Journal of New Theory
ISSN: 2149-1402

34 (2021) 64-71
*Journal of New Theory*
https://dergipark.org.tr/en/pub/jnt
Open Access

# The Effects of Kernel Functions and Optimal Hyperparameter Selection on Support Vector Machines

**Aslı Yaman[1], Mehmet Ali Cengiz[2]**

**Article History**
Received: 08 Feb 2021
Accepted: 10 Mar 2021
Published: 30 Mar 2021
Research Article

**Abstract** – Support Vector Machine (SVM) is a supervised machine learning method used for classification and regression. It is based on the Vapnik-Chervonenkis (VC) theory and Structural Risk Minimization (SRM) principle. Thanks to its strong theoretical background, SVM exhibits a high performance compared to many other machine learning methods. The selection of hyperparameters and the kernel functions is an important task in the presence of SVM problems. In this study, the effect of tuning hyperparameters and sample size for the kernel functions on SVM classification accuracy was investigated. For this, UCI datasets of different sizes and with different correlations were simulated. Grid search and 10-fold Cross-Validation methods were used to tune the hyperparameters. Then, SVM classification process was performed using three kernel functions, and classification accuracy values were examined.

***Keywords*** – *Support vector machines, kernel function, tune parameter*

***Mathematics Subject Classification (2020)*** – 62H30, 62P99

## 1. Introduction

The first mentions on Support Vector Machine (SVM) were made by Vapnik in 1979. However, after the presentation at the conference named COLT (Conference on Computational Learning Theory), held in America in 1992 [1], the use of SVM became widespread. Then, it was officially introduced by Vapnik in 1995 [2].

The SVM theory is based on the idea of Vapnik-Chervonenkis (VC) theory and Structural Risk Minimization (SRM). The VC theory is a subbranch of statistical learning theory. The main goal in learning problems is to reach the most accurate results with the minimum error. For this, the expected risk is desired to be minimum. The basic idea in SRM principle and VC theory is to select the model with the correct level of complexity to minimize the expected risk or generalization error among many models. The SRM principle aims to minimize the upper bound of the expected risk. For a function with distribution, the SRM principle converges to the optimal solution. SVM tries to keep both experimental risk and VC dimension to a minimum so that the expected risk reaches the minimum [3].

SVM aims to classify the observations most accurately by finding the optimal separating hyperplane between two or more classes. It is used in linear and non-linear classification and regression problems. Datasets in which training data cannot be separated linearly are transferred to a higher dimensional feature space using mapping functions. The dataset mapped to the feature space can be linearly separated using kernel functions

[1]asliyamann@gmail.com (Corresponding Author); [2]macengiz@omu.edu.tr
[1,2]Department of Statistics, Faculty of Arts and Sciences, Ondokuz Mayıs University, Samsun, Turkey

[4]. In feature space, SVM tries to solve the quadratic optimization problem to find the optimal separating hyperplane.

SVM is used in many different domains: pattern recognition (handwriting [5], face [6], speech [7], emotion [8], disease diagnosis [9], treatment success [10], time series [11], criminology [12], stock market prediction [13], etc.).

This study aims to introduce the two-class SVM classification theory and examine the effects of sample size and optimal hyperparameter selection on classification accuracy. Besides, determining the optimal values of the hyperparameters of kernel functions has a significant impact on SVM results. For tuning the hyperparameters, many algorithms have been proposed, such as grid search, random search, Bayesian optimization, simulated annealing, particle swarm optimization, genetic algorithm, etc. [14]. In this study, we used a grid search CV algorithm to tune hyperparameters. After tuning the hyperparameters, the SVM classification results were examined on the simulated dataset with different scenarios.

The rest of the paper is organized as follows. A brief review of the theory of SVM is described in Section 2. The experiments are presented in Section 3. Results are given in Section 4, and we conclude the paper with a summary of results by emphasizing the importance of this study and mentioning some viable future work.

## 2. Support Vector Machines

The theory of SVMs in classification problems is given in this section [15,16]. SVMs are used to optimally separate dataset belonging multiple classes by specifying a hyperplane. With linear SVM, the dataset can be separated completely (hard margin) or partially (soft margin), and the dataset cannot be separated linearly in any way with non-linear SVM.

