Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1485–1491
Marseille, 11–16 May 2020

Evaluation of Deep Gaussian Processes for Text Classification

P. Jayashree, P. K. Srijith
Department of Computer Science and Engineering
IIT Hyderabad
{cs16resch11002, srijith}@iith.ac.in

Abstract

With the tremendous success of deep learning models on computer vision tasks, there are various emerging works on the Natural Language Processing (NLP) task of Text Classification using parametric models. However, it constrains the expressability limit of the function and demands enormous empirical efforts to come up with a robust model architecture. Also, the huge parameters involved in the model causes over-fitting when dealing with small datasets. Deep Gaussian Processes (DGP) offer a Bayesian non-parametric modelling framework with strong function compositionality, and helps in overcoming these limitations. In this paper, we propose DGP models for the task of Text Classification and an empirical comparison of the performance of shallow and Deep Gaussian Process models is made. Extensive experimentation is performed on the benchmark Text Classification datasets such as TREC (Text REtrieval Conference), SST (Stanford Sentiment Treebank), MR (Movie Reviews), R8 (Reuters-8), which demonstrate the effectiveness of DGP models.
Keywords: Text Classification, Bayesian deep learning, Gaussian Process, Convolutional Gaussian Process.

1. Introduction and Related Work

Text classification is a primary task in NLP which helps in solving multifold problems such as sentiment analysis (Le and Mikolov, 2014, Socher et al., 2013), spam detection (Wang, 2010) etc. Recently, deep learning models have obtained significant performance in computer vision. Inspired by which, there are some works on Deep learning models for NLP tasks (Kim, 2014, Yih et al., 2014, Shen et al., 2014, Kalchbrenner et al., 2014). However, deep learning models involve millions of learnable parameters, requires large dataset, does not make any uncertainty estimates, and incurs a tedious model selection procedure.

Gaussian Processes (GPs) offers a Bayesian non-parametric alternative framework to the existing parametric models. (Neal, 1994) discusses the equivalence between a bayesian neural network (with single hidden layer having infinite hidden units) and gaussian process. As the number of units in the hidden layer tend to infinity, the network converges to a gaussian process. The stacking of GPs lead to a Deep Gaussian Process (DGP) model (Damianou and Lawrence, 2013a). That is, a model where the observations are modelled as the output of the GP whose input is facilitated by another GP is DGP and it helps to overcome the aforementioned limitations.

There are many variants of DGP models with emphasis on the inferencing techniques used. (Damianou and Lawrence, 2013a) discusses on mean field variational posterior over hidden layers with factorised form as in (Titsias and Lawrence, 2010) and (Hensman and Lawrence, 2014) presents nested variational inference approach . (Dai et al., 2016) discusses on amortized inference whereas (Bui et al., 2016) uses approximate Expectation Propagation approach. (Salimbeni and Deisenroth, 2017) extends (Damianou and Lawrence, 2013a) with variational posterior conditioned on the previous layer, thereby facilitating parallelization of variational posterior computation as mini-batches.

(Kumar et al., 2018a, Kumar et al., 2018b) introduces the convolutional kernel (van der Wilk et al., 2017) into DGP framework for the task of image classification on benchmark datasets.

For the text classification task, the input sentence is represented as a matrix with rows representing the words and columns their embeddings. Given the sentence embedding matrix, the traditional RBF kernel fails to capture the similarity of non-contiguous words but captures only location-wise similarity. Alternatively, convolutional kernel can be evaluated on the sentence embedding where the patch size is decided based on the ngram feature similarity to be learnt. It helps to capture the semantic similarity of texts along with the syntactic similarity. In this paper, we propose DGP models for solving the task of Text Classification with various types of kernels such as convolutional kernel and weighted convolutional kernel. Weighted convolutional kernel is an extended version of convolutional kernel which facilitates weighting of sentence patches to capture semantic text similarity. Several experiments are carried out using various shallow and Deep Gaussian Process models evaluated on various benchmark Text Classification datasets such as TREC (Text REtrieval Conference), SST1 (Stanford Sentiment Treebank), MR (Movie Reviews), R8 (Reuters-8), which demonstrate the effectiveness of DGP models.

The rest of the paper is organized as follows: Section 2 introduces the notations used in the paper. Section 3 presents a background on Gaussian Process (GP) and Deep Gaussian Process (DGP) models. Section 4 elaborates on the Convolutional Deep Gaussian Process (CDGP) model for Text Classification. Section 5 discusses about the experimentation of various DGP models and analysis of results and Section 6 concludes with future research directions.

