SETSCI Conference Proceedings 15,18-21, 2023

Cognitive Models and Artificial Intelligence Conference

26-28 October 2023, Ankara, Türkiye

https://doi.org/10.36287/setsci.6.1.010

Improving Computational Performance of Least Squares Multiple Birth Support Vector Machines with *k*-Means Clustering

Güvenç Arslan[1*] and Nisa Nur Bülbül[2]

[1]*Department of Statistics, Kırıkkale University, Kırıkkale, Turkey*
[2]*Graduate School of Natural and Applied Sciences, Kırıkkale University, Kırıkkale, Turkey*
[*]*(guvenc.arslan@kku.edu.tr)*

Abstract – Multiple Birth Support Vector Machines (MB-SVM) were introduced as a powerful extension of SVM. Although the basic idea is like SVM, optimal non-parallel hyperplanes are used for each class category. One may find different implementations of this approach in the literature. One of these is the Least Squares MB-SVM. An appealing property of this implementation is that an analytical solution is obtained that is used for classification. On the other hand, this solution involves matrix computations of sizes depending on the number of attributes and the size of the data set. In this study, we propose to use the k-means clustering algorithm before applying the Least Squares MB-SVM algorithm to improve the computational performance of MB-SVM. The preliminary results with the Iris data set indicate that by using only a small number of examples obtained from the k-means clusters, comparable performance can be obtained with the LS-MB-SVM method.

Keywords – support vector machine, multiple birth, k-means

I. Introduction

Support vector machines (SVM) are one of the supervised learning algorithms used with success in various applications. They were introduced in the framework of statistical learning by Vapnik and Chervonenkis [1]. The basic idea in SVM classification is to obtain the optimal hyperplane for classification by finding the maximum margin between examples at the boundaries of the examples in each class (..).

In recent years one may find various extensions and variations of the basic idea used in SVMs [2]. One of these variations is the Twin Support – SVM (Twin-SVM) introduced by Jayadeva et al. [3]. In contrast to standard SVMs, non-parallel hyperplanes are used in Twin-SVMs. Multiple Birth – SVM introduced by Yang et al. [4] may be considered as an extension of the basic idea in Twin-SVM to multiple classes.

MB-SVMs have lower computational complexity with improved classification accuracy [4]. On the other hand, there are various implementations of the MB-SVM approach. One of these is the Least Squares - MB-SVM (LS-MB-SVM) introduced by [5]. One advantage of this approach is that the solution for the optimal hyperplane is expressed analytically. On the other hand, this solution involves basic matrix computations and inverses of matrices whose dimensions depend on the dimensions of the data set. This may result in some computational problems especially for moderately large data sets.

The main goal of this study is to reduce the computational complexity of the LS-MB-SVM while keeping classification performance. As noted before, the solution in LS-MB-SVM is expressed by using large matrix products and inverses of matrices, which may result in some computational problems. Thus, it is expected that applying the k-means clustering algorithm before using LS-MB-SVM will considerably reduce the computational complexity of this algorithm.

II. Materials and Method

Assuming that there are t class categories in the classification problem, the set of class categories will be denoted by $\mathcal{C}_t = \{k_1, k_2, \dots, k_t\}$. Thus, the data set for a classification problem can be expressed as

$$\mathcal{D} = \{(\boldsymbol{x}_j, y_j): \boldsymbol{x}_j \in \mathbb{R}^n, y_j \in \mathcal{C}_t \; ; \; j = 1,2,\dots,m\}, \quad (1)$$

where m denotes the number of examples. Since in the MB-SVM approach there is an optimization problem for each class category it is more convenient to express this data set with respect to each class separately. Therefore, for $i = 1, 2, \dots, t$ the examples corresponding to the class category k_i can be described as

$$\mathcal{D}_i = \left\{\left(\boldsymbol{x}_j^{(i)}, k_i\right): \boldsymbol{x}_j^{(i)} \in \mathbb{R}^n \text{ ve } j = 1,2,\dots,m_i\right\}, \quad (2)$$

where m_i denotes the number of examples in class k_i. With this notation it follows that

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \dots \cup \mathcal{D}_t, \quad (3)$$

where $m = m_1 + m_2 + \dots + m_t$.

Although there are different implementations of the MB-SVM approach we used the approach introduced in [5]. In this approach the solution is obtained by using the least squares method yielding an analytical solution of the classification problem.

