SSN_MLRG1 at SemEval-2017 Task 4: Sentiment Analysis in Twitter Using Multi-Kernel Gaussian Process Classifier

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

Abstract

The SSN_MLRG1 team for Semeval-2017 task 4 has applied Gaussian Process, with bag of words feature vectors and fixed rule multi-kernel learning, for sentiment analysis of tweets. Since tweets on the same topic, made at different times, may exhibit different emotions, their properties such as smoothness and periodicity also vary with time. Our experiments show that, compared to single kernel, multiple kernels are effective in learning the simultaneous presence of multiple properties.

1 Introduction

Twitter is a huge microblogging service with more than 500 million tweets per day from different locations of the world and in different languages (Nabil et al., 2016). The sentiment analysis in Twitter has been applied in various domains such as commerce (Jansen et al., 2009), disaster management (Verma et al., 2011) and health (Chew and Eysenbach, 2010). The task is challenging because of the informal writing style, the semantic diversity of content as well as the "unconventional" grammar. These challenges in building a classification model can be handled by using proper approaches to feature generation and machine learning.

The heart of every Gaussian process model is a covariance kernel. 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 Alpaydn, 2011) and (Wilson and Adams, 2013) explain how multiple kernels definitely give a powerful performance. (Gonen and Alpaydn, 2011) also describe 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.

2 Gaussian Process

Gaussian Process is a non-parametric Bayesian modelling in supervised setting. Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). Using a Gaussian process, we can define a distribution over functions $f(x)$,

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{1}$$

where $m(\mathbf{x})$ is the mean function, usually defined to be zero, and $k(\mathbf{x}, \mathbf{x}')$ 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).

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 709–712,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

- The kernel hyper-parameters can be learned via evidence maximization.
- GP provides full probabilistic prediction, and an estimate of uncertainty in the prediction.
- Unlike SVMs which need unbiased version of dataset for probabilistic prediction, yet does not take into account the uncertainty of $f(\mathbf{x})$, GP does not 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 Classification

In Gaussian Process Classification (GPC), we place a GP prior over a latent function $f(\mathbf{x})$ and then "squash" this prior through the logistic function to obtain a prior on $\pi(\mathbf{x}) \triangleq p(y = +1|\mathbf{x}) = \sigma(f(\mathbf{x}))$. Note that π is a deterministic function of f, and since f is stochastic, so is π.

Inference is divided into two steps: first, computing the distribution of the latent variable corresponding to a test case

$$p(f_*|X, \mathbf{y}, \mathbf{x}_*) = \int p(f_*|X, \mathbf{x}_*, \mathbf{f})p(\mathbf{f}|X, \mathbf{y})d\mathbf{f} \tag{2}$$

where $p(\mathbf{f}|X, \mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)/p(\mathbf{y}|X)$ is the posterior over the latent variables, and subsequently using this distribution over the latent to produce a probabilistic prediction

$$\pi_*(x) \triangleq (y_* = +1|X, \mathbf{y}, \mathbf{x}_*) \tag{3}$$

$$= \int \sigma(f_*)p(f_*|X, \mathbf{y}, \mathbf{x}_*)df_* \tag{4}$$

In classification, the non-Gaussian likelihood in Equation 2 makes the integral analytically intractable. Similarly, Equation 4 can also be analytically intractable for certain sigmoid functions. Therefore, we need an analytical approximation of integrals. We can approximate the non-Gaussian joint posterior with a Gaussian one, using Expectation Propagation (EP) method (Minka, 2001). EP, however, uses the probit likelihood

$$p(y_i|f_i) = \Phi(f_i y_i), \tag{5}$$

which makes the posterior analytically intractable. To overcome this hurdle in the EP framework, the likelihood is approximated by a *local likelihood approximation* in the form of an un-normalized Gaussian function in the latent variable f_i which defines the *site parameters* $\tilde{Z}_i$, $\tilde{\mu}_i$ and $\tilde{\sigma}_i^2$.

$$p(y_i|f_i) \simeq t_i(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma}_i^2) \triangleq \tilde{Z}_i \mathcal{N}(f_i|\tilde{\mu}_i, \tilde{\sigma}_i^2) \tag{6}$$

The posterior $p(\mathbf{f}|X, \mathbf{y})$ is approximated by $q(\mathbf{f}|X, \mathbf{y}) = \mathcal{N}(\mu, \Sigma)$, where $\mu = \Sigma\tilde{\Sigma}^{-1}\tilde{\mu}$, $\tilde{\Sigma}$ is diagonal with $\tilde{\Sigma}_{ii} = \tilde{\sigma}_i^2$, $\Sigma = (K^{-1} + \tilde{\Sigma}^{-1})^{-1}$, and K is the covariance matrix.

A practical implementation of Gaussian Process Classification (GPC) for binary class (Rasmussen and Williams, 2006) is outlined in the following algorithm:

Algorithm: Predictions for Expectation Propagation GPC.

Input:$\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\tau}}$ (Natural site param), X (Training inputs), $\mathbf{y}$ (Training targets), k (Covariance function), x_* (Test input).

