INCREASING CLASSIFICATION ACCURACY IN LIBSVM USING STRING KERNEL FUNCTIONS

Ivan Homoliak
Master Degree Programme (2), FIT BUT
E-mail: xhomol11@stud.fit.vutbr.cz

Supervised by: Maroš Barabas
E-mail: ibarabas@fit.vutbr.cz

Abstract: This paper explores dependencies of text classification used with string kernel functions. There are described experiments with single string kernel function and also experiments with combinations of them with arithmetic operations of addition and multiplication. Gathered results are applied to detect spam messages of e-mail communication.

Keywords: classification, libSVM, spam detection, string kernel functions

1 INTRODUCTION

In general, it is known that text objects, which we want to separate are represented by their mappings into n-dimensional feature vectors representing properties of the text objects. We can accomplish the classification of these objects in multidimensional feature space, where we try to find a hyperplane, which is capable to separete clusters of these objects. In such approach there exists one disadvantage – it does not designate the criterion of classification. In most cases this criterion is abstract, because it is determined by an inteligent being. If we want some algorithm or system to work at an abstract level of classification, it is neccessary to learn how to distinguish objects according to the mentioned abstract criterion.

This learning or training is made in the area of artificial inteligence with training patterns of data, which are classified by the inteligent being – human. After training it is possible to make decisions similiar to those made by the human being.

We face the question, whether the decision made by trained classificator is enough to accept it. We can respond the answer to this question only after series of experiments aimed to modify settings of classification algorithms and after modififfications made on the classification algorithms itself.

Our work aims to increase the classification accuracy by modififfications of generated inner product of string kernel functions. Theese products are modified by combinations of two or more products of different string kernel functions with arithmetic operations of addition and multiplication.

2 THE PROCESS OF EXPERIMENS

In suggested experiments, we kept the following:

1. We used normalized text strings with labels of associated classes as the input for implemeted console application. The application generate the inner product matrix of all combinations of particular strings.

2. Console application used string kernel functions to determine the rate of similiarity of two text strings.

3. The output of the mentioned console application was a file containing inner product matrix of all combinations of input strings.

4. We used the inner product matrix as the input for SVM classificator and it computes the classification accuracy in the n-fold cross validation manner with respect of used paramters.

5. We recorded results of the classification into appropriate tables for the evaluation of approach effectiveness later.

Experiments with the suggested modification combine the inner product of the different string kernel function. We slightly modify the second step of the process: we generate one kernel matrix for each kernel function. These matrixes are merged into one matrix by the application of the appropriate binary operator after that.

We suggested and implemented some python scripts managing execution of these experiments. Their purpose relates with the use of different kernel functions and their specific parameters used during the generation of inner products matrixes. Another python script serves to combine the cost parameter of the classificator and inner product matrixes. The process is illustrated in figure 1.

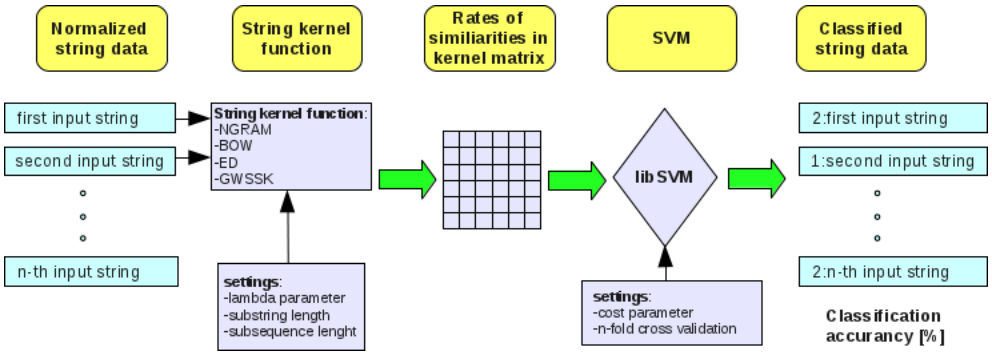

Figure 1: The process of classification based on string kernel functions [1].

