Support Vector Machines Training Data Selection Using a Genetic Algorithm

Michal Kawulok and Jakub Nalepa*

Institute of Informatics, Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
{michal.kawulok,jakub.nalepa}@polsl.pl

Abstract. This paper presents a new method for selecting valuable training data for support vector machines (SVM) from large, noisy sets using a genetic algorithm (GA). SVM training data selection is a known, however not extensively investigated problem. The existing methods rely mainly on analyzing the geometric properties of the data or adapt a randomized selection, and to the best of our knowledge, GA-based approaches have not been applied for this purpose yet. Our work was inspired by the problems encountered when using SVM for skin segmentation. Due to a very large set size, the existing methods are too time-consuming, and random selection is not effective because of the set noisiness. In the work reported here we demonstrate how a GA can be used to optimize the training set, and we present extensive experimental results which confirm that the new method is highly effective for real-world data.

1 Introduction

Support vector machines (SVM) [1] is a widely adopted classifier which has been found highly effective for a variety of pattern recognition problems. Based on a labeled training set, it determines a hyperplane that linearly separates two classes in a higher-dimensional kernel space. The hyperplane is defined by a small subset of the vectors from the entire training set, termed *support vectors* (SV). Afterwards, the hyperplane is used to classify the data of the same dimensionality as the training set data.

SVM training is a constrained quadratic programming problem of $O(n^3)$ time and $O(n^2)$ memory complexity, where n is the number of samples in the training set. This is one of the most important shortcomings of SVM, as it makes it virtually inapplicable in case of huge amounts of training samples. Therefore, some attempts have been made to refine the training sets and use only those samples, from which the support vectors are selected. Existing techniques are focused either on random selection or analysis of the data geometry.

Our contribution lies in using a genetic algorithm (GA) for selecting the relevant data from the entire available set of training samples. From the work

* This work has been supported by the Polish Ministry of Science and Higher Education under research grant no. IP2011 023071 from the Science Budget 2012–2013.

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 557–565, 2012.

reported here we conclude that in certain cases it is better to use only a small portion of the available data for training SVM. Moreover, we demonstrate that the data must be selected carefully as it has a crucial impact on the obtained classification score, and the selection process can be effectively managed using a GA. Our work was motivated by the problems related to skin detection. SVM have been already used for this purpose [2], however training set selection was not investigated there. It is worth noting that due to huge amount of available training data, proper data selection is very important in this case, which was confirmed by obtained experimental results.

The paper is organized as follows. Existing training set reduction techniques are outlined in Section 2. The details of proposed method are presented in Section 3, while the validation results are shown and discussed in Section 4. Conclusions and directions for our future work are given in Section 5.

2 Related Literature

Initial approaches towards dealing with large training sets were aimed at decomposing the optimization problem into a number of sub-problems that can be easily solved, reducing the overall training time [3]. However, for very large training sets this is insufficient, and the number of training samples must be significantly decreased. The simplest method for reducing large training sets is to select a smaller subset randomly [4]. Such an approach was the basis for reduced support vector machines (RSVM) [5]. Not only does random sampling help reduce the training time, but the classification is accelerated as well. This is because the classification time is linearly dependent on the number of SV, and generally for smaller training sets there are less SV determined.

Random sampling may be extended by analyzing the geometry of the training data in the input space. In particular, k-means clustering has been found effective here [6]. Another approach is to find crisp clusters with safety regions [7]. This method rejects the vectors inside single-class clusters, preserving those positioned at clusters' boundaries. Recently, the clustering-based approach has also been applied for one-class SVM [8]. The entire training set must be processed using these methods, which increases the computation time.

In order to achieve better performance, the clustering can be performed only in proximity of the decision boundary [9]. As the boundary is unknown before the SVM is trained, it is estimated using heterogeneity analysis based on entropy measure. Another approach to estimate the boundary is to classify the training data based on their mutual Mahalanobis distances and use only the misclassified vectors for training [10]. Mahalanobis distance-based data clustering was also studied in [11]. The points that are closest to the decision boundary are selected from every cluster. This process is well-demonstrated using artificial 2D data. Another method that operates in the kernel space rather than in the input space, applied to two-teachers-one-student problem was recently presented in [12].