### 2.1. Linear Support Vector Machines

Let $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), x \in R^n$ be the training dataset for SVM with separable two-class labels such as $y \in \{+1, -1\}$. The main purpose of SVM is to find the most suitable separating hyperplane that will enable to classify training observations correctly. Hyperplane represents a separating surface in a multidimensional space. There can be thousands of different hyperplanes between two classes, so the most suitable (optimal) hyperplane must be found for strong classification accuracy and better generalization performance.

To find the optimal separating hyperplane, it is necessary to determine the distance between training observations with different two-class labels called Margin. The maximum margin classifier will give the optimal separating hyperplane. Separating hyperplanes are formulated as in Equation 1,

$$D(x) = (w.z) + b = 0 \tag{1}$$

and should provide Equation 2 for both classes.

$$y_i[(w.z) + b] \geq 1, i = 1, \dots, n \tag{2}$$

The distance between hyperplane and origin is

$$d = \frac{|b|}{\|w\|} \tag{3}$$

in which $b$ and $w$ are the parameters of the optimal hyperplane. Here, $|\,.\,|$ is the absolute value and $\|\,.\,\|$ is the Euclidean norm of a vector. Assume two hyperplanes $(+d, -d)$ for two-class label $(+1, -1)$. Thus, the margin is computed as

$$\text{margin} = d_+ - d_- = \frac{|1-b|}{\|w\|} - \frac{|-1-b|}{\|w\|} = \frac{2}{\|w\|} \tag{4}$$

To maximize this margin (hard margin), the norm of $w$ is minimized. Hence, the primal form of the optimization problem obtained for the maximum margin classifier or, in other words, the optimal separating hyperplane is as follows:

$$\text{minimize } \frac{1}{2}\|w\|^2$$

$$\text{subject to } y_i(w.z_i + b) \geq +1, i = 1,..,n \tag{5}$$

The optimal hyperplane problem is a classical optimization problem and can be solved by the Lagrangian multiplier method and Krush Kuhn Tucker (KKT) conditions, so the problem transforms into the dual form, and the dual form of the problem is solved as in Equation 6,

$$\text{maximize } W(x) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j z_i.z_j$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0, i = 1, \dots, n \tag{6}$$

The KKT theorem is important in the theory of SVM. According to KKT conditions, there are two different situations for $\alpha_i(y_i(w.z_i + b) - 1) = 0$, which are the correctly classified features outside of hyperplanes ($\alpha_i = 0$) and the correctly classified features located on hyperplanes ($\alpha_i \geq 0$), called *support vectors*. The structure of an SVM is shown in Figure 1.

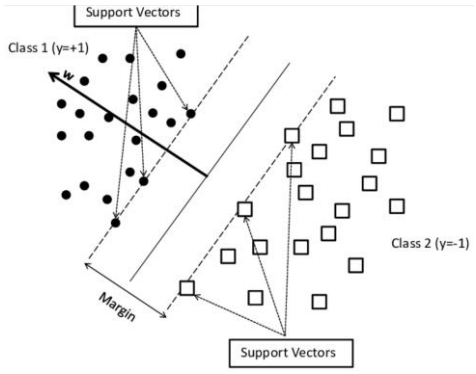


**Fig 1:** The structure of an SVM

## 2.2. Non-Linear Support Vector Machines

The previous sections mentioned that the dataset is completely and linearly separable (hard margin). Moreover, when the dataset is partially non-separable (soft margin), slack variables (ξ) are added to Equation 2, and the computations are performed as in the hard margin optimization. Then, separating hyperplane for the partially non-separable dataset is found as in Equation 7,

$$y_i[(w.z) + b] \geq 1 - \xi_i, \ i = 1, \dots, n \tag{7}$$

The optimal hyperplane for the partially non-separable case is obtained from Equation 8,

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i(w.z_i + b) \geq 1 - \xi_i, \ \ i = 1,..,n, \ \xi_i \geq 0 \tag{8}$$

in which the regularization parameter C is constant.