2. Preliminaries

We consider the text classification task with K classes and N training instances, $X = \{\mathbf{x_i}\}_{\mathbf{i=1}}^{\mathbf{N}}$ and the corre-

sponding labels $\mathbf{y} = \{\mathbf{y_i}\}_{\mathbf{i=1}}^{\mathbf{N}}$, where $\mathbf{x_i} \in \mathcal{R}^{\mathbf{H}\times\mathbf{W}}$ and $y_i \in \mathcal{Y} = \{1, 2, \ldots K\}$. Each $\mathbf{x_i}$ denotes a sentence, where each row h represents a word, H represents a sentence and W represents the word embedding dimension. Let f be a latent function $f : \mathcal{R}^{H\times W} \rightarrow \mathcal{Y}$ mapping the training sentences to output classes. Gaussian processes offers a Bayesian non-parametric approach to perform the task of text classification.

3. Background

3.1. Gaussian Process (GP)

A GP is defined as a collection of random variables such that any finite subset of which is Gaussian distributed (Rasmussen and Williams, 2005). A prior distribution of real valued functions f is given by a Gaussian Process (GP), denoted as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ where $m(\mathbf{x})$ represents the mean function and $k(\mathbf{x}, \mathbf{x}')$ represents the covariance of the function values at the data points $\mathbf{x}$ and $\mathbf{x}'$. A widely used traditional kernel function is the radial basis function (RBF) (squared exponential kernel), primarily used for modelling any smooth function. It is expressed as $\sigma_f^2 \exp(-\frac{1}{2\delta}||\mathbf{x} - \mathbf{x}'||^2)$ where the length scale δ captures the smoothness in function values across the inputs. Thereby, the choice of kernel function helps to determine various functional properties such as stationarity, smoothness etc.

3.2. Deep Gaussian Process (DGP)

Recent success in deep learning relies heavily on the representational power captured by stacking of layers in a Deep neural network. Similarly, (DGPs) (Damianou and Lawrence, 2013b, Damianou, 2015, Salimbeni and Deisenroth, 2017) stack GP layers resulting in a deep GP architecture, thereby learning rich representational functions along with uncertainty estimates. DGPs learn the function mapping of the input sentences to the output classes using composition of functions represented as $f(\mathbf{x}) = f^L \circ (f^{L-1} \ldots \circ (f^1(\mathbf{x})))$, for given L layers. The l^{th} layer primarily comprises of D^l functions $f^l = \{f_j^l\}_{j=1}^{D^l}$ which maps the representations obtained from the previous layer $l-1$ to obtain D^l representations for layer l. For every j^{th} representation and every layer l, independent GP priors are used. For example, the function f_j^l has prior as $f_j^l(\cdot) \sim \mathcal{GP}(m_j^l(\cdot), k^l(\cdot,\cdot))$. The j^{th} function in layer 1, f_j^1 is evaluated on the input data point $\mathbf{x}_i$ to obtain $F_{i,j}^1 = f_j^1(\mathbf{x}_i)$. Generalizing this to build the hierarchy of layers, the j^{th} function of layer l, $f_j^l(\cdot)$ is evaluated on the data representation $\mathbf{x}_i$ obtained from the previous layer $l-1$, F_i^{l-1}, thereby outputting the representation $F_{i,j}^l = f_j^l(F_i^{l-1})$. Let $\mathbf{f}_j^l$ denote the j^{th} representation of layer l computed across all inputs. The final layer L will have K functions with respect to all the output classes and squashing of these functions values using a softmax function is done to obtain the final class probabilities.

The kernel hyper-parameters are learnt by maximizing the evidence $p(y) = \int \prod_{n=1}^{N} p(y_n/F_n)p(F)dF$. However, computing the evidence is analytically intractable. Therefore, the variational parameters $\{\mathbf{m}^l, S^l\}_{l=1}^{L}$ corresponding to the posterior distribution and the kernel hyper-parameters are learnt by maximizing the variational Evidence Lower BOund (ELBO) which is shown in the Equation 1. For faster computation of the inversion of kernel covariance matrix K_{XX} among all the data points X in the lower bound, variational sparse gaussian process approximation technique (Hensman et al., 2013, Titsias, 2009) is employed. It takes $O(NM^2)$ time complexity where $M \ll N$ and M represents the number of inducing points and U^l represents the inducing variables across the dimensions D^l in the layer l.

