SSN_MLRG1 at SemEval-2017 Task 5: Fine-Grained Sentiment Analysis Using Multiple Kernel Gaussian Process Regression Model

Angel Deborah S, S Milton Rajendram, T T Mirnalinee
SSN College of Engineering
Kalavakkam 603 110, India
angeldeboarahs@ssn.edu.in

Abstract

The system developed by the SSN_MLRG1 team for Semeval-2017 task 5 on fine-grained sentiment analysis uses Multiple Kernel Gaussian Process for identifying the optimistic and pessimistic sentiments associated with companies and stocks. Since the comments on the same companies and stocks may display different emotions depending on time, their properities like smoothness and periodicity may vary. Our experiments show that while single Kernel Gaussian Process can learn some properties well, Multiple Kernel Gaussian Process are effective in learning the presence of different properties.

1 Introduction

Sentiments have been widely studied as they play an important role in human intelligence, rational decision making, social interaction, perception, memory, learning and creativity (Pang and Lee, 2008; Strapparava and Mihalcea, 2008; Maas et al., 2011; Li et al., 2015). The ability to discern and understand human sentiments is critical for making interactive human-like computer agents, and requires the use of machine learning approaches (Alm et al., 2005).

2 Gaussian Process

Gaussian Process (GP) is a Bayesian non-parametric approach to machine learning. A Gaussian Process is a collection of random variables, any infinite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). Using a Gaussian process, we can define a distribution over functions $f(x)$,

$$f(x) \sim GP(m(x), k(x, x')) \quad (1)$$

where $m(x)$ is the mean function, usually defined to be zero, and $k(x, x')$ is the covariance function (or kernel function) that defines the prior properties of the functions considered for inference. Gaussian Process has the following main advantages (Cohn and Specia, 2013; Cohn et al., 2014).

- The kernel hyper-parameters can be learned via evidence maximization.
- GP provides full probabilistic prediction, and an estimate of uncertainty in the prediction.
- Compared to SVMs which need unbiased datasets for good performance, GPs do not usually suffer from this problem.
- GP can be easily extended and incorporated into a hierarchical Bayesian model.
- GP works really well when combined with kernel models.
- GP works well for small datasets too.

2.1 Gaussian Process Regression

The Gaussian Process regression framework assumes that, given an input x, output y is a noise corrupted version of a latent function evaluation. In a regression setting, we usually consider a Gaussian likelihood, which allows us to obtain a closed form solution for the test posterior (Ebden, 2008). Gaussian Process model, as they are applied in machine learning, is an attractive way of doing non-parametric Bayesian modeling for a supervised learning problem. GP-based modeling has the ability to learn hyper-parameters directly from data by maximizing the marginal likelihood. Like other kernel methods, the Gaussian Process can be optimized exactly, given the values of their hyper-parameters and this often allows a fine and precise trade-off between fitting the data and smoothing.

A practical implementation of Gaussian Process Regression (GPR) (Rasmussen and Williams, 2006) is outlined in the following algorithm:

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 823–826,
Vancouver, Canada, August 3 - 4, 2017.

Algorithm: Predictions and log-marginal likelihood for GP regression.
Input: X (training inputs) , $\mathbf{y}$ (training targets), k (covariance function), σ_n^2 (noise level), x_* (test input).
Output: Predictive mean, variance and log-marginal likelihood.

1. $L := \text{cholesky}(K + \sigma_n^2 I)$
2. $\alpha := L^T \backslash (L \backslash y)$
3. $\overline{f}_* := \mathbf{k}_*{}^T \alpha$
4. $\mathbf{v} := L \backslash \mathbf{k}_*$
5. $V[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$
6. $\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^T \alpha - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$
7. return f_* (mean), $V[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log-marginal likelihood)

2.2 Multiple Kernel Gaussian Process

The heart of every Gaussian process model is a covariance kernel. The kernel $\mathbf{k}$ directly specifies the covariance between every pair of input points in the dataset. The particular choice of covariance function determines the properties such as smoothness, length scales, and amplitude, drawn from the GP prior. Therefore, it is an important part of GP modelling to select an appropriate covariance function for a particular problem. Multi Kernel Learning (MKL) — using multiple kernels instead of a single one — can be useful in two ways:

- Different kernels correspond to different notions of similarity, and instead of trying to find which works best, a learning method does the picking for us, or may use a combination of them. Using a specific kernel may be a source of bias which is avoided by allowing the learner to choose from among a set of kernels.
- Different kernels may use inputs coming from different representations, possibly from different sources or modalities.