Output: Predictive class probability.

```
1. $L :=$ cholesky$(I_n + \tilde{\mathrm{S}}^{1/2} K \tilde{\mathrm{S}}^{1/2})$
2. $z := \tilde{\mathrm{S}}^{1/2} L^T \backslash (L \backslash \tilde{\mathrm{S}}^{1/2} K \tilde{\boldsymbol{\nu}})$
3. $\overline{f}_* := \mathbf{k}(\mathbf{x}_*)^T (\tilde{\boldsymbol{\nu}} - \mathbf{z})$
4. $\mathbf{v} := L \backslash (\tilde{\mathrm{S}}^{1/2} \mathbf{k}(\mathbf{x}_*))$
5. $V[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$
6. $\overline{\pi}_* := \Phi(\overline{f}_* / \sqrt{1 + V[f_*]})$
7. return: $\overline{\pi}_*$ (predictive class probability)
```

The natural site parameters $\tilde{\boldsymbol{\nu}}$ and $\tilde{\boldsymbol{\tau}}$ for Expectation Propagation GPC are found using EP approximation algorithm. Multi-class classification can be performed using either one-versus-rest or one-versus-one for training and prediction. For Gaussian Process classification, "one-vs-one" might be computationally cheaper, so we have used it to for subtasks A and C.

2.2 Multiple Kernel Gaussian Process

The covariance kernel $\mathbf{k}$ of Gaussian Process 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.

We have used Exponential kernel and Multi-Layer Perceptron kernel combined 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 are effective in detecting the simultaneous 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 Alpaydn, 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 uses 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

$$\mathbf{k}(x,x') = \mathbf{k_1}(x,x') + \mathbf{k_2}(x,x') + \ldots + \mathbf{k_n}(x,x'). \tag{7}$$

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 data is preprocessed with lemmatization and tokenization, using NLTK toolkit. Then train variable is assigned an integer value. A data dictionary is built using training sentences, and feature vectors for train sets are generated by encoding BoW representation. These feature vectors are given as input to build the MKGPC model.

The Multi-Kernel Gaussian Process Classification (MKGPC) 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 classification model using
     appropriate kernel function.
  3. Optimize the classification model with the
     hyper-parameters (length scale, variance,
     noise).
  4. Return the learned model.
end
```

There are different kernels that can be used to build a GPC model. 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.

$$\mathbf{k}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right). \tag{8}$$

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.

$$\mathbf{k}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')}{2l^2}\right) \tag{9}$$

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

$$\frac{2\sigma^2}{\pi}\sin^{-1}\frac{(\sigma_w^2 x^T x' + \sigma_b^2)}{\sqrt{\sigma_w^2 x^T x + \sigma_b^2 + 1}\sqrt{\sigma_w^2 x'^T x' \sigma_b^2 + 1}} \tag{10}$$

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 Results and Discussion

The output submitted for the task was obtained using MKGPC with Radial Basis Function kernel and Exponential Kernel. We also used Multi-Layer Perceptron Kernel. The results of the SGPC using SE kernel for subtask B and MKGPC for subtask B are shown in Table 1. The evaluation was done on SemEval-2017 labeled test dataset. Only 1000 tweets were used to train the model due to the time-complexity of GP and hardware limitations, and from among the remaining 9551 tweets test set was taken.

Table 1: A Performance Evaluation based on Recall, F-measure and Precision (all macro-averaged) for subtask B

Model	Recall	F-measure	Precision
SGPC	0.57	0.58	0.64
MKGPC(R+E)	0.56	0.56	0.63
MKGPC(R+M)	0.61	0.62	0.64
MKGPC(R+E+M)	0.62	0.63	0.64

The kernel combinations used in Table 1 are

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

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

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

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

We observe from Table 1 that though the macro-averaged precision of the MKGPC models is the same as SGPC, their macro-averaged recall and F-measure are better than SGPC (except for MKGPC(R+E)), because the Multi-Layer Perceptron kernel learns the periodicity better than RBF and Exponential kernels do. These different models, when evaluated on dataset for subtask A and subtask C, exhibited similar performance as in subtask B. The system underperform compared to the baseline system in task C, and to logistic regression on 1-gram in tasks A and B since only a small fraction of the dataset was used for training.

5 Official Evaluation

Our system scored a macro-averaged recall of 0.431 and was ranked 35 for subtask A, macro-averaged recall of 0.586 and was ranked 20 for subtask B, and macro-averaged mean absolute error of 1.325 and was ranked 15 for subtask C.

6 Conclusion

In this paper, we have presented a Gaussian Process classification model for sentiment analysis in Twitter. We used Bag of Words feature vectors and fixed rule multi kernel learning to build the GP model. We observed that combining Multi-Layer Perceptron kernel improves the performance of the system, perhaps due to its more effective learning of the periodicity property in the dataset. There is scope for enhancing the results by using different feature generation algorithms, different multi-kernel learning approaches, and increasing the data size.

References

C. Chew and G. Eysenbach. 2010. Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak. *PloS ONE* 5(11):1–13.

Trevor Cohn, Daniel Beck, and Lucia Specia. 2014. Joint emotion analysis via multi-task gaussian processes. In *Proceedings of EMNLP 2014, the International Conference onEmpirical 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.

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

B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology* 60(11):21692188.

T P Minka. 2001. *Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT.

Mahmoud Nabil, Mohamed Aly, and Amir F. Atiya. 2016. Cufe at semeval-2016 task 4: A gated recurrent model for sentiment classification. In *Proceedings of SemEval-2016*. ACL, pages 52 –57.

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

S. Verma, S. Vieweg, W. J. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. M. Anderson. 2011. Natural language processing to the rescue? extracting situational awareness tweets during mass emergency. In *Proceedings of 5th International Conference on Web and Social Media (ICWSM)*.

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