There is also a group of methods which use alternative techniques to the clustering to analyze the data geometry. In [13] the convex hulls are determined

which embed the training data. Later, the vectors are selected using Hausdorff distance between the convex hulls of opposite classes. It was presented there that appropriate reduction of the training set makes it possible to achieve almost as good results as using the entire set. In [14] the points from the training set are interpreted as a graph and subject to β-skeleton algorithm. This makes it possible to reduce both training and testing time while being almost as effective as using the entire training set. Other geometry-based approaches include minimum enclosing ball [15] and smallest enclosing ball with a ring region [16].

Huge training sets can also be reduced using active learning techniques [17, 18].They operate based on a large unlabeled set, and labels for the individual samples are acquired dynamically. According to [17], these algorithms determine the points near the decision boundary, similarly to the clustering methods.

The aforementioned methods report similar conclusions. Classification accuracy for reduced training sets is comparable to that obtained using the entire training set. In some of the referenced works it is indicated that the results are slightly better than using random sampling.

3 Genetic Training Set Optimization

It must be noted that the methods which analyze the data geometry or perform clustering need to process the entire training set, and therefore their execution time depends on the total number of samples. Contrary to these methods, random sampling is applicable regardless of the number of available samples, but it is not reliable for noisy sets or when the data may be mislabeled. In such cases, it is difficult to select "good" vectors based on random drawing. In the work reported here we have successfully solved this problem using a GA to select appropriate subset of training samples. Our approach is based on the iterative random sampling, during which different draws (i.e. *individuals*) are verified, and optimal training set is selected using a GA process. This approach combines the advantages of RSVM and geometry-based methods.

A GA, firstly introduced by Holland [19], is a heuristic search approach inspired by the biological mechanism of evolution and natural selection. Encoded solutions belonging to the solution space S are called *chromosomes*. The initial population is a subset of N chromosomes, and it is successively improved during the subsequent *generations*. The chromosomes p_A and p_B are selected and recombined using the crossover operator to generate one or more offspring solutions. Selected individuals are mutated with a certain probability to avoid premature convergence of the search. The quality of each chromosome is assessed by the *fitness function* corresponding to the objective function of the problem. These with a high fitness survive and form the next generation.

3.1 Genetic Operators

For the problem reported here, a chromosome defines the content of a single subset from the entire training set $\boldsymbol{T}$, which consists of labeled samples belonging to

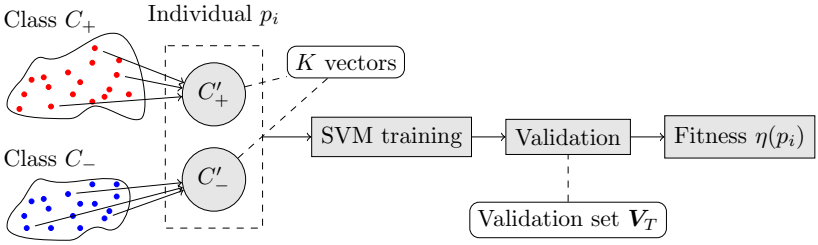

Fig. 1. Creation and validation of an individual

two classes C_+ and C_-. The chromosome's length ($2K$) is equal to the number of samples that are used for training after the reduction. The first generation of N individuals is created based on random sampling, which is illustrated in Fig. 1. From each class, K vectors are selected randomly to create a new individual p_i. This initial selection is independent from the cardinality of $\boldsymbol{T}$, which means that the genetic operations are independent from the training set size. Afterwards, SVM is trained using p_i and its fitness $\eta(p_i)$ is determined based on the classification score obtained for the validation set $\boldsymbol{V}_T$.