A. Least Squares Multiple Birth Support Vector Machines

MB-SVMs were introduced by Yang et al. [4], which may be considered as a generalization of Twin-SVM introduced by Jayadeva et al. [3]. An important property of MB-SVM is that one optimization problem is expressed for each class to find the corresponding optimum hyperplane. On the other hand, in contrast to standard SVMs, examples are classified with these non-parallel hyperplanes in MB-SVMs.

Although there are different implementations of the MB-SVM approach we used the approach in [5]. In this approach the optimization problem for classifying examples in class k_i is expressed as

$$\min_{\boldsymbol{w}_i, b_i, \boldsymbol{\xi}_i} \frac{1}{2}\|B_i\boldsymbol{w}_i + \boldsymbol{e}_i b_i\|_2^2 + \frac{c_i}{2}\boldsymbol{\xi}_i^T\boldsymbol{\xi}_i + \frac{v_i}{2}(\|\boldsymbol{w}_i\|^2 + b_i^2) \quad (4)$$
$$s.t.\ (A_i\boldsymbol{w}_i + \boldsymbol{e}_i' b_i) + \boldsymbol{\xi}_i = \boldsymbol{e}_i',\ \boldsymbol{\xi}_i \geq 0$$

Here, A_i represents the matrix of the examples in class k_i without the class label. Similarly, matrix B_i represents all examples (without any class label) in the data set except examples belonging to class k_i.

Using the least squares method, the solution for the optimal hyperplane parameters is obtained as

$$\boldsymbol{u}_i = \begin{bmatrix}\boldsymbol{w}_i \\ b_i\end{bmatrix} = c_i[H_i^T H_i + v_i I_i + c_i G_i^T G_i]^{-1} G_i^T \boldsymbol{e}_i', \quad (5)$$

where $H_i = [B_i\ \boldsymbol{e}_i]$ and $G_i = [A_i\ \boldsymbol{e}_i']$. A new example $\boldsymbol{x}_0$ is classified by using

$$j = \arg\max_{i=1,2,\dots t} \frac{|\boldsymbol{w}_i^T x_0 + b_i|}{\|\boldsymbol{w}_i\|}, \quad (6)$$

and assigning it to class k_j.

It should be noted that in the setting above it is assumed that the examples are almost linearly separable. If not, appropriate kernels can be used to transform the data set into a higher dimensional space to improve classification performance. In this case, the classification of a new example $\boldsymbol{x}_0$ is determined by

$$j = \arg\max_{i=1,2,\dots t} \frac{|K(x_0^T, C^T)\widetilde{\boldsymbol{w}}_i + b_i|}{\sqrt{\widetilde{\boldsymbol{w}}_i^T K(C^T, C^T)\widetilde{\boldsymbol{w}}_i}}, \quad (7)$$

where $K(.,.)$ represents the kernel function to be used. For more details about the optimization problem and its solution the reader is referred to [5].

B. Least Squares Multiple Birth Support Vector Machines with k-Means

In Yang et al. [4] it is noted that the construction of different optimization problems for each class results in lower computational complexity and that it is expected to be faster than existing multi-class SVMs. On the other hand, as can be seen from equations (5) - (7) in LS-MB-SVM the solution involves computation of matrix multiplications depending on the size of the data set. For moderately large data sets and the increased number of parameters to be selected for high classification performance in LS-MB-SVM, this can negatively affect the training time and performance of the algorithm. Therefore, it is expected that using the *k*-means clustering algorithm before applying LS-MB-SVM, as introduced by Wang et al. [6] for SVMs, will reduce computational complexity and reduce training time.

The basic idea in MB-SVM is to find the optimal hyperplane in a particular class that is farthest to the examples in the remaining classes. It is expected that only little information is lost, when using centers of clusters after *k*-means clustering, to construct the optimal hyperplane. This approach will be denoted by *k*-LS-MB-SVM. The basic steps of the proposed algorithm are as follows:

1. Determine the number of clusters (k), the parameters ($c_1, \dots, c_t,\ v_1, \dots, v_t$) and appropriate kernel function $K(.,.)$ for LS-MB-SVM.
2. Apply *k*-means clustering to the training set.
3. Construct the reduced training set consisting of cluster centers and its corresponding class labels.
4. Apply LS-MB-SVM to the reduced training set to obtain the classifier.

Figure 1 shows how the training data set is reduced by *k*-means from 14 examples to 5 examples (denoted by the larger red points) for a small data set. The line represents the optimal hyperplane obtained by the reduced data set for examples in the class at the top left corner. The size of reduction depends on the number of clusters chosen. In this example, the training data set is reduced with a ratio of 14/5=2.8.

