A Hybrid Method for Textual Data Classification Based on Support Vector Machine with Particle Swarm Optimization Metaheuristic and k-Means Clustering

Konstantinas Korovkinas

Kaunas Faculty, Vilnius University, Muitines str. 8, LT-44280, Kaunas, Lithuania
konstantinas.korovkinas@knf.vu.lt

Abstract. This paper introduces a hybrid method for textual data classification. The goal of this paper is to improve classification accuracy of method presented in our previous work by integrating to it k-Means method for decreasing training dataset and particle swarm optimization metaheuristic for a linear support vector machine parameter tuning. The paper reports that the introduced method is characterized by higher improvements in all effectiveness metrics than the methods presented in our previous works.

Keywords: Support Vector Machine · Particle Swarm Optimization · k-Means · Textual Data Classification

1 Introduction

Textual data analysis became a very popular since people started using Internet, to be more concrete when e-shops, social networks, Blogs etc., where people can write their comments, appeared. This area is considered as a very challenging – although a lot of work has been done in this field, accuracy is still rather average due to comments, slang, smiles etc. A Support Vector Machine (SVM) is one of the most widely used method which has proved its efficiency in different tasks and domains. It is very flexible to parameter tuning, as well as internal modifications, which allows to improve its performance and accuracy. However, despite all advantages, typical for SVM algorithm is, that it is characterized by slow performance in the big data arrays. The higher number of features is, the longer computation time it requires. There have been a number of efforts to speed up SVM, and most of them focus on reduction of the training set [15, 18, 22]. One of the most widely known and promising method for that is k-Means [5, 8, 23], which can be used as standalone method or in combination with others. Aforementioned authors conclude, that properly selected training data can improve executing time with no losing or similar accuracy. Increasing accuracy is another common problem. Particle swarm optimization (PSO) is a very promising option [9, 11, 16]. One of its strengths is combination with other evolutionary methods. Its efficiency is also proved in SVM parameter selection tasks [6, 10, 21].

Motivated by these improvements, this paper proposes a hybrid method for textual data classification, which is suitable to work with large datasets. The proposed hybrid method is a combination of three methods: SVM, k-Means and PSO, which are integrated into SpeedUP method, earlier presented in [12]. Standalone SpeedUP method increased SVM classification speed, while slightly lost to ordinary SVM method in terms of accuracy [12]. Considering on it, were proposed two separate methods for improving classification accuracy: k-Means method in [13] – for training data reduction and PSO in [14] – for finding the best cost (penalty) parameter for SVM. Both aforementioned methods still lost to ordinary SVM, when are used separately, so it led to conclusion of possibility to combine these methods. The rest of the paper is organized as follows. Section 2 introduces the methods which were used in the experiments and proposed method is described, whereas Section 3 gives a description of datasets and experimental settings used to evaluate proposed approach, together with results obtained during experimenting. Finally, Section 4 outlines the conclusions.

2 Methodology

This section shortly presents the methods relevant to research presented in this paper: Support Vector Machine [1, 2, 4], k-Means [17], Particle Swarm Optimization [3] and Term Frequency — Inverse Document Frequency (TF-IDF) [20]. The brief description of a proposed hybrid method, also presented herein.

2.1 Relevant methods

TF-IDF Since machine learning algorithms cannot work with text data directly, it should be converted into vector of numbers. TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. This calculation determines how relevant a given word is in particular document. *TfidfVectorizer* module from *scikit-learn* [19] library is used to implement TF-IDF.

Support Vector Machine Herein is used linear SVM (LSVM), which is optimized for large-scale learning. *LinearSVC* module from *scikit-learn* [19] library is used to implement LSVM.

k-Means The main idea of this method is to partition the input dataset into k clusters, represented by adaptively-changing centroids. k-Means computes the squared distances between the input data points and centroids, and assigns inputs to the nearest centroid. *KMeans* module from *scikit-learn* [19] library used to implement k-Means method.

Particle Swarm Optimization It is a population-based stochastic meta-heuristic algorithm for solving continuous and discrete optimization problems. Herein is used global best variant, which was manually programmed using Python language and adopted for LSVM parameter tuning for textual data classification tasks.

2.2 The proposed method

The proposed hybrid method ($LSVM^{PSO}\_km\_30K\_SpeedUP$) is a combination of three methods: SVM, k-Means and PSO, which are integrated into SpeedUP method. The main idea of it is to reduce training dataset size regarding to subset size. Thus the testing dataset is split into equal subsets and the size of training data is calculated on the basis of the first subset size. k-Means and PSO methods are used for increasing accuracy of SpeedUP.