(Gonen and Alpaydin, 2011; Wilson and Adams, 2013) explain how multiple kernels definitely give a powerful performance. (Gonen and Alpaydin, 2011) also describes in detail various methodologies to combine kernels. (Wilson and Adams, 2013) introduces simple closed form kernels that can be used with Gaussian Processes to discover patterns and enable extrapolation. The kernels support a broad class of stationary covariances, but Gaussian Process inference remains simple and analytic.

We studied the possibility of using multiple kernels to explain the relation between the input data and the labels. While there is a body of work on using Multi Kernel Learning (MKL) on numerical data and images, yet applying MKL on text is still an exploration. We have used Exponential kernel and Multi-Layer Perceptron kernel together with Squared Exponential kernel, and found the combinations to give better results. The text data used in sentiment analysis is collected over a period of time. Comments on the same topic may exhibit different emotions, depending on the time it was made, and hence their properties, such as smoothness and periodicity, also vary with time. Since any one kernel learns only certain properties well, multiple kernels will be effective in detecting the presence of different emotions in the data.

The MKL algorithms use different learning methods for determining the kernel combination function. It is divided into five major categories: Fixed rules, Heuristic approaches, Optimization approaches, Bayesian approaches and Boosting approaches. The combination of kernels in different learning methods can be performed in one of the two basic ways, either using linear combination or using non-linear combination. Linear combination seems more promising (Gonen and Alpaydin, 2011), and have two basic categories: unweighted sum (i.e., using sum or mean of the kernels as the combined kernel) and weighted sum. Non-linear combination use non-linear functions of kernels, namely multiplication, power, and exponentiation. We have studied the fixed rule linear combination in this work which can be represented as

$$\mathbf{k}(x, x') = \mathbf{k_1}(x, x') + \mathbf{k_2}(x, x') + \ldots + \mathbf{k_n}(x, x'). \quad (2)$$

For training, we have used one-step method together with the simultaneous approach. One-step methods, in a single pass, calculate both the parameters of the combination function, and those of the combined base learner; and the simultaneous approach ensures that both sets of parameters are learned together.

3 System Overview

The system comprises of the following modules: data extraction, preprocessing, feature vector generation, and multi-kernel Gaussian Process model

building. The algorithm for preprocessing of the data and feature vector building is outlined below:
Algorithm: Preprocess the data and generate feature vectors.
Input: Input dataset.
Output: Dictionary with the key - value pair and BoW Feature vector.
begin

1. Perform lemmatization using `WordNet Lemmatizer` from the NLTK tool kit.
2. Perform tokenization using the `wordpunct` tokenize function of the NLTK toolkit.
3. Set the integer value for the `train` variable.
4. Build data dictionaries for training sentences.
5. Build a data dictionary with words mapped to their indices.
6. Generate feature vectors for the train sets that encode a BoW representation.
7. Build a dictionary with the key-value pairs. The key is the emotion and the value is a matrix where rows are BoW vectors.

end

The Multi-Kernel Gaussian Process (MKGP) model building is outlined in the following algorithm.
Algorithm: Build a Multi-Kernel Gaussian Process model.
Input: Input dataset with BoW feature representation.
Output: Learned model,
begin

1. Split the training dataset into XTrain which contains the features and YTrain that contains the emotion scores.
2. Build the initial regression model using appropriate kernel function.
3. Optimize the regression model with the hyper-parameters (length scale, variance, noise).
4. Return the learned model.

end
The Multi-Kernel Gaussian Process model is implemented using linear combination method which takes the unweighted sum of the kernels.

4 Comparison Using Different Kernels

The output submitted for the task was based on the linear combination of Squared Exponential kernel and Exponential kernel.

4.1 Kernels

The *Squared Exponential (SE)* kernel, sometimes called the Gaussian or Radial Basis Function (RBF), has become the default kernel in GPs. To model the long term smooth-rising trend we use a Squared Exponential covariance term.

$$\mathbf{k}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right). \quad (3)$$

where σ^2 is the variance and l is the length-scale.