Non-linear SVM classifier is used in cases where training observations cannot be separated by a linear decision surface. Generally, datasets cannot be separated linearly in real analysis. For such problems, mapping functions are used to transform the input space in which training observations cannot be separated linearly into a higher dimensional feature space where observations can be linearly separated [17]. To access this aim, kernel functions are used because the transition to a higher dimensional space with mapping functions and processing with dot products in this space is computationally difficult and time-consuming. Kernel function $K(.,.)$ is given in Equation 9,

$$K(x_i, x_j) = z_i.z_j = \varphi(x_i).\varphi(x_j) \tag{9}$$

The function satisfies Mercer's theorem. The most known kernel functions are linear, radial basis function, polynomial, sigmoid, dot product, and two-layer neural network kernel [18]. Some kernel function algorithms are given in Table 1.

**Table 1.** Kernel functions

| Kernel Functions | Algorithms |
|---|---|
| Linear | $K(u', v) = u'v$ |
| Polynomial | $K(u', v) = (u'v + 1)^d$ |
| Radial Basis Function | $K(u', v) = \exp(-\|u - v\|^2/\sigma^2)$ |

The non-linear separating hyperplane can be found as

$$\text{maximize } W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j K(x_i, x_j)$$
$$\text{subject to } \sum_{i=1}^{n} y_i\alpha_i = 0, \quad 0 \le \alpha_i \le C, i = 1, \dots, n \tag{10}$$

and the decision function is as in Equation 11,

$$f(x) = \text{sign}(w.z + b) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b) \tag{11}$$

## 3. Experiments

It is aimed to determine the optimal hyperparameters of the kernel function to be used in SVM classification and to examine the effect of these optimal values on the classification accuracy.

For this purpose, firstly datasets with standard normal distribution according to different correlation levels and different sample sizes were created by using the "MASS" package in the R. Correlation levels for simulated datasets were determined as 0.25,0.50,0.75 and sample sizes were determined as 20,50,100 and 200, respectively. The number of features has been kept constant as 2.

For these datasets, optimal hyperparameter selection was performed according to kernel functions with Grid search and 10-fold Cross Validation (CV) methods with 30 iterations. The intervals for the hyperparameters determined as in Table 2.

**Table 2.** Hyperparameter searching intervals setting

| | Hyperparameters | | | |
|---|---|---|---|---|
| Kernel Function | C | Sigma | Degree | Scale |
| Linear | {1, 2, … ,20} | - | - | - |
| Radial Basis Function | {1, 2, … ,20} | {0.1, 0.6, 1.1, … ,10} | - | - |
| Polynomial | {1, 2, … ,20} | - | {1,2,3} | $\{10^{-3}, 10^{-2}, \dots, 10^{1}\}$ |

2 class-SVM classification process was carried out by using "e1071" packages. Then, datasets for 3 different kernel functions (linear, polynomial and radial basis function) were analysed for optimal hyperparameter values and the kernel functions with the highest test accuracy were examined according to the obtained test prediction results.

## 4. Results

Firstly, optimal hyperparameters were selected according to 3 different correlation levels and SVM classifier performances were obtained for 3 different kernel functions when the number of observations was 20. Results were given in Table 3.

**Table 3.** Optimal values for kernel hyperparameters and classification accuracies when sample size was 20

| Sample size | Kernel function | Correlation levels | Optimal hyperparameter values | Classification accuracy% |
|---|---|---|---|---|
| 20 | Linear | 0.25 | C=1 | 0.60 |
| | | 0.50 | C=1 | 0.60 |
| | | 0.75 | C=1 | 0.75 |
| 20 | Polynomial | 0.25 | d=2, s=1, C=1 | 0.60 |
| | | 0.50 | d=2, s=1, C=1 | 0.60 |
| | | 0.75 | d=3, s=1, C=1 | 0.74 |
| 20 | RBF | 0.25 | C=2, $\sigma$=2.5 | 0.60 |
| | | 0.50 | C=2, $\sigma$=2.6 | 0.60 |
| | | 0.75 | C= 2, $\sigma$=9.6 | 0.75 |

In Table 3, it was observed that when the sample size was 20, optimal hyperparameter values got almost the same values according to different correlation levels. Similar results were obtained for 3 kernel functions of SVM classification accuracies. While the sample size was 20, the most accurate classification percentage was obtained when the correlation level was 0.75 for all 3 kernel functions. While the sample size was small, it was concluded that the classification accuracy varied according to the correlation levels, not the kernel functions.

Secondly, for the number of observations 50, optimal hyperparameters were selected according to 3 different correlation levels and test accuracy percentages were obtained for 3 kernel functions according to the parameters. Results were given in Table 4.

