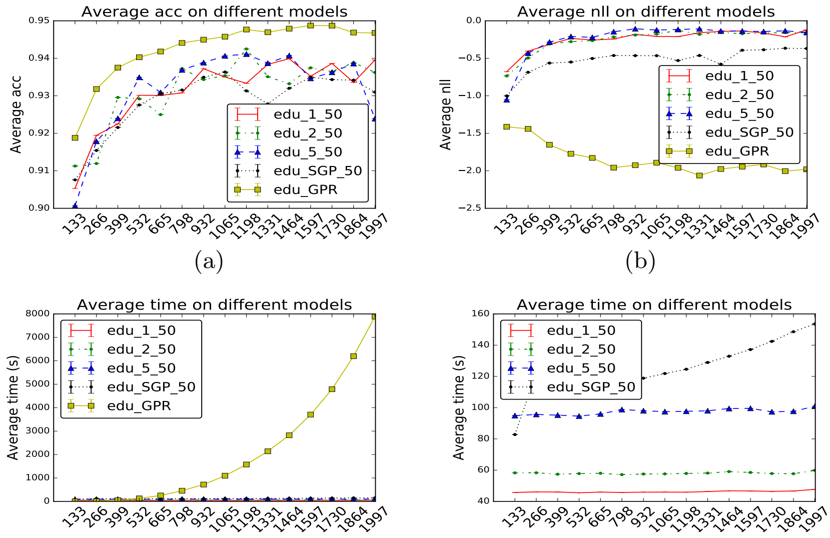
4.1 Experimental Setting

The used data have 2663 cases and for each case we represent it as a 50-dimension vector after using word2vec. In DGPs, the maximum iteration number is set to 1000. The mini-batches of the stochastic updates are set to 50 and the inducing point number per layer is set to 50, which is the same as the SGP. In our experiment, we pick up 5% more training points of the whole data set each time and compute the accuracy, the log likelihood and the training time on the rest points. We run the experiments for five times and record the average results.

4.2 Experimental Result and Analysis

The experimental results in terms of different criteria are exhibited in Figure 2, Figure 3 and Table 1.

Figure 2 (a) shows the average accuracies of different models over five experiments. Firstly, the ascending curves show the increasing accuracy with the increasing number of training data, which is in coincide with the fact that the model will be finer with more training data. Secondly, the GP gives the best result over other models for it is computed exactly while other methods use



(a)

(b)

(c)

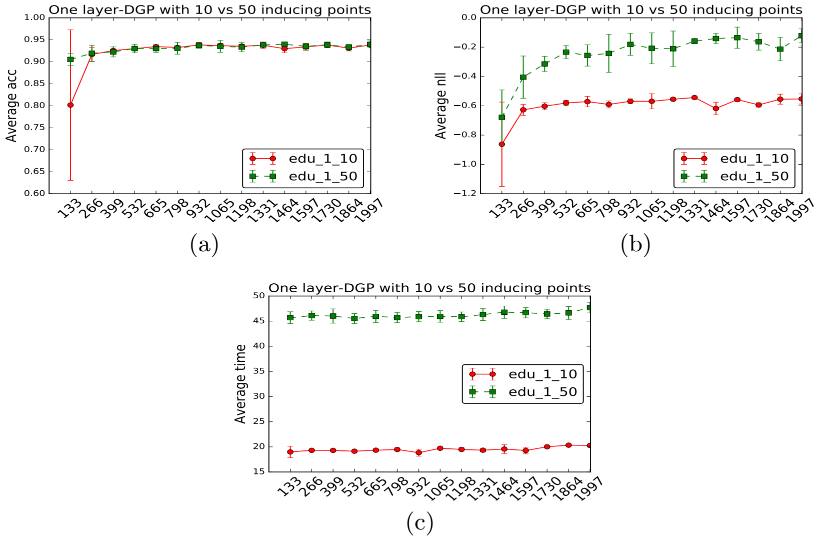
(d)

Fig. 2. The average results on education data set.

the approximation methods. DGPs are better than the SGP, which may be attributed to the deep structure and EP inference. At the same time, the accuracy of the DGP does not variate much with the increase of the latent layer number. In a whole, the higher accuracies of DGPs over the SGP show the effectiveness of the proposed method. Note that we remove the standard deviations in Figure 2 in order to make the curves clear, and list the average accuracies with standard deviations in Table 1. From Table 1, we can find that the variances of DGPs are slightly higher than the GP and SGP models, which may due to the adding up of the randomness of the model through using the stochastic optimization and EP process.

Figure 2 (b) shows the log likelihood results of different models. The higher likelihood of DGPs over the GP shows a better fitting result and a higher confidence, which is own to the deep structure of DGPs. We can also tell from the figure that the more layers the DGP has, the higher results the likelihood are. This fits the fact that the model will be more flexible with a deeper structure.