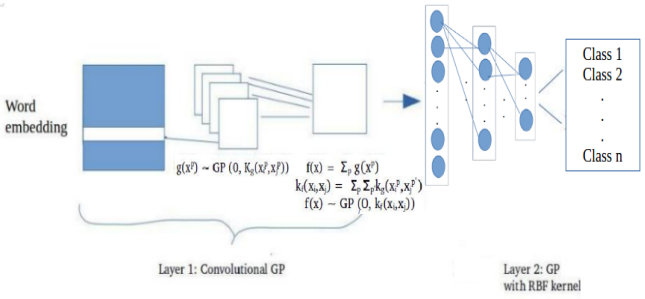

Figure 1: CDGP model for Text Classification

$$L(\{\mathbf{m}^l, S^l\}_{l=1}^{L}) = \sum_{n=1}^{N} \mathbb{E}_{q(F_n^L)}[log\ p(y_n|F_n^L)] - \sum_{l=1}^{L} KL[q(U^l)||p(U^l)] \quad (1)$$

Details on the variational lower bound, "reparameterization trick", and doubly stochastic variational inference technique are discussed in (Salimbeni and Deisenroth, 2017).

4. Convolutional Deep Gaussian Process (CDGP) model for Text Classification

In recent literature, convolutional kernel was employed in the covariance function of GPs and DGPs (van der Wilk et al., 2017, Kumar et al., 2018a, Kumar et al., 2018b) and it performed well for object recognition tasks and image classification task (Kumar et al., 2018a, Kumar et al., 2018b). For the NLP task of Text Classification, both syntactic and semantic similarity need to be captured in order to obtain a better generalization performance. And the convolutional kernel in the CDGP framework aids to better capture the semantic and syntactic text similarity.

Figure 1 shows the Convolutional Deep Gaussian Process (CDGP) model for Text Classification task. The kernel similarity between the sentences is obtained by the summation of the base kernel across various patches of the sentence. CGP performs the function evaluation on the input text data as sum of functions over the patches of the text input. Let P denote the number of patches in the input text $\mathbf{x}$ with each text patch $\mathbf{x}^{[p]}$ having $h \times w$ dimensions where h represents the number of words in the patch and w represents the embedding size. CGP is given as $f(\mathbf{x}) = \sum_{p=1}^{P} g(\mathbf{x}^{[p]})$ with $\mathcal{GP}$ prior over the function $g(\mathbf{x}^{[p]})$ is represented as

$g(\mathbf{x}^{[p]}) \sim \mathcal{GP}(0, k_g(\mathbf{x}_i^{[p]}, \mathbf{x}_j^{[p]}))$which produces a $\mathcal{GP}$ prior on the function $f(\mathbf{x})$ with zero mean and a convolutional kernel (Conv kernel) k_f given as,

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k_f(\mathbf{x}_i, \mathbf{x}_j)),$$
$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{P} \sum_{p'=1}^{P} k_g(\mathbf{x}_i^{[p]}, \mathbf{x}_j^{[p']}). \tag{2}$$

k_g is called as the base kernel. Exploiting a convolutional kernel in computing the sentence similarity helps to capture non-local similarities between the sentences. That is, the convolutional kernel compares an n-gram region in a sentence $\mathbf{x}_i$ with n-gram regions of the other sentence $\mathbf{x}_j$, and could result in a high similarity even when sentences are syntactically different but semantically same. Whereas with conventional RBF kernel, only respective location-wise similarity across sentences can be computed.

Convolutional DGP uses multiple functions from a GP prior with convolutional kernels to obtain a representation of the text in the first layer. The function corresponding to 0^{th} representation for layer 1 is obtained as

$$f_0^1(\mathbf{x}) = \sum_{p=1}^{P} g_0^1(\mathbf{x}^{[p]});$$
$$g_0^1(\mathbf{x}^{[p]}) \sim \mathcal{GP}(m_0^1(\mathbf{x}^{[p]}), k_g^1(\mathbf{x}_i^{[p]}, \mathbf{x}_j^{[p]}))$$
$$f_0^1(\mathbf{x}) \sim \mathcal{GP}(m_0^1(\mathbf{x}), k_f^1(\mathbf{x}_i, \mathbf{x}_j));$$
$$k_f^1(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{P} \sum_{p'=1}^{P} k_g^1(\mathbf{x}_i^{[p]}, \mathbf{x}_j^{[p']}). \tag{3}$$