**Table 4.** Optimal values for kernel hyperparameters and classification accuracies when sample size was 50

| Sample size | Kernel function | Correlation levels | Optimal hyperparameter values | Classification accuracy% |
|---|---|---|---|---|
| 50 | Linear | 0.25 | C=5 | 0.89 |
| | | 0.50 | C=6 | 0.87 |
| | | 0.75 | C=3 | 0.88 |
| 50 | Polynomial | 0.25 | d=2, s=1, C=18 | 0.98 |
| | | 0.50 | d=2, s=1, C=17 | 0.95 |
| | | 0.75 | d=3, s=1, C=18 | 0.99 |
| 50 | RBF | 0.25 | C=2, ,$\sigma$=1.6 | 0.99 |
| | | 0.50 | C=19, $\sigma$=1.1 | 0.99 |
| | | 0.75 | C=1, $\sigma$=0.6 | 0.99 |

In Table 4, while the number of observations 50 with different correlation levels, it was seen that although the optimal hyperparameter values were different, more accurate classification percentages were obtained with the polynomial and RBF kernel functions.

The optimal hyperparameters were selected according to 3 different correlation levels and test accuracy percentages were obtained for 3 kernel functions according to the parameters for the number of observations 100. Results were given in Table 5.

**Table 5.** Optimal values for kernel hyperparameters and classification accuracies when sample size was 100

| Sample size | Kernel function | Correlation levels | Optimal hyperparameter values | Classification accuracy% |
|---|---|---|---|---|
| 100 | Linear | 0.25 | C=3 | 0.76 |
| | | 0.50 | C=4 | 0.78 |
| | | 0.75 | C=1 | 0.76 |
| 100 | Polynomial | 0.25 | d=2, s=1, C=8 | 0.96 |
| | | 0.50 | d=2, s=1, C=8 | 0.92 |
| | | 0.75 | d=2, s=1, C=8 | 0.96 |
| 100 | RBF | 0.25 | C=12, $\sigma$ =1.1 | 0.92 |
| | | 0.50 | C=20, $\sigma$ =0.6 | 0.92 |
| | | 0.75 | C=20, $\sigma$ =1.1 | 0.88 |

It is seen in the Table 5 that the highest classification accuracy values were obtained with the polynomial kernel. For the polynomial kernel, the result is that the optimal parameter values are the same despite different correlation levels. The same analyses were performed for the number of observations 200, and the results were obtained as in Table 6.

**Table 6.** Optimal values for kernel hyperparameters and classification accuracies when sample size was 200

| Sample size | Kernel function | Correlation levels | Optimal hyperparameter values | Classification accuracy% |
|---|---|---|---|---|
| 200 | Linear | 0.25 | C=2 | 0.92 |
| | | 0.50 | C=5 | 0.90 |
| | | 0.75 | C=2 | 0.90 |
| 200 | Polynomial | 0.25 | d=2, s=1, C=20 | 0.98 |
| | | 0.50 | d=2, s=1, C=12 | 0.99 |
| | | 0.75 | d=2, s=1, C=20 | 0.99 |
| 200 | RBF | 0.25 | C=3, $\sigma$=0.6 | 0.98 |
| | | 0.50 | C=1, $\sigma$=1.6 | 0.98 |
| | | 0.75 | C=20, $\sigma$=1.1 | 0.99 |

In Table 6, the highest accuracy values are obtained with polynomial and radial kernel. Although the optimal hyperparameter values are different in the radial kernel according to different correlation levels, it is seen that similar results are obtained for the appropriate hyperparameter values in the polynomial kernel. Furthermore, Haberman's Survival dataset from the University of California Irvine (UCI) repository [19] was used in the experiments. It was described in Table 6 with the number of classes, instances, and features.

**Table 6.** Information about UCI dataset

| Dataset | Number of Classes | Number of Instances | Number of Features |
|---|---|---|---|
| Haberman's Survival | 2 | 306 | 3 |

The analysis on the simulated datasets were also performed for the Haberman's Survival UCI dataset. The intervals specified in Table 2 were used to obtain the optimal hyperparameters. The optimal hyperparameter selection was performed according to kernel functions with Grid search and 10-fold CV methods with 30 iterations. After determining the optimal hyperparameter values, the SVM classification process was performed. The results were given in Table 7.