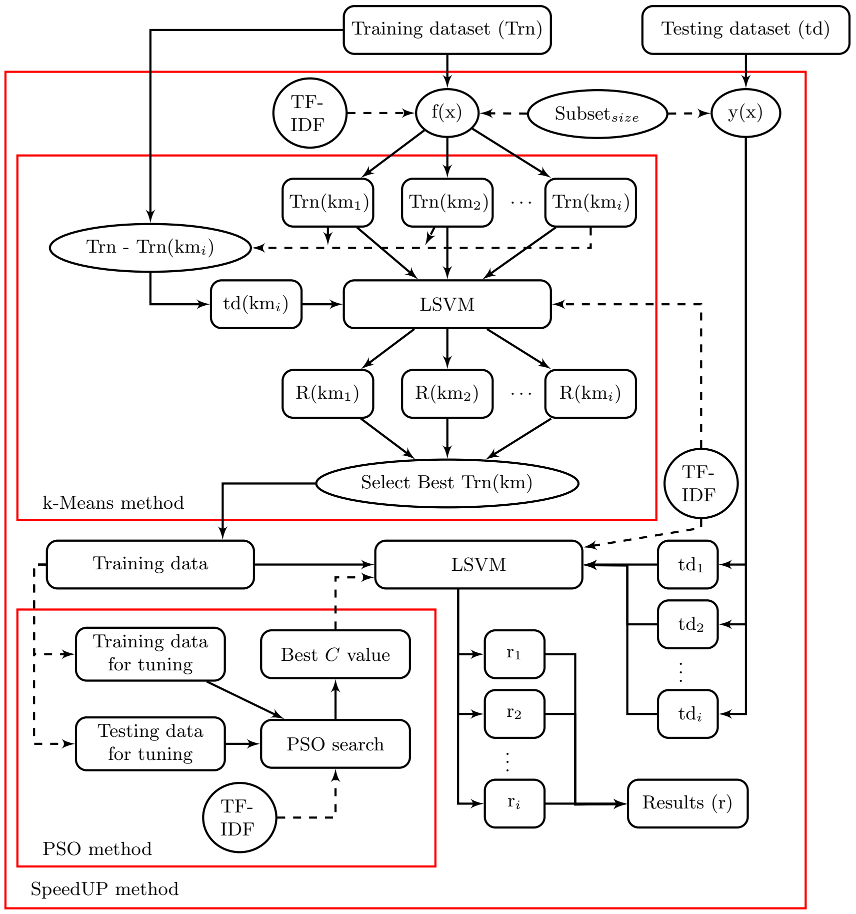

Fig. 1. Diagram of proposed hybrid method

Fig. 1 presents a diagram of the proposed method. The parts of proposed method are presented in the region bounded by a rectangle. Dashed line in figure means extra calculations steps, which are needed to be executed before the concrete step of main algorithm is joined with.

List of parameters used in diagram:

Trn – training dataset
td – testing dataset
$\mathbf{Subset_{size}}$ – size of testing data subsets td_i into which testing dataset is divided
$\mathbf{y(x)}$ – function which divides testing dataset into subsets according $Subset_{size}$
$\mathbf{f(x)}$ – function which calculates training data size according $Subset_{size}$ and defines the number of training data sets
$\mathbf{Trn(km_i)}$ – sets of training data after *k-Means* method is applied
$\mathbf{td(km_i)}$ – testing data for *k-Means* method
$\mathbf{R(km_i)}$ – results of every training data subset
Select Best Trn(km) – method, which returns the best training data selected by *k-Means* method according to the results $R(km_i)$
$\mathbf{r_i}$ – sets of results achieved from each subset
Results – the final result set, which contains results of all r_i sets

Diagram consists of steps as follows:

1. Before passing to SpeedUP method, training and testing datasets are preprocessed. Preprocessing contains two actions: text preprocessing and data cleaning. Text preprocessing includes actions like converting to lowercase, removing redundant tokens such as hashtag, symbols @, numbers, "http" for links, punctuation symbols, usernames etc. Data cleaning performs dataset checking for empty strings and removing them.
2. Depending on $Subset_{size}$, function $f(x)$ calculates training data size and number of sets for k-Means method.
3. Selected training data is converted into vector of numbers with TF-IDF and passed to k-Means method for the selection of the best training data for SpeedUP method, which is performed depending on results $R(km_i)$.
4. The best training data selected is converted into vector of numbers with TF-IDF and passed to PSO method, which returns the best C value for LSVM. After, the same training data is passed into LSVM. LSVM is trained with it and parameter C is set to one, which is returned from PSO method.
5. Depending on $Subset_{size}$ testing dataset is dividing into subsets td_1, td_2 etc. (function *y(x)*).
6. Subsets are converted into vector of numbers with TF-IDF and are passed to LSVM algorithm one by one and achieved results are stored in separate sets r_1 – for td_1, r_2 – for td_2 etc.
7. The results are combined into one result set – Results.