3 STRING KERNEL FUNCTIONS

The string kernel function is a function with two string arguments and it returns an inner product representing the rate of similiarity of input strings. Kernel oriented methods of learning use implicit mapping of input data to multidimensional feature space defined by string kernel function [2].

Subsequence kernel (SSK) is based on searching of the same subsequences in input strings and according to all occurences it returns the similiarity rate.

Gap weighted subsequence kernel (GWSSK) – it works similar as subsequence function based on searching of the same subsequences, but it differs in taking into consideration also character of these subsequences (consistency weighted by the number and length of gaps).

Bag of words (BOW) – it uses mapping of a text documents into bags (sets) of words and it counts occurences of particular words.

N-gram (NGRAM) assumes n-gram as a substring compounds from n adjacent characters or words. It works similar as the bag of words but it considers n-grams instead of words in BOW. The higher dimensionality is also implied than in the case of BOW.

Edit distance (ED) is also called Levenshtein distance and it measures the minimal operations count of deletion, insertion and substitution neccessary to transform the first input string to the second [3].

4 COMBINATION OF INNER PRODUCTS

We use the approach which tries to increase the classification accuracy in the text string data according to identified facts and properties of string kernel functions. It realizes operations of addition and multiplication of inner products generated by particular string kernel functions. First experiments were performed on Reuters database. There were combined pairs of different string kernel functions by mentioned operators. Small improvements were achieved with the *plus* operator. The multiplication of kernels did not bring any improvements. We experimented with combinations of three and four string kernel functions later, but these experiments did not bring any improvements.

Some improvements (emphasized with bold font) were achieved with combined inner products of kernel functions in experimens with spam detection. Gathered results are depicted in table 1.

Part of the message	**Classification accuracy of used string kernel function or combination of them [%]**							
	NGRAM	**GWSSK**	**ED**	**BOW**	**NGRAM+ED**	**ED+GWSSK**	**NGRAM+BOW**	**BOW+GWSSK**
Header	100.000	100.000	100.000	91.752	100.000	100.000	100.000	100.000
Body	98.648	99.027	98.080	73.148	**98.999**	98.999	90.076	98.810
Subject	97.377	97.620	96.593	83.315	**97.404**	97.620	94.943	97.377

Table 1: Summary results of *plus* combination of kernel functions applied on the spam detection.

5 CONCLUSION

The goal of this paper was to find out the influnce of combination of the string kernels onto classification accuracy. Therefore it was necessary to experiment with simple string kernel functions at first, where we were searching for optimal parameters of their performance (the length of substrings in the case of NGRAM, the length of subsequences in the GWSSK case and the decay factor lambda). The next goal was to optimize the performance of the classificator SVM [4].

The knowledge gained from experiments was applied as spam detection. Later, experiments were run with particular string kernel function and with their combinations. There were only used those combinations, which bring results in experiments with text strings. A small improvement was achieved in the comparison with experiments where the separated kernel functions were used. So we can assume the operator *plus* as a perspective method combining products of string kernel functions. The problem of experiments with the spam detection was, that we used training data containing ham messages simple to detect. So it resulted into 100% classification accuracy in some cases and we were unable to assess the improvement of kernels combinations at the dataset.

Suggested methods could be used as spam filter in the future. The problem of large time complexity at the training phase still remains, but it could be solved by the distributed computation or the hardware implementation and achieve the minimum latence in the throughput of legitimate e-mail messages.

REFERENCES

[1] Z. Michlovský. High-speed systems for identification of attacks and malware. Technical report, Brno University of Technology, Faculty of Information Technology, 2009.

[2] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[3] D. Pearson and J.C. Janodet. String distances and uniformities. *Adaptive and Natural Computing Algorithms*, pages 333–339, 2009.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, May 2011.