Different strategies may be applied to use the information about classes in each cluster. In this study, we have used a simple strategy: assign the majority class label as the class label for each cluster center.

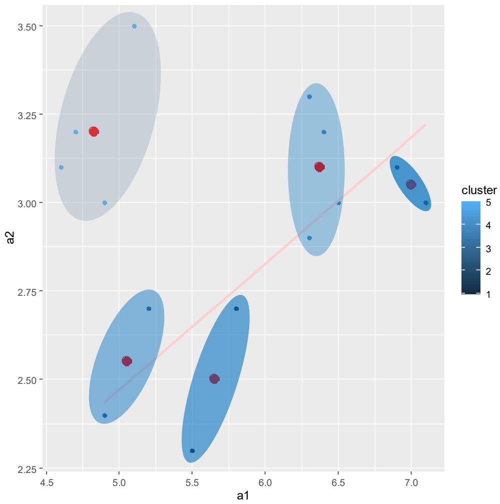

Fig. 1 Example for clustering with *k*-means

III. Results

The proposed approach was applied to the IRIS data set [7] to demonstrate how it reduces the training data set while preserving comparable classification performance. The data set is split into training and test sets with %30 as the test data set. This is repeated 5 times to take into consideration the variations of the splitting. Since for each run of the *k*-means algorithm different clusters may be obtained *k*-means clustering is applied to each of these training data sets 10 times for each value of *k*.

The implementation of the proposed algorithm was done by using the R Software [8]. For SVM calculations the e1071 package [9] was used whereas for the LS-MB-SVM calculations we developed our own implementation in R with the linear and radial basis kernels.

Figure 2 shows how the test accuracies change with respect to the number of clusters for *k*=3,5,7,11,13 (*a*=3, *b*=5, … *f*=13). The results clearly indicate that for the radial basis kernel (*k*2) better classification results are obtained than for the linear kernel (*k*1).

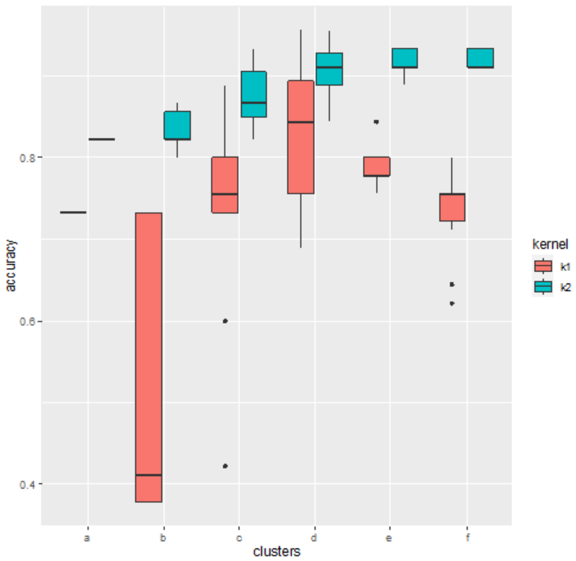

Fig. 2 Accuracies for test set with k values of 3,5,7,9,11,13

Figure 3 shows the average test accuracies for 10 trials with each k value for one test data set. These calculations are repeated for each different partition of train and test sets. The resulting average accuracies for each partition are compared with SVM and MB-SVM accuracies in Figure 4.

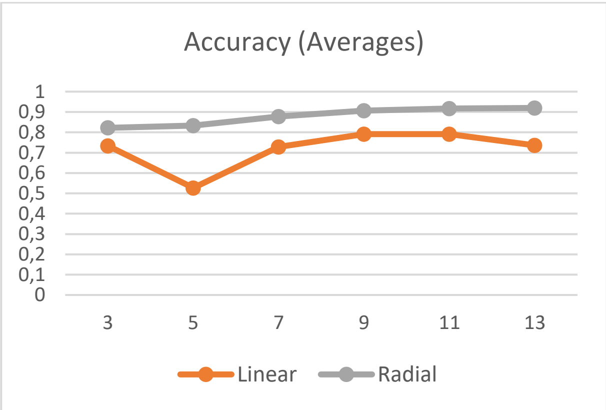

Fig. 3 Average test accuracies for k values of 3,5,7,9,11,13

Figure 4 shows how the test accuracies vary according to three different implementations of the SVM approach: SVM, MB-SVM, k-LS-MB-SVM. The accuracies for the k-LS-MB-SVM method are obtained by taking the average of 10 repeated calculations with k=11.