A set of individuals from every *i-th* generation are used for reproduction to create the $(i+1)$-*th* generation. This process is similar to generating a new individual. First, two individuals p_A and p_B create an initial training set consisting of $4K$ samples, from which $2K$ samples are selected randomly as individual p_{A+B}. Then, the new individual is subject to mutation with the probability P_m. It is performed by random changes to the training subsets of the individual. Some samples are randomly substituted with others from the entire training set $\boldsymbol{T}$. At every step it is reassured that the chromosome contains unique samples, and the same sample cannot be selected twice to the same chromosome.

3.2 Operator Strategies

The performance of a GA depends on the genetic operators including parents selection, crossover and mutation. The selection strategies address the problem of choosing two individuals from the population for recombination. The offspring solutions inherit the features of both parents p_A and p_B, thus the well-adapted individuals should be drawn from the population with a larger probability. However, recombining only the best individuals may cause saturating the population with the chromosomes of similar configurations, which in turn leads to the *diversity crisis* [20]. Four selection strategies are discussed here, namely: *high-low fit*, *AB-selection*, *truncation* and *enhanced truncation*.

1. **High-low fit**– this selection method was proposed in [21]. The population is sorted according to the fitness. The parent p_A is selected from the $c_h \cdot N$ fittest individuals, where c_h is the high-low coefficient. The parent p_B is drawn from the less-fitted part of the population. The offspring solutions are appended

to the population forming a new population of size $2N$. The N individuals with the highest fitness survive to maintain the constant population size.

2. **AB-selection**– this selection strategy was successfully used in the memetic algorithms to solve the vehicle routing problem with time windows [22, 23]. Each individual is selected for reproduction twice: first as p_A, then as p_B. If the offspring solution p_i generated for a pair of parents has higher fitness than the parent p_A then it replaces the parent p_A.
3. **Truncation.** At first, the population is sorted according to the fitness. Both parents p_A and p_B are selected from the $c_t \cdot N$ fittest individuals, where c_t is the truncation coefficient. The new population is composed of the offspring solutions generated for N pairs of parents.
4. **Enhanced truncation.** At first, the population is sorted according to the fitness. The $c_r \cdot N$ pairs of parents p_A and p_B are selected from the $c_e \cdot N$ fittest individuals, where c_r is the reproduction coefficient and c_e is the enhanced truncation coefficient. To maintain the constant population size N, the $N - c_r \cdot N$ individuals are generated randomly. The randomization simulates additional mutation for the search diversification.

The individuals of the child population are mutated with a certain probability as described in Section 3.1. In case of the AB-selection the best individuals will survive the recombination. However, they may be mutated and their fitness can decrease. Similarly, it is not guaranteed that the best chromosomes will survive for the other selection and replacement strategies. In order to keep the well-adapted individuals, the $c_c \cdot N$ best chromosomes replace a set of randomly chosen chromosomes with lower fitness, where c_c is the restoring coefficient.

The best fitness $\eta(p_b^i)$ and the average fitness $\bar{\eta}(p^i)$ in subsequent generations determine the necessity of regenerating the population. More formally, if $\eta(p_b^i) - \eta(p_b^{i-1}) < \epsilon$ for s_b consecutive steps and $\bar{\eta}(p^i) - \bar{\eta}(p^{i-1}) < \epsilon$ for s_a consecutive steps, where ϵ is the minimal improvement threshold, then the population is regenerated. The regeneration is based on copying $c_g \cdot N$ best individuals and drawing $N - c_g \cdot N$ individuals randomly, where c_g is the regeneration coefficient. The GA finishes after r regenerations.

4 Experimental Validation

The proposed method (termed GASVM) has been validated using two data sets, namely: 1) real-world data derived from ECU skin image database [24], and 2) artificial set of 2D points. ECU database consists of 4000 images coupled with binary ground-truth skin masks. The training set $\boldsymbol{T}$ was formed out of 6938255 pixels from 100 images. Every pixel was represented by a three-dimensional vector, indicating its color in YC_bC_r. Two validation sets were created, namely: $\boldsymbol{V}_T$ for evaluating the individual's fitness during the GA optimization and $\boldsymbol{V}$, which was not fed back to the GA process (all the results are presented for $\boldsymbol{V}$). The validation sets were created by sampling pixels from the remaining images. As a result, 560732 pixels were selected to every validation set. The sets are available at `http://sun.aei.polsl.pl/~mkawulok/spr`.