3 Experiments and results

3.1 Dataset

In this paper are used two existing labeled datasets available: The Stanford Twitter sentiment corpus (sentiment140[1]) dataset and Amazon customer reviews dataset[2]. The Stanford Twitter sentiment corpus dataset is introduced in [7] and contains 1.6 million tweets automatically labeled as positive or negative based on emotions. The dataset is split to 70% (1.12M tweets) for training and 30% (480K tweets) for testing. Amazon customer reviews dataset contains 4 million reviews and star ratings; it was also split to select 70% (2.8M reviews) entries for training and 30% (1.2M reviews) for testing.

3.2 Experiments and settings

The main goal of this research is to improve classification accuracy of method *30K_SpeedUP* introduced in [12] by integrating k-Means (*km_30K_SpeedUP*) and PSO ($LSVM^{PSO}$*_30K_SpeedUP*) methods presented respectively in [13] and [14], also to compare the aforementioned methods with proposed method by performing comparative analysis between them. Two experiments are performed to reach the goal: one experiment with the Stanford Twitter sentiment corpus dataset (sentiment140) and second experiment with Amazon customer reviews dataset (Amazon reviews). Table 1 shows the sizes of training and testing data for LSVM input. It is assumed that the testing subset size should be 30K instances (30%), then training data calculated dependently on subset size is 70K instances (70%). All testing data is divided into subsets containing 30K instances (the last subset is the remainder and it could contain less than 30K instances).

Table 1. Experimental settings [12]

Exp. No.	Dataset	Testing data size (TDs)	Subset size (Ss)	Subsets quantity (SQ) trunc(TDs/Ss)	Remainder TDs-(Ss*SQ)	Calculated training data dependently on Ss
1	sentiment140	480K	30K	16	0	70K
2	Amazon reviews	1.2M	30K	40	0	70K

Python programming language and *scikit-learn* [19] library for machine learning were used to implement and evaluate the proposed method.

Computer with processor Intel(R) Core(TM) i7-4712MQ CPU 2.30 GHz and 16.00 GB installed memory (RAM) is used for experiments.

[1] http://help.sentiment140.com/

[2] https://www.kaggle.com/bittlingmayer/amazonreviews/

3.3 Performance evaluation

Effectiveness is measured using statistical measures: accuracy (ACC), precision (PPV – positive predictive value and NPV – negative predictive value), recall (TPR -– true positive rate and TNR -– true negative rate) and $F_1 score$ (harmonic mean of PPV and TPR).

3.4 Results

Two experiments were performed to evaluate the effectiveness of proposed method in terms of accuracy, precision, recall and $F_1 score$. Table 2 presents averaged results for proposed method in comparison with *30K_SpeedUP*, *km_30K_SpeedUP* and $LSVM^{PSO}$*_30K_SpeedUP*. It is worth to mention, that all experiments were redone by using the same training and testing datasets for all methods, so they can be different from results in previous works. Also k-Means method from previous work was implemented without SVM tuning part, because it is a part of PSO method.

Table 2. Experimental results

Method	ACC	PPV	NPV	TPR	TNR	$F_1 score$
The Stanford Twitter sentiment corpus dataset						
30K_SpeedUP	77.10%	76.60%	77.62%	78.05%	76.16%	77.32%
km_30K_SpeedUP	77.30%	76.78%	77.83%	78.26%	76.33%	77.51%
$LSVM^{PSO}$*_30K_SpeedUP*	77.90%	77.46%	78.35%	78.70%	**77.09%**	78.08%
$LSVM^{PSO}$*_km_30K_SpeedUP*	**78.12%**	**77.55%**	**78.72%**	**79.17%**	77.08%	**78.35%**
The Amazon customer reviews dataset						
30K_SpeedUP	87.59%	87.50%	87.68%	87.71%	87.47%	87.60%
km_30K_SpeedUP	87.74%	87.76%	87.72%	87.71%	87.76%	87.73%
$LSVM^{PSO}$*_30K_SpeedUP*	**88.46%**	**88.63%**	88.28%	88.22%	**88.69%**	88.43%
$LSVM^{PSO}$*_km_30K_SpeedUP*	88.45%	88.57%	**88.33%**	**88.29%**	88.61%	88.43%

The results clearly show, that $LSVM^{PSO}$*_km_30K_SpeedUP* performs better compared with *30K_SpeedUP* method – 1.02%, *km_30K_SpeedUP* – 0.80% and $LSVM^{PSO}$*_30K_SpeedUP* – 0.22%, when applied on sentiment140 dataset. In the case of Amazon reviews dataset the proposed hybrid method also performs better compared with *30K_SpeedUP* – 0.86% and *km_30K_ SpeedUP* – 0.71%, while slightly lost (0.01%) to $LSVM^{PSO}$*_30K_SpeedUP*.

