

Evolutionarily-tuned support vector machines

Wojciech Dudzik
Silesian University of Technology
Gliwice, Poland
wojciech.dudzik@polsl.pl

Michal Kawulok
Silesian University of Technology
Gliwice, Poland
michal.kawulok@polsl.pl

Jakub Nalepa
Silesian University of Technology
Gliwice, Poland
jakub.nalepa@polsl.pl

ABSTRACT

Support vector machine (SVM) classifiers can cope with many different classification tasks but improperly selected hyperparameters may deteriorate their performance. Moreover, datasets are getting bigger in terms of their size and the number of features. This is often coupled with low training data quality and presence of redundant features, which can adversely affect classification accuracy and time performance. Furthermore, high memory and computational complexity of SVM training can be a limiting factor for its application over huge datasets. We address these issues with evolutionarily-tuned SVM, where we utilize evolutionary algorithms for optimizing hyperparameters, along with selecting features and training instances. The performance of our method is compared on several benchmark datasets to other methods for optimizing SVMs, as well as to other classifiers. The results show that our algorithm gives high performance in both accuracy and classification time when compared with the state-of-the-art methods for SVM optimization.

CCS CONCEPTS

• **Computing methodologies** → **Support vector machines; Genetic algorithms; Feature selection;**

KEYWORDS

support vector machine, feature selection, training set selection, memetic algorithm

ACM Reference Format:

Wojciech Dudzik, Michal Kawulok, and Jakub Nalepa. 2019. Evolutionarily-tuned support vector machines. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion)*, July 13–17, 2019, Prague, Czech Republic. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3319619.3321924>

1 INTRODUCTION

SVMs are popular classifiers that were applied to solve a variety of tasks. Nonetheless, using SVM is hard due to its sensitivity to hyperparameters (\mathcal{M}). Here, we consider radial basis function as kernel, hence the SVM depends on two hyperparameters: kernel width γ and a SVM slack variable C . As grid search (GS) approach to tune \mathcal{M} is computationally expensive, more efficient methods

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6748-6/19/07...\$15.00

<https://doi.org/10.1145/3319619.3321924>

were developed, including evolutionary algorithms (EA), such as particle swarm optimization [3] and genetic algorithms [1].

The problem of costly training can be tackled with proper selection of training set (T). Using a subset of T (denoted as T') can be sufficient, as only a small part of T is chosen as support vectors. Methods used for T' selection can be divided into two groups: (i) the ones whose complexity is dependent on the cardinality of T , and (ii) those that are independent from T . The methods from the first group exploit the information about the layout of T , including clustering method [8], and exploiting statistical properties of T [4]. The second group includes, among others, EA, with a highly effective memetic algorithm to select training data for SVM (MASVM)[7].

The problem of redundant features can be solved with selection of a feature set (F). We cannot *a-priori* know which features are useful for a classifier. In most cases, selecting a feature subset (F') for SVM is coupled with hyperparameters optimization [5, 6]. Other interesting work is feature selection followed by alternating memetic algorithm (FSALMA) presented in [2], where we optimize \mathcal{M} and T' on preselected F .

All these problems can make it hard to use SVMs. In this paper, we propose a memetic algorithm ESVM, which unfolds to the Evolutionarily-tuned SVM) for optimizing \mathcal{M} , T' and F' .

2 EVOLUTIONARY ALGORITHMS IN SVM

A general schema of ESVM is presented in Figure 1. This algorithm is composed of three optimization phases, in which we optimize: (i) \mathcal{M} (differential evolution), (ii) T' , (iii) F' . First, our workflow starts with ranking F using committee of mutual information, variance thresholding, recursive feature elimination and stability selection, averaging their results. These scores are used for the roulette wheel initialization scheme of F' phase, where K_F features are selected. Initialization of T' starts with choosing a random subset of T in size of K_T , \mathcal{M} are set deterministically with a logarithmic step in range $[10^{-5}, 10^3]$. The \mathcal{M} phase starts with reduced T' and F' which are best individuals assessed after initialization.

At the beginning of the \mathcal{M} optimization, we create $Q = 20$ individuals with the crossover process, where hyperparameter values of a new child become $x_{a+b} = x_a + \alpha \cdot (x_a - x_b)$, where α is the weight randomly drawn from the interval $[-0.5, 1.5]$. Next, each individual is mutated, with probability $P_m = 0.1$, by modifying a value x within a range $x \in [x - u \cdot x, x + u \cdot x]$, where u is randomly drawn from $[0.0, 0.1]$. Eventually, we select the Q fittest individuals to maintain a constant size of population. The process of evolution lasts till there is no improvement in the average fitness of population, otherwise we switch to optimize T' .

The phases of T' and F' are similar. In both cases, a population is processed with crossover operator in which for each selected pair of individuals two sets (of F' or T') are summed. Next, we select no more than K_T (or K_F) elements. Then each individual is

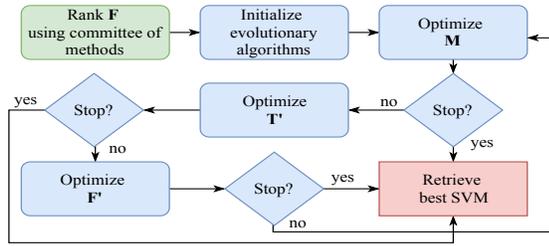


Figure 1: Flowchart of ESVM.

subjected to mutation with the probability $P_t = 30\%$ for T' and $P_f = 50\%$ for F' . Mutation for T' consists in replacing $f_t = 20\%$ of examples with random ones from T . For F' , $f_f = 10\%$ features are replaced by ones that are above mean in feature ranking used for initialization. In both algorithms, we exploit information about the previous individuals to enhance the current solutions and adaptively grow K_T and K_F . At the end, the selection rule from M phase is used. The switching scheme lasts till one of the phases improves fitness. If no improvement can be made, we perform regeneration of all individuals. After $R = 5$ regenerations, we take the best model from the entire evolution process.