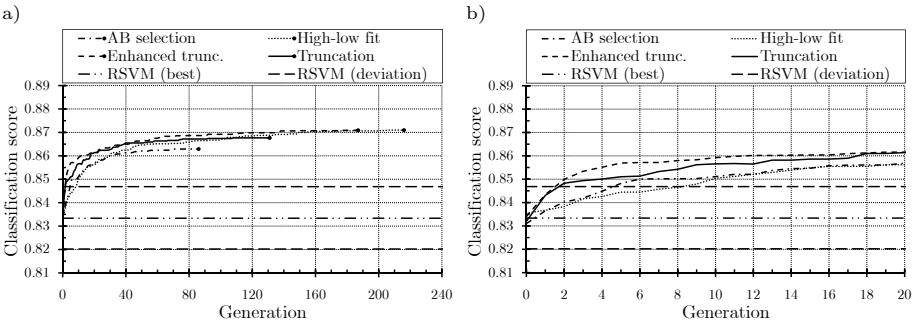

Fig. 2. Optimization process using different GA strategies compared with random sampling: a) whole process, b) first 20 generations

The GA was implemented in C++ and the experiments were performed using Intel Core i7 2.3 GHz with 16 GB RAM. We used LIBSVM [25], which is a popular SVM implementation, with RBF kernel: $K(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\|\boldsymbol{u} - \boldsymbol{v}\|^2/\sigma^2\right)$, where σ is the kernel width. SVM parameters (i.e. σ and C) were selected based on a grid search approach [25] using ranges $0.1 \leq \sigma \leq 10$ and $0.1 \leq C \leq 1000$ with a dynamic step. This simple approach was sufficient in the analyzed case and more sophisticated methods [26] were not exploited here. For skin detection we used $\sigma = 1$ and $C = 10$, and for 2D points $\sigma = 0.26$ and $C = 100$. The GA parameters were tuned experimentally in a similar manner. The following values were used: $N = 50$, $P_m = 0.3$, $c_h = 0.5$, $c_t = 0.5$, $c_r = 0.9$, $c_e = 0.2$, $c_c = 0.1$, $c_g = 0.1$, $\epsilon = 10^{-5}$, $s_a = s_b = r = 3$. In order to verify performance of RSVM [5], 20 independent tests were performed for every configuration, and within each test $N = 50$ subsets were drawn and validated to make it comparable to a single GA generation. Hence, a total number of 1000 random draws were executed to validate each setting. The best result out of each test, averaged over all the tests, is referred to as RSVM (best), while a global average result – RSVM (average). Minimal and maximal scores for all the draws are presented as RSVM (deviation) in Fig. 2 and as error bars for RSVM (best) in Fig. 3.

For each GA strategy discussed in Section 3.2, five optimization processes were run. Average maximal fitness obtained in subsequent generations for $K = 50$ samples in each class is presented in Fig. 2 for the skin data. GA strategies are compared here with RSVM. It can be seen from the graphs that after just a few generations GASVM outperforms RSVM. Enhanced truncation offers the fastest improvement, however it is the high-low fit strategy which delivers the best final score, and it has been chosen for further validation. The premature convergence of the search occurs in case of AB-selection strategy and after a relatively small number of generations the best individual cannot be further improved.

For high-low fit strategy we ran extensive tests to validate performance for various number of samples (K) in each class of the training set. In Fig. 3 our method is compared with RSVM. Error bars present maximal and minimal value.