The usage of *Exponential kernel* is particularly common in machine learning and hence is also used in GPs. They perform tasks such as statistical classification, regression analysis, and cluster analysis on data in an implicit space.

$$\mathbf{k}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')}{2l^2}\right) \quad (4)$$

The *Multi-Layer Perceptron* kernel has also found use in GP as it can learn the periodicity property present in the dataset; its $\mathbf{k}(x, x')$ is given by

$$\frac{2\sigma^2}{\pi} \sin^{-1} \frac{(\sigma_w^2 x^T x' + \sigma_b^2)}{\sqrt{\sigma_w^2 x^T x + \sigma_b^2 + 1}\sqrt{\sigma_w^2 x'^T x' \sigma_b^2 + 1}} \quad (5)$$

where σ^2 is the variance, σ_w^2 is the vector of the variances of the prior over input weights and σ_b^2 is the variance of the prior over bias parameters. The kernel can learn more effectively because of the additional parameters σ_w^2 and σ_b^2.

4.2 Performance Evaluation

Other combinations of the kernel were also tried after submission. One such kernel used for experimentation purpose was Multi-Layer Perception Kernel. The results of the Single Kernel and Multi-Kernel GP on subtask 1 dataset are collated in Table 1. The results of the Single Kernel and

Table 1: A performance comparison based on Cosine Similarity (CS), Pearson Score (PS) and Mean Absolute Error (MAE) for subtask 1 dataset

Model	CS	PS	MAE
SGP	0.6942	0.6694	0.2003
MKGP(R+E)	0.7044	0.6809	0.1965
MKGP(R+E+M)	0.7099	0.6864	0.1931
MKGP(R+M)	**0.7106**	**0.6872**	**0.1930**

Multi-Kernel GP on subtask 2 dataset are shown in Table 2. The kernel combinations used in Table 1 and Table 2 are

Table 2: A performance comparison based on Cosine Similarity (CS), Pearson Score (PS) and Mean Absolute Error (MAE) for subtask 2 dataset

Model	CS	PS	MAE
SGP	0.5590	0.5615	0.2506
MKGP(R+E)	0.5530	0.5569	0.2558
MKGP(R+E+M)	0.5864	0.5870	0.2445
MKGP(R+M)	**0.5931**	**0.5928**	**0.2426**

SGP: Single Kernel Gaussian Process with Radial Basis Function (RBF) kernel,

MKGP(R+E): Multi Kernel Gaussian Process with sum of RBF and Exponential kernels,

MKGP(R+E+M): Multi Kernel Gaussian Process with sum of RBF, Exponential, and Multi-Layer Perceptron kernels,

MKGP(R+M): Multi Kernel Gaussian Process with sum of RBF and Multi-Layer Perceptron kernels.

The evaluation considered 70% of the dataset for training and 30% for testing. The greater the Cosine Similarity (CS) and the Pearson Score (PS), and the smaller the Mean Absolute Error (MAE), the better the performance of the system. The tables show that MKGP(R+M), Multi Kernel Gaussian Process with sum of Squared Exponential and Multi-Layer Perceptron kernels, performs better.

5 Official Evaluation

The systems developed were evaluated based on Cosine Similarity measure. Our system ranked fifth position with Cosine Similarity of 0.7347 for subtask 1 and fifteenth position with Cosine Similarity of 0.6657 for subtask 2.

6 Conclusion

In this paper, we have presented a Multi Kernel Gaussian Process(MKGP) regression model for fine-grained sentiment analysis of financial microblogs and news. We used Bag of Words input feature vectors as input and fixed rule multi kernel learning to build GP model and found it to perform better than single kernel learning. The results can be further enhanced by using different feature generation approaches and multi kernel learning approaches.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)-2005*. ACL, pages 579–586.

Trevor Cohn, Daniel Beck, and Lucia Specia. 2014. Joint emotion analysis via multi-task gaussian processes. In *Proceedings of EMNLP 2014, The International Conference on Empirical Methods in Natural Language Processing*. Journal of Machine Learning Research, pages 1798–1803.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the ACL-2013*. ACL, pages 32–42.

M Ebden. 2008. Gaussian processes for regression: A quick introduction. *arXiv.org* .

Mehmet Gonen and Ethem Alpaydin. 2011. Multiple kernel learning algorithms. *Journal of Machine Learning Research* 24(11):2211–2268.

Shoushan Li, Lei Huang, Rong Wang, and Guodong Zhou. 2015. Sentence-level emotion classification with label and context dependence. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing*. ACL, pages 1045–1053.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *PHLT '11 Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*. ACL, pages 142–150.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1):1–135.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*, volume 1. MIT Press Cambridge, Englewood Cliffs, NJ.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *SAC '08 Proceedings of the 2008 ACM symposium on Applied computing*. pages 1556–1560.

A. G. Wilson and R. P. Adams. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of ICML 2013, The International Conference on Machine Learning*. Journal of Machine Learning Research, pages 1067–1075.