Other metrics in terms of — PPV, NPV, TPR, TNR, $F_1 score$ – also show the superiority of the proposed method compared with previously introduced methods on sentiment140 dataset, while $LSVM^{PSO}$*_30K_SpeedUP* slightly outperform it in term of TNR. In the case of Amazon reviews dataset the proposed hybrid method perform better in terms of — NPV, TPR, while slightly lost to $LSVM^{PSO}$*_30K_SpeedUP* in terms of — PPV, TNR.

4 Conclusions

The main advantage of the proposed hybrid method is that training data selection is performed with k-Means method, which ensure the variety of the training data and could positively affect PSO metaheuristics choice in finding the best cost parameter C for LSVM. When training data is selected randomly, there is a risk, that training data will contain the same data or data will be not useful and this could negatively affect accuracy in different runs; therefore, multiple runs are required for more objective results, which is affecting classification speed. The proposed method increased the classification accuracy, without minor losses in classification speed to compare with previously presented methods. It is also proved that by using only 70,000 instances for training the proposed hybrid method can classify much bigger testing datasets (starting from 480K, 1.2M etc.) with minor losses in accuracy.

Acknowledgments. I would like to thank my supervisor Prof. Dr. Gintautas Garšva for support and advices.

References

1. Boser, B. E., Guyon, I. M., Vapnik, V. N. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, 144–152 (1992)
2. Cortes, C., Vapnik, V. Support-vector networks. Machine learning, 20(3), 273–297 (1995)
3. Eberhart, R., Kennedy, J. A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 39–43 (1995)
4. Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., Lin, C. J. LIBLINEAR: A library for large linear classification. Journal of machine learning research, 9(Aug), 1871–1874 (2008)
5. Gan, J., Li, A., Lei, Q.L., Ren, H., Yang, Y. K-means based on active learning for support vector machine. In: Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on, 727–731 (2017)
6. Garšva, G., Danėnas, P. Particle swarm optimization for linear support vector machines based classifier selection. Nonlinear Analysis: Modelling and Control, 19(1), 26–42 (2014)
7. Go, A., Bhayani, R., Huang, L. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12) (2009)
8. Gu, Q., Han, J. Clustered support vector machines. In: Artificial Intelligence and Statistics, 307–315 (2013)
9. Gu, X., Li, T., Wang, Y., Zhang, L., Wang, Y., Yao, J. Traffic fatalities prediction using support vector machine with hybrid particle swarm optimization. Journal of Algorithms & Computational Technology, 12(1), 20–29 (2018)
10. Hoang, T. T., Cho, M. Y., Alam, M. N., Vu, Q. T. A novel differential particle swarm optimization for parameter selection of support vector machines for monitoring metal-oxide surge arrester conditions. Swarm and Evolutionary Computation, 38, 120–126 (2018)

11. Huang, C. L., Dun, J. F. A distributed PSO–SVM hybrid system with feature selection and parameter optimization. Applied soft computing, 8(4), 1381–1391 (2008)
12. Korovkinas, K., Danėnas, P., Garšva, G. SVM Accuracy and Training Speed Trade-Off in Sentiment Analysis Tasks. In: International Conference on Information and Software Technologies, 227–239 (2018)
13. Korovkinas, K., Danėnas, P., Garšva, G. SVM and k-Means Hybrid Method for Textual Data Sentiment Analysis. Baltic Journal of Modern Computing, 7(1), 47–60 (2019)
14. Korovkinas, K., Danėnas, P., Garšva, G. Support Vector Machine Parameter Tuning Based on Particle Swarm Optimization Metaheuristic. Nonlinear Analysis: Modelling and Control, 25(2), 266–281 (2020)
15. Lee, Y.J., Mangasarian, O.L. RSVM: Reduced support vector machines. In: Proceedings of the 2001 SIAM International Conference on Data Mining, 1–17 (2001)
16. Lin, S. W., Ying, K. C., Chen, S. C., Lee, Z. J. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert systems with applications, 35(4), 1817–1824 (2008)
17. MacQueen, J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, No. 14, 281–297 (1967)
18. Mourad, S., Tewfik, A., Vikalo, H. Data subset selection for efficient SVM training. In: Signal Processing Conference (EUSIPCO), 2017 25th European, 833–837 (2017)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825–2830 (2011)
20. Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, 242(Dec), 133–142 (2003)
21. Sunkad, Z. A. Feature selection and hyperparameter optimization of SVM for human activity recognition. In: 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI), 104–109 (2016)
22. Wang, S., Li, Z., Liu, C., Zhang, X., Zhang, H. Training data reduction to speed up SVM training. Applied intelligence, 41(2), 405–420 (2014)
23. Yao, Y., Liu, Y., Yu, Y., Xu, H., Lv, W., Li, Z., Chen, X. K-SVM: An Effective SVM Algorithm Based on K-means Clustering. JCP, 8(10), 2632–2639 (2013)