Each output of layer 1 is meant to capture specific representational features of the text. These feature representations of the text obtained in the initial layer are then subsequently mapped by making use of a GP with RBF kernel to obtain complex representations. In general, the 0^{th} function representation of layer l is represented as

$$f_0^l(F^{l-1}) \sim \mathcal{GP}(m_0^l(F^{l-1}), k_f^l(F_i^{l-1}, F_j^{l-1}))$$
$$k_f^l(F_i^{l-1}, F_j^{l-1}) = \sum_{p=1}^{P} \sum_{p'=1}^{P} k_g^l(F_i^{l-1[p]}, F_j^{l-1[p']}). \tag{4}$$

All kernel matrices $K^l_{F^{l-1}F^{l-1}}$, $K^l_{F^{l-1}Z^l}$ and $K^l_{Z^lZ^l}$ used in the conditional distribution computation (Kumar et al., 2018a, Kumar et al., 2018b) use the convolutional kernel as defined in the Equation 4. Z^l denotes the inducing points in layer l with the same dimension as that of F^{l-1}.

A variant of the convolutional kernel such as weighted convolutional kernel (Wconv kernel) (van der Wilk et al., 2017) is also considered. It associates a differential weightage to each patch which facilitates better generalization. The function $f(\mathbf{x})$ for any layer l is generally written as

$$f(\mathbf{x}) = \sum_{p=1}^{P} w_p g(\mathbf{x}^{[p]});$$
$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{P} \sum_{p'=1}^{P} w_p w_{p'} k_g(\mathbf{x}_i^{[p]}, \mathbf{x}_j^{[p']}). \tag{5}$$

Table 1: Data Statistics for R8 Dataset

R8 Data Statistics			
Class name	**#Train Docs.**	**#Test Docs.**	**#Total Docs.**
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	**5485**	**2189**	**7674**

5. Experimentation and Results

5.1. Datasets used for experimentation

We validate variants of Gaussian Process models both shallow and deep models on the following benchmark datasets:

1. **TREC 1:** Text REtrieval Conference Question Classification dataset (Li and Roth, 2002) with 1000 labelled train samples and 500 test samples.

2. **TREC 5:** Text REtrieval Conference Question Classification dataset (Li and Roth, 2002) consists of 5452 train and 500 test instances, and 6 labels representing the question types.

3. **MR:** Movie Reviews dataset (Pang and Lee, 2005) is a binary classification dataset with positive and negative labels, and 10662 instances in total.

4. **SST1:** Stanford Sentiment Treebank dataset (Socher et al., 2013) contains 10,754 instances.The train split is 8544 and test split is 2210 and the corresponding multi-class labels represent sentiments as very negative, negative, neutral, positive, and very positive.

5. **R8:** It is a subset of Reuters-21578 (Lewis, 1992), a commonly used text categorization dataset. R8 (Debole and Sebastiani, 2005, Cardoso-Cachopo, 2007) contains the top 8 classes with frequent number of data samples. It is a highly skewed dataset and its data statistics are presented in Table 1. This dataset is chosen to primarily analyze the behaviour of gaussian process models on skewed datasets.

5.2. Pre-processing

For the generation of word vectors for all the datasets aforementioned, we used the pre-trained *word2vec* vectors trained using 100 billion words from Google news. The word vectors are of 300 dimensions and trained by using C-BOW (Mikolov et al., 2013). The words which are not in pre-trained list are randomly initialized. For uniformity across all sentences, the row length for every input embedding is fixed to be the maximum length among all given sentences.

Table 2: Hyper-parameter values after training the CDGP models

Dataset	Layer 1 (conv/wconv)		Layer 2 (RBF)	
	δ	σ	δ	σ
TREC1	9.95	2.10	10.21	2.13
TREC5	10.16	3.06	11.19	3.90
MR	8.49	11.01	9.31	0.37
SST1	9.24	22.13	9.99	7.68
R8	24.30	11.15	35.00	32.13