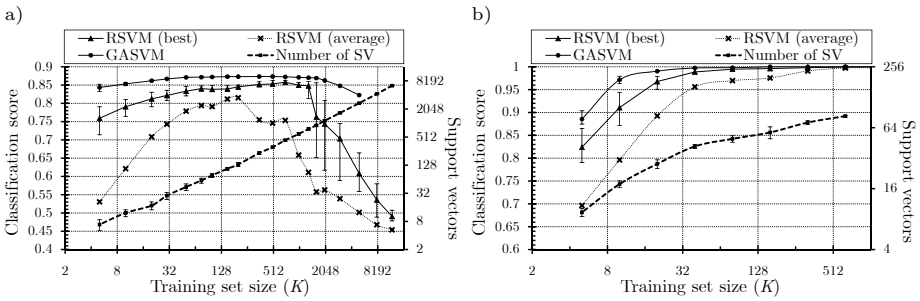

Fig. 3. GA and random sampling results depending on the training set size for skin segmentation set (a) and for artificial 2D data (b)

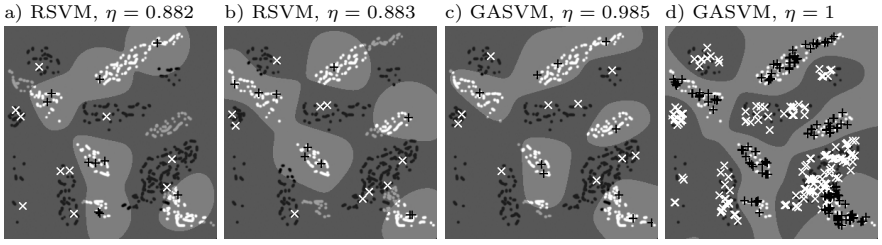

Fig. 4. Examples of training set selection using RSVM (a, b) and GASVM (c) for $K = 10$ vectors in each class, and GASVM for $K = 160$ vectors in each class (d)

For RSVM (average) the error bars were skipped as RSVM (best) indicates the maximal scores, and the minimal scores are irrelevant here. In addition, the dependence between the training set size and the number of SV is presented. For small value of K, GASVM selects definitely better training sets than those generated using random sampling, and this influences the final classification score. It is less dependent on K than RSVM, and the scores achieved in different runs are very similar. It is worth to note that the number of SV is linearly dependent on K, which induces linear dependence between K and the classification time. Theoretically, it is possible that using random sampling the same set is drawn as in case of GASVM, but for huge training sets this is little probable and has not been observed during our experiments– the best score achieved using RSVM was always worse than the worst obtained after the GA optimization.

For the skin data (Fig. 3a), the best RSVM score drops drastically after exceeding a certain threshold (ca. $K = 1500$), and the score variance increases. GASVM is more stable, but the decrease is observed as well. This can be explained by the fact that for larger sets it is hard to eliminate noisy data, which seriously affects the effectiveness. However, it is still easier to eliminate them using GASVM. We have not run GASVM for K greater than 5000 due to the required computation time. For $K = 5000$ the GA process required 4800 min to reach the stop condition, but for smaller sets the times were definitely shorter

(e.g. 80 min for $K = 30$ and 210 min for $K = 200$). Due to the SVM training complexity it would be virtually impossible to use the entire training set.

Contrary to the skin data, the artificial set of 2D points can be classified without any error using the whole set for training, which is possible due to small data set size. For smaller K, the classification error appears, however it is smaller using GASVM. For $K = 160$ GASVM eliminated the classification error, which has not been achieved using RSVM for $K < 320$. The data are visualized in Fig. 4. Black and white points indicate the vectors from the entire set, and those marked with white and black crosses show the data selected to the training set (here the colors are altered for better visualization). Also, the decision boundary is presented. It can be noticed that the selected points do not follow any specific geometric pattern as proposed in [11]. In some cases they are located near the decision boundary, but in others they are positioned in the centers of the point groups. This can be observed in particular for $K = 160$ in Fig. 4d.

5 Conclusions and Future Work

In this paper we proposed to use a genetic algorithm for selecting SVM training sets. Presented experimental results show that while in some cases our method helps reduce the training set size, which means shorter training and validation times, it also makes it possible to achieve higher classification scores for noisy or mislabeled data. Although the GA process may require many generations to converge, it is independent from the total number of available samples, which cannot be offered by existing geometry-based approaches. Furthermore, after just a few generations it manages to select better training sets than those found using random sampling, so the optimization process can be terminated earlier, if it is critical to reduce the training time.