Figure 2 (c) and (d) show the training times of different models. From (c), the rapid climbing of the GP training time shows that training a standard GP makes considerable demands on time, which is a cubic of the training data size. The time reaches 8000s when there are about 2000 training points. This limits the use of GPs although it gives the best accuracy result. From (d) we can see that the training time of a SGP is more than that of a DGP with the same number of inducing points. This is attributed to the stochastic optimization of the DGPs. While for the DGPs, more latent layers will cost more training time, but the cost is linear with the layer number.



(a) (b) (c)

Fig. 3. The results with different inducing point numbers.

Additionally, Figure 3 shows the experimental results on different numbers of inducing points. The results show that training with more inducing points results in a similar accuracy, a better likelihood and more training time. The choice of the inducing point number is a tradeoff between the performance and time.

To sum up, DGPs get the most competitive classification results compared with the SGP and GP. The results show the advantages of DGPs over SGP both in accuracy and efficiency. The GP gets the best classification result at the cost of training time, while the DGPs get comparable results with much less training time. Also, the increase of the layer number in DGPs will help to improve the performance of the classification but with a little extra time.

Table 1. Average accuracy with a standard deviation of different models for different training set sizes.

Model\Points#	15%	30%	45%	60%	75%
edu_1_50	92.25±1.14	93.08±1.35	93.33±1.06	93.52±0.66	93.95±0.79
edu_2_50	92.96±0.87	93.72±0.46	94.25±0.36	93.75±0.86	93.62±0.79
edu_5_50	92.40±1.22	93.69±0.46	94.11±0.37	93.46±0.78	92.39±1.80
edu_SGP_50	92.16±1.09	93.15±0.50	93.13±0.70	93.50±0.18	93.10±0.98
edu_GPR	93.75±0.28	94.41±0.34	94.77±0.45	94.87±0.47	94.68±0.61

5 Conclusion and Future work

In this paper, we proposed an approach for educational and non-educational text data classification based on DGPs. We first process the text data into words, and then represent the discrete words as continuous vectors by word2vec. We apply the DGP to the processed data to perform educational and non-educational text data classification. In order to show the effectiveness of the proposed method, we conduct additional experiments on some related models as comparisons. From the experimental results, we conclude that the DGP is a reasonable and flexible method for text data classification. This is attributed to the advantages of deep structure and Bayesian characters of the DGP. In addition, the stochastic EP inference of DGPs makes it highly efficient. In future work, we will try to further classify the educational text data into sub-categories such as preschool education, primary and secondary education and adult education.

Acknowledgments

The first two authors Huijuan Wang and Jing Zhao are joint first authors. This work is sponsored by Shanghai Sailing Program. The corresponding author Shiliang Sun would also like to thank supports by NSFC Projects 61673179 and 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

1. T. Hsu. Research methods and data analysis procedures used by educational researchers. *International Journal of Research and Method in Education*, 28(2):109–133, 2005.
2. S. Sun. Computational education science and ten research directions. *Communications of the Chinese Association for Artificial Intelligence*, 9:15–16, 2015.
3. M. Yin, J. Zhao, and S. Sun. Key course selection for academic early warning based on Gaussian processes. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 240–247, 2016.
4. W. Limprasert and S. Kosolsombat. A case study of data analysis for educational management. In *International Joint Conference on Computer Science and Software Engineering*, pages 1–5, 2016.
5. X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657, 2015.
6. A. El-Halees. Arabic text classification using maximum entropy. *IUG Journal of Natural Studies*, 15(1):157–167, 2015.
7. C. Rasmussen. *Gaussian processes for machine learning.* Citeseer, 2006.
8. H. Nickisch and C. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(12):2035–2078, 2008.
9. H. Kim and Z. Ghahramani. Bayesian Gaussian process classification with the EM-EP algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1948–1959, 2006.

10. J. Zhao and S. Sun. Variational dependent multi-output Gaussian process dynamical systems. *Journal of Machine Learning Research*, 17:1–36, 2016.
11. N. Lawrence and A. Moore. Hierarchical Gaussian process latent variable models. In *International Conference on Machine Learning*, pages 481–488, 2007.
12. A. Damianou and N. Lawrence. Deep Gaussian processes. pages 207–215, 2013.
13. Z. Dai, A. Damianou, J. Gonzlez, and N. Lawrence. Variational auto-encoded deep Gaussian processes. *Computer Science*, 14(9):3942–3951, 2015.
14. T. Bui, D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.
15. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
16. X. Bai, F. Chen, and S. Zhan. A study on sentiment computing and classification of sina weibo with word2vec. In *International Congress on Big Data*, pages 358–363, 2014.
17. D. Rahmawati and M. Khodra. Word2vec semantic representation in multilabel classification for Indonesian news article. In *International Conference On Advanced Informatics: Concepts, Theory And Application*, pages 1–6, 2016.
18. J. Luo, S. Sorour, K. Goda, and T. Mine. Predicting student grade based on free-style comments using word2vec and ANN by considering prediction results obtained in consecutive lessons. *International Educational Data Mining Society*, pages 396–399, 2015.
19. E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1257–1264, 2006.
20. Y. Li, J. Hernández-Lobato, and R. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems*, pages 2323–2331, 2015.