5.3. Experimental Details

For our experiments, SGP denotes a shallow GP with single layer. DGP denotes a deep GP model with multiple $(2/3)$ layers. The inducing points are taken as the centroids obtained by clustering the training data samples. The kernels used for evaluation are RBF, convolutional, and weighted convolutional kernels. RBF kernel is used as the base kernel in the convolutional/weighted convolutional kernels for CDGP models. The models were experimented with various initializations of kernel hyper-parameters such as lengthscales and variance in the scale of 0.1 to 20. The learnt hyper-parameter (lengthscale δ, variance σ) values for each layer of the best performing CDGP models for all the datasets are shown in the Table 2. The evaluation results of various Gaussian Process (GP) models are reported in the Table 3. The bracketed terms next to the model type, represents the kernel used in each layer separated by a $'+'$ sign respectively. For CDGP models, experiments were performed on varying patch sizes $h * w$ where h denotes the ngram filter size and w denotes the word embedding size i.e., Empirical evaluation on various filter sizes (such as $2, 3$) were done and the best results are reported in the Table 3. The number of hidden units in the second/third layers were varied in the scale of 10 to 50. The number of outputs in the final layer is determined by the number of classes of each dataset. The parameters of the model are learnt by ADAM optimizer with a step size ranging from 0.01 to 0.1 and varying mini-batch sizes in the scale of 100 to 1000. The number of epochs taken for training for TREC1 dataset is 200, TREC5 dataset ranges from 200 to 1000, SST1 dataset ranges from 1600 to 2500, MR dataset ranges from 3800 to 4200, and R8 dataset ranges from 1600 to 5500. The performance metrics used for comparison are Accuracy, and Negative Log-likelihood Predictive Probability (NLPP) to account for uncertainty in predictions.

Table 4 compares the text classification performance of Gaussian Process models with the existing Deep Learning models for TREC5, SST1, MR, and R8 datasets. For TREC5 dataset, CDGP 2 performs better than most of the models and almost similar to the (Cer et al., 2018) model. For SST1, MR, and R8 datasets, deep learning models such as (McCann et al., 2017, Zhao et al., 2015, Tellez et al., 2018) perform correspondingly better. Since SST1 is a fine-grained multi-class sentiment classification dataset, it remains a challenging task for all the models to achieve notable performance as in TREC5 dataset.

Table 3: Experimental results of various Gaussian Process models. Performance metrics: Accuracy (higher the better) and Negative Log-likelihood Predictive Probability (lower the better)

Dataset	Model	Acc	NLPP
TREC1	SGP (RBF)	94	0.25
	CDGP 1 (conv+RBF)	**94.80**	**0.21**
	CDGP 2 (wconv+RBF)	94.60	0.22
TREC5	SGP (RBF)	97.40	0.10
	CDGP 1 (conv+RBF)	97.40	0.15
	CDGP 2 (wconv+RBF)	**97.40**	**0.08**
	CDGP 3 (wconv+RBF+RBF)	97.20	0.10
SST1	SGP (RBF)	36.29	4.19
	CDGP 1 (conv+RBF)	35.02	3.84
	CDGP 2 (wconv+RBF)	**41.45**	**2.86**
MR	SGP (RBF)	70.95	0.60
	CDGP 1 (conv+RBF)	68.42	0.60
	CDGP 2 (wconv+RBF)	**77.13**	**0.49**
R8	SGP (RBF)	73.02	1.25
	CDGP 1 (conv+RBF)	79.23	1.59
	CDGP 2 (wconv+RBF)	**89.86**	**0.68**

Table 4: Comparison of Gaussian Process models with deep learning models. Performance metric: Accuracy (higher the better)

Model/Dataset	TREC5	SST1	MR	R8
	Accuracy			
CDGP 2 (wconv+RBF)	97.40	41.45	77.13	89.86
DCNN (Kalchbrenner et al., 2014)	93.0	48.5	-	-
CNN non-static (Kim, 2014)	93.6	48.0	81.5	-
TBCNN (Mou et al., 2015)	96.0	51.4	-	-
AdaSent (Zhao et al., 2015)	92.4	-	**83.1**	-
TWS (Escalante et al., 2015)	-	-	-	91.35
Multi-task (Liu et al., 2016)	-	49.6	-	-
DSCNN (Zhang et al., 2016)	95.6	50.6	82.2	-
BLSTM-2DCNN (Zhou et al., 2016)	96.1	52.4	82.3	-
CoVe (McCann et al., 2017)	95.8	**55.2**	-	-
USE_T+CNN (Cer et al., 2018)	**98.70**	-	82.70	-
TextEnt (Yamada et al., 2018)	-	-	-	96.7
μTC (Tellez et al., 2018)	-	-	-	**96.98**