Our ongoing research includes comparing GASVM with the geometry-based methods using benchmark data sets. This should allow us to design a memetic approach, which would combine a GA with the data structure analysis to further improve the classification results. Also, our aim is to design a parallel GA to accelerate the computations. Finally, we want to use the method for selecting the training data from unlabeled data sets.

References

1. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
2. Khan, R., Hanbury, A., Stöttinger, J., Bais, A.: Color based skin classification. Pattern Recogn. Lett. 33(2), 157–163 (2012)
3. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in kernel methods, pp. 169–184. MIT Press, USA (1999)
4. Balc'azar, J., Dai, Y., Watanabe, O.: A Random Sampling Technique for Training Support Vector Machines. In: Abe, N., Khardon, R., Zeugmann, T. (eds.) ALT 2001. LNCS (LNAI), vol. 2225, pp. 119–134. Springer, Heidelberg (2001)

5. Lee, Y.J., Huang, S.Y.: Reduced support vector machines: A statistical theory. IEEE Trans. on Neural Networks 18(1), 1–13 (2007)
6. Chien, L.J., Chang, C.C., Lee, Y.J.: Variant methods of reduced set selection for reduced support vector machines. J. Inf. Sci. Eng. 26(1), 183–196 (2010)
7. Koggalage, R., Halgamuge, S.: Reducing the number of training samples for fast support vector machine classification. Neural Information Process. Lett. and Reviews 2(3), 57–65 (2004)
8. Li, Y.: Selecting training points for one-class support vector machines. Pattern Recogn. Lett. 32(11), 1517–1522 (2011)
9. Shin, H., Cho, S.: Neighborhood property–based pattern selection for support vector machines. Neural Comput. 19(3), 816–855 (2007)
10. Abe, S., Inoue, T.: Fast Training of Support Vector Machines by Extracting Boundary Data. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 308–313. Springer, Heidelberg (2001)
11. Wang, D., Shi, L.: Selecting valuable training samples for SVMs via data structure analysis. Neurocomputing 71, 2772–2781 (2008)
12. Chang, C.C., Pao, H.K., Lee, Y.J.: An RSVM based two-teachers-one-student semi-supervised learning algorithm. Neural Networks 25, 57–69 (2012)
13. Wang, J., Neskovic, P., Cooper, L.N.: Training Data Selection for Support Vector Machines. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3610, pp. 554–564. Springer, Heidelberg (2005)
14. Zhang, W., King, I.: Locating support vectors via β-skeleton technique. In: Int. Conf. on Neural Information Process, pp. 1423–1427 (2002)
15. Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast SVM training on very large data sets. J. of Machine Learning Research 6, 363–392 (2005)
16. Zeng, Z.Q., Xu, H.R., Xie, Y.Q., Gao, J.: A geometric approach to train SVM on very large data sets. Intell. System and Knowledge Eng. 1, 991–996 (2008)
17. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: 17th Int. Conf. on Machine Learning, pp. 839–846. Morgan Kaufmann Publishers Inc., USA (2000)
18. Musicant, D.R., Feinberg, A.: Active set support vector regression. IEEE Trans. on Neural Networks 15(2), 268–275 (2004)
19. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press (1975)
20. Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., Price, K.V. (eds.): New ideas in optimization. McGraw-Hill Ltd., UK (1999)
21. Elamin, E.E.A.: A proposed genetic algorithm selection method. In: 1st National Symposium, NITS (2006)
22. Nagata, Y., Bräysy, O., Dullaert, W.: A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Computers & OR 37(4), 724–737 (2010)
23. Nalepa, J., Czech, Z.J.: A parallel heuristic algorithm to solve the vehicle routing problem with time windows. Studia Informatica 33(1), 91–106 (2012)
24. Phung, S.L., Chai, D., Bouzerdoum, A.: Adaptive skin segmentation in color images. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal, pp. 353–356 (2003)
25. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Trans. on Intell. Systems and Technology 2, 27:1–27:27 (2011)
26. Staelin, C.: Parameter selection for support vector machines. Technical Report HPL-2002-354. HP Laboratories, Israel (2002)