3 EXPERIMENTS AND DISCUSSION

All of the experiments were run on Intel i5-6500 CPU. For measuring fitness, we rely on area under the receiver operating characteristic curve (AUC), measured on validation set. We picked three balanced binary datasets: Ionosphere 351×34 (vectors by features), Madelon 2600×500 and Gisette 7000×5000 . The datasets were divided into 5 folds containing training, validation and test set (Ψ) in 3:1:1 proportion, respectively. The presented results are average scores over all 5 folds. Each EA was run 10 times per fold. The best AUC scores and times are boldfaced in tables.

As it can be seen in Table 1, ESVM yields best results on the Gisette and Madelon datasets in comparison to other methods. Our algorithm performs best, regarding classification time, being multiple times faster than any other technique, while having high AUC scores at the same time. This fast classification was results from smaller number of support vectors and reduced F . We performed

Table 1: Results obtained for benchmark datasets.

Set	Algorithm	Training time [s]	Test set AUC	Classification Time of Ψ [ms]
Ionosphere	FSALMA	0.34 ± 0.15	0.953 ± 0.042	0.12 ± 0.06
	GS	0.29 ± 0.03	0.974 ± 0.024	0.25 ± 0.07
	ESVM	0.82 ± 0.45	0.950 ± 0.036	0.08 ± 0.03
	MASVM	0.26 ± 0.23	0.964 ± 0.031	0.17 ± 0.05
Madelon	FSALMA	87.7 ± 15.8	0.814 ± 0.036	21.2 ± 4.1
	GS	196.6 ± 0.4	0.656 ± 0.027	280.8 ± 4.0
	ESVM	215.1 ± 154.3	0.861 ± 0.054	7.0 ± 3.4
Gisette	MASVM	366.9 ± 182.9	0.578 ± 0.104	194.6 ± 18.6
	FSALMA	358 ± 446	0.993 ± 0.002	817.9 ± 779.5
	GS	11147 ± 768	0.996 ± 0.002	8453.9 ± 704.2
	ESVM	388 ± 499	0.996 ± 0.002	175.9 ± 144.1
	MASVM	1015 ± 227	0.986 ± 0.013	5026.6 ± 18.8

Table 2: Comparison of ESVM with other popular classifiers. Scores are AUC on Test set averaged over 5-folds.

Algorithm	Set	Ionosphere	Madelon	Gisette
ExtraTree		0.987	0.698	0.996
Lasso		0.907	0.640	0.996
KNN		0.920	0.581	0.988
Logistic Regression		0.918	0.630	0.996
ESVM		0.950	0.861	0.996

Wilcoxon test to verify whether the differences between the results obtained with ESVM and other algorithms are statistically significant. Test shows $p < 0.001$ for all algorithms on Madelon and Gisette datasets while on Ionosphere $p > 0.05$ was observed for comparison with GS and FSALMA and $p < 0.005$ when compared with MASVM. Thus our algorithm is statistically different for two out of three presented datasets.

In Table 2, we compared our ESVM with other popular classifiers. For KNN we used $K = 5$ and for ExtraTree the number of trees was set to $N = 100$. In case of Lasso and Logistic Regression, we used cross-validation for hyperparameter tuning. ESVM outperformed other methods on Madelon dataset and achieved the same as others high AUC for Gisette dataset while being only slightly worse on Ionosphere dataset, where it outperformed all but ExtraTree. These scores show ESVM as competitive method.

4 CONCLUSIONS

In this paper, we introduced a new memetic algorithm that is the first to select M , T' and F' . The experiments were run on three benchmark datasets, so we should still be careful in premature conclusions. Nonetheless, the results of ESVM show it has a great potential for providing high AUC scores and giving the best classification times among evolutionary methods.

ACKNOWLEDGMENTS

This work was supported by the National Science Centre (DEC-2017/25/B/ST6/00474). WD was co-financed by the European Union through the European Social Fund (POWR.03.02.00-00-1029) and by the Silesian University of Technology (02/020/BKM18/0155).

REFERENCES

- [1] J.-S. Chou, M.-Y. Cheng, Y.-W. Wu, and A.-D. Pham. 2014. Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification. *Expert Systems with Applications* 41, 8 (2014), 3955–3964.
- [2] W. Dudzik, J. Nalepa, and M. Kawulok. 2018. Automated Optimization of Non-linear SVMs for Binary Classification. In *InCoS*. Springer, 504–513.
- [3] X. Zhang et al. 2015. SVM with parameter optimization by a novel hybrid method and its application to fault diagnosis. *Neurocomputing* 149 (2015), 641 – 651.
- [4] L. Guo and S. Boukir. 2015. Fast data selection for SVM training using ensemble margin. *Pattern Recognition Letters* 51 (2015), 112–119.
- [5] C.-L. Huang and C.-J. Wang. 2006. A GA-based feature selection and parameters optimization for SVM. *Expert Systems with Applications* 31, 2 (2006), 231 – 240.
- [6] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee. 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications* 35, 4 (2008), 1817 – 1824.
- [7] J. Nalepa and M. Kawulok. 2014. A Memetic Algorithm to Select Training Data for SVM. In *GECCO '14*. ACM, New York, NY, USA, 573–580.
- [8] X.-J. Shen and et al. 2016. Large-scale support vector machine classification with redundant data reduction. *Neurocomputing* 172 (2016), 189–197.