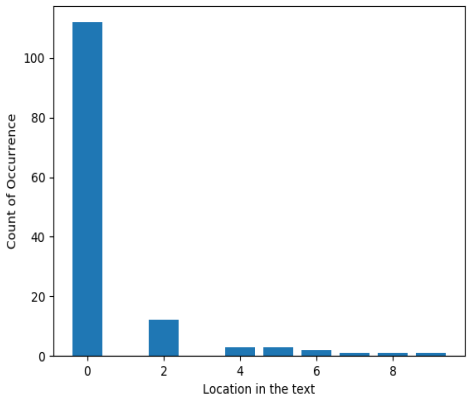

Figure 2: Histogram of location-wise word frequency for the word 'animal'

5.4. Empirical Observations

From Table 3, the following insights are drawn:

1. For small scale datasets such as TREC1, CDGP models are able to perform better than SGP models by utilizing the benefits of convolutional kernel to capture location independent similarity across texts.

2. For medium size dataset such as TREC5 which is a Question-Answering dataset, both SGP and CDGP models perform comparatively but CDGP 2 (wconv+RBF) model has better NLPP estimate. To analyze the reasoning behind this, the top keywords in the dataset are chosen and its count of occurrence in the data with respect to location in the sentence are plotted. It was found that words occur with higher frequency in the initial location slots as shown in the Figure 2 for the example word 'animal'. Hence, even SGP (RBF kernel [location-based]) is able to perform relatively.

3. For binary classification task of MR dataset, CDGP with weighted convolutional kernel better captures the sentiment associated with the text and hence achieves higher accuracies and better NLPP estimates when compared to other gaussian process models.

4. For fine-grained and multi-class sentiment classification task such as SST1, CDGP 2 (wconv+RBF) model achieves higher accuracy and better NLPP estimate when compared to SGP (RBF) and CDGP 1 (conv+RBF) models. This behaviour is caused by weighting of patches done by the weighted convolutional kernel in CDGP 2 model.

5. For a highly skewed dataset such as R8, CDGP 2 (wconv+RBF) achieves higher accuracy and better NLPP estimate than other gaussian process models. The CDGP 2 model captures the double-fold benefits of bayesian modelling and weighted convolutional kernel.

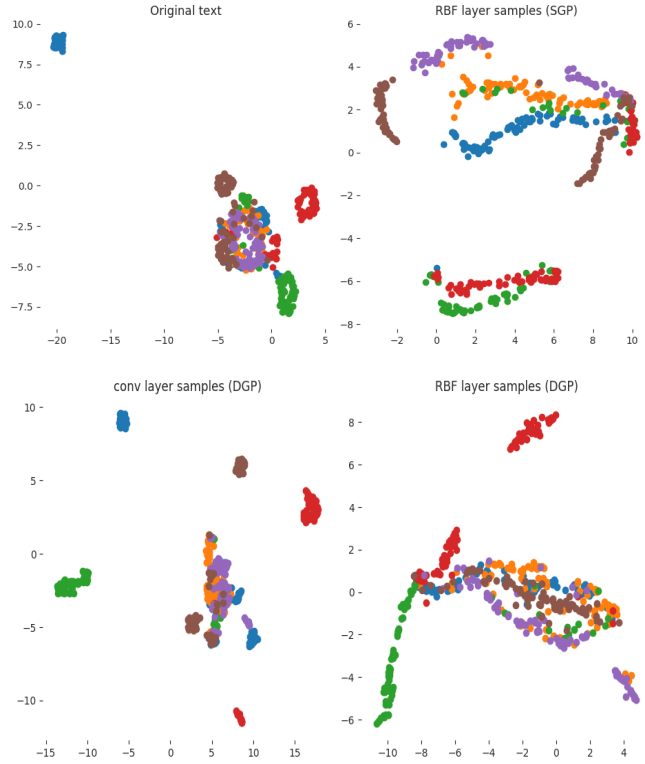

Figure 3: TREC5: original data, samples from SGP and CDGP models

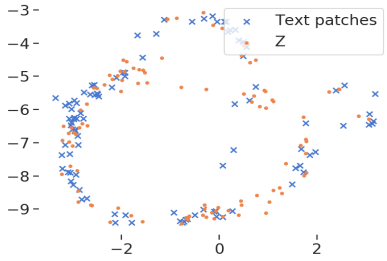

Figure 4: TREC5:inducing points of SGP (RBF) and original data samples

5.5. Samples Visualization

Figure 3 shows the UMAP embedding of the original TREC5 data points, mean sample from RBF layer of SGP model, and mean sample from the first layer with convolutional kernel and second layer with RBF kernel of CDGP model. Since TREC5 dataset has comparable results for SGP and CDGP, it is treated for such visualization. Convolutional kernel in the first layer of CDGP brings in a better class separability, and the RBF layer of SGP looks comparable to CDGP's RBF layer. Figure 4 shows the UMAP embedding of sample patches from original text embedding, and that of the inducing points which is widely spread across the original text.