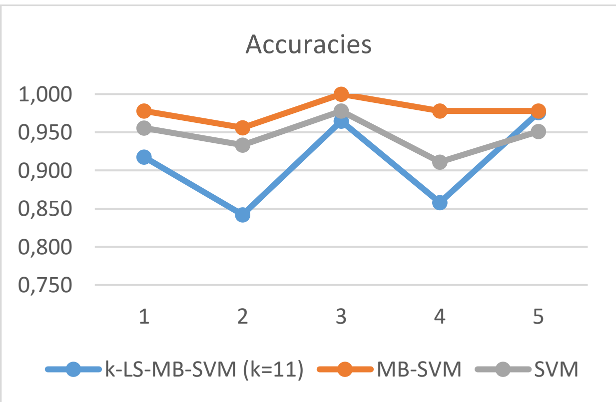

Fig. 4 Average test accuracies for three different SVM implementations

From Figure 4 it is clear that the best performance for this data set is obtained with the MB-SVM method. The accuracies for the SVM method are slightly smaller than the accuracies for the MB-SVM method. We note that the training data size for both methods is 105 whereas for the k-LS-MB-SVM method it consists only of 11 examples. Therefore, the k-LS-MB-SVM method yields comparable accuracies with only a small number of examples obtained from the clusters.

IV. DISCUSSION

Although k-means clustering has been applied with SVM in several studies, (see for example [6]) to the best of our knowledge there is no study in the literature that uses the same approach with MB-SVMs.

The preliminary findings of this study indicate that using k-means with the LS-MB-SVM method indeed helps to improve the computational performance of this method. In this study, we selected a fixed value for each train/test splitting. The results indicated that overall, the best choice (with respect to test accuracies) for the considered k-values was 11. We note here that in general increasing the k value also increases classification accuracies up to some threshold value. Therefore, using larger k values even higher classification accuracies for the k-LS-MB-SVM approach can be expected than the results given in Figure 4.

In this study, we only used a limited number of k-values because of time constraints. An alternative could be to choose a wider range of k values and selecting different k values for different training data sets. The results demonstrate that by using only a small number of examples obtained from the k-means clustering, comparable performance can be obtained. In addition, the time needed for classification is also considerable reduced with the introduced approach.

Another advantage of the proposed approach is that fine-tuning of the additional parameters resulting from the LS-MB-SVM approach can be addressed more effectively. We only used a fixed set of values for these parameters ($c_1 = c_2 = c_t = 0.5,\ v_1 = v_2 = v_3 = 1.0$). Fine-tuning of these and kernel function parameters may further improve classification accuracies.

V. CONCLUSION

The main objective of this study was to investigate the use of k-means clustering algorithm before applying LS-MB-SVM, as introduced by Wang et al. [6] for SVMs. The preliminary findings indicate that with appropriately chosen k values and cluster representatives as training examples the computational complexity of LS-MB-SVM can be reduced with comparable classification performance. Although the results of this study need to be verified with other data sets our preliminary findings show that the proposed approach is promising in improving the computational performance of the LS-MB-SVM method.

REFERENCES

[1] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition*, Nauka, Moscow, 1974.

[2] G. Arslan, U. Madran, and D. Soyoğlu. "An Algebraic Approach to Clustering and Classification with Support Vector Machines," *Mathematics*, 10.1: 128, Jan. 2022.

[3] Jayadeva, R. Khemchandani, and S. Chandra, "Twin Support Vector Machines for Pattern Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 29 (5), pp. 905–910, May 2007.

[4] Z.-X. Yang, Y.-H. Shao, and X.-S Zhang, "Multiple birth support vector machine for multi-class classification" *Neural Comput & Applic*, 22 (Suppl 1), 153-161, 2013. doi: 10.1007/s00521-012-1108-x

[5] S.-G. Chen and X.-J Wu, "Multiple birth least squares support vector machine for multi-class classification*" Int. J. Mach. Learn. & Cyber*. 8, 1731-1742, 2017. doi: 10.1007/s13042-016-0554-7

[6] J. Wang, X. Wu, and C. Zhang, "Support vector machines based on K-means clustering for real-time business intelligence systems", *International Journal of Business Intelligence and Data Mining*, 1(1), 54-64, 2005.

[7] R. A. Fisher (1988) Iris. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.

[8] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

[9] Dimitriadou, E.; Hornik, K.; Leisch, F.; Chang, C.-C.; Lin, C.-C. Package 'e1071'. R Software Package. Available online: https://cran.r-project.org/web/packages/e1071 (accessed on 14 August 2021).