6. Conclusion

In this paper, we have proposed CDGP models for the task of Text Classification. It is mainly motivated by the advantages of bayesian non-parametric models, such

as unconstrained expressability limit of the function, automated model selection, better generalization even for smaller datasets, and uncertainty estimates. An extensive empirical evaluation of various shallow and DGP models is performed on the benchmark text classification datasets, which demonstrate the performance of DGP models. As a future work, we would like to explore the benefits of uncertainty estimates given by CDGP for text classification in medical and legal domain applications.

7. Bibliographical References

Bui, T. D., Hernández-Lobato, J. M., Hernández-Lobato, D., Li, Y., and Turner, R. E. (2016). Deep gaussian processes for regression using approximate expectation propagation. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16, page 1472–1481. JMLR.org.

Cardoso-Cachopo, A. (2007). Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 169–174.

Dai, Z., Damianou, A., González, J., and Lawrence, N. (2016). Variational auto-encoded deep Gaussian processes. *International Conference on Learning Representations (ICLR)*.

Damianou, A. and Lawrence, N. (2013a). Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215.

Damianou, A. and Lawrence, N. (2013b). Deep Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 207–215.

Damianou, A. (2015). Deep Gaussian processes and variational propagation of uncertainty. *PhD Thesis, University of Sheffield*.

Debole, F. and Sebastiani, F. (2005). An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and Technology*, 56(6):584–596.

Escalante, H. J., García-Limón, M. A., Morales-Reyes, A., Graff, M., Montes-y Gómez, M., Morales, E. F., and Martínez-Carranza, J. (2015). Term-weighting learning via genetic programming for text classification. *Know.-Based Syst.*, 83(C):176–189.

Hensman, J. and Lawrence, N. D. (2014). Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370*.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 655–665.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Kumar, V., Singh, V., Srijith, P., and Damianou, A. (2018a). Deep gaussian processes with convolutional kernels. *arXiv preprint arXiv:1806.01655*.

Kumar, V., Singh, V., Srijith, P., and Damianou, A. (2018b). Deep gaussian processes with convolutional kernels. *UAI workshop on uncertainty in deep learning*.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, page 37–50. Association for Computing Machinery.

Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics (COLING)-Volume 1*, pages 1–7.

Liu, P., Qiu, X., and Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems (NIPS)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119.

Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L., and Jin, Z. (2015). Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2315–2325.

Neal, R. (1994). Priors for infinite networks (tech. rep. no. crg-tr-94-1). *University of Toronto*.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics (ACL)*, pages 115–124.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 4588–4599.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). Learning semantic representations using convo-

lutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP)*, pages 1631–1642.

Tellez, E. S., Moctezuma, D., Miranda-Jimnez, S., and Graff, M. (2018). An automated text categorization framework based on hyperparameter optimization. *Know.-Based Syst.*, 149(C):110–123.

Titsias, M. and Lawrence, N. D. (2010). Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 844–851.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574. PMLR.

van der Wilk, M., Rasmussen, C. E., and Hensman, J. (2017). Convolutional Gaussian processes. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 2849–2858.

Wang, A. H. (2010). Don't follow me: Spam detection in twitter. In *Security and cryptography (SECRYPT), proceedings of the 2010 international conference on*, pages 1–10. IEEE.

Yamada, I., Shindo, H., and Takefuji, Y. (2018). Representation learning of entities and documents from knowledge base descriptions. *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

Yih, W.-t., He, X., and Meek, C. (2014). Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 643–648.

Zhang, R., Lee, H., and Radev, D. R. (2016). Dependency sensitive convolutional neural networks for modeling sentences and documents. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (ACL): Human Language Technologies*, pages 1512–1521.

Zhao, H., Lu, Z., and Poupart, P. (2015). Self-adaptive hierarchical sentence model. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495.