**Table 7.** SVM classification results of Haberman's Survival dataset

| Kernel function | Optimal hyperparameter values | Classification accuracy % |
|---|---|---|
| Linear | C=5 | 0.7368 |
| Polynomial | C=12, d=3, s=0.1 | 0.7337 |
| RBF | C=8, $\sigma$=0.1 | 0.7763 |

In Table 7, the highest SVM classification accuracy for the UCI dataset was achieved with the RBF kernel function with C=8 and $\sigma$=0.1 values. In addition, the graphical representation of obtaining optimal hyperparameter values for 3 kernel functions was given in Figure 2.

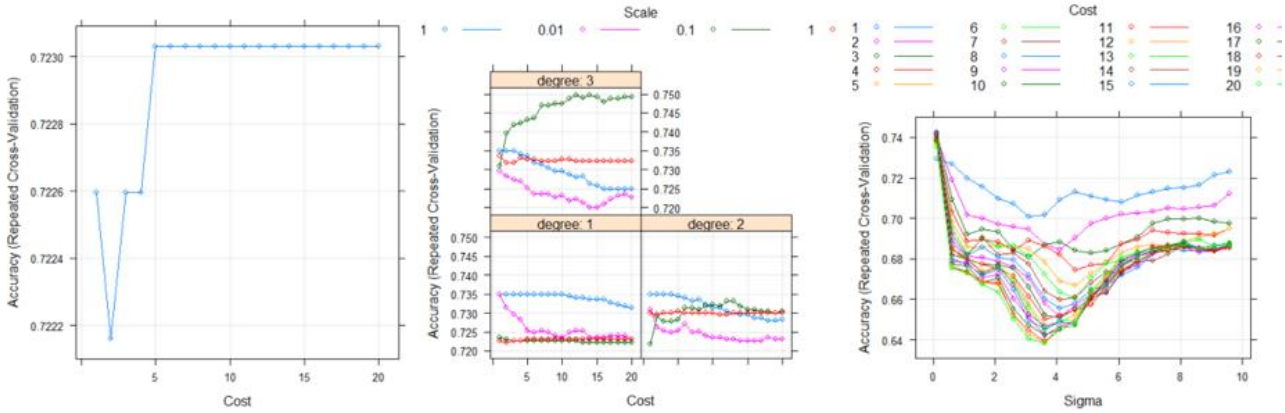


**Fig 2:** Optimal hyperparameters search for Haberman's Survival dataset

In Figure 2, the graphical representation of the values of the Cost parameter for the linear kernel according to SVM classification accuracy percentages was given on the left. It is seen that the highest accuracy values are obtained when the C parameter was greater than or equal to 5. In the middle, graphical representation of polynomial kernel parameters according to classification accuracy was given. It was observed that there is a relationship between C, scale and degree parameters and they have different effects on classification accuracy in different situations. On the right, a graphical representation of the values for the C and sigma parameters of the RBF kernel function was given.

## 5. Conclusion

The present study was focused on tuning the hyperparameters in SVM classification problems. Grid search and ten-fold CV methods were used to obtain optimal values of hyperparameters according to kernel functions with different sample sizes and correlation levels. Then, the classification accuracy values were examined by performing SVM classification.

SVM is a powerful method developed for classification and regression problems. Although grid search and five- or ten-fold CV yields successful results in finding the optimal value for hyperparameters, it may still pose a risk to determine the intervals for these parameters by the users. Therefore, in future studies, developing new approaches in addition to existing methods on the automatic selection of hyperparameter and kernel function will save time in analyses and produce more reliable results.

## Conflict of Interest

The authors declare no conflict of interest.

## References

[1] B. E. Boser, I. M. Guyon, V. N. Vapnik, *A Training Algorithm for Optimal Margin Classifiers*, In Proceedings of the Fifth Annual Workshop on Computational Learning Theory (1992) 144-152.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.

[3] V. Jakkula, *Tutorial on Support Vector Machine (SVM)*, School of EECS, Washington State University, (2006) 37.

[4] S. Han, C. Qubo, H. Meng, *Parameter Selection in SVM with RBF Kernel Function*, In World Automation Congress (2012) 1-4 Puerto Vallarta, Mexico.

[5] X. Chen, J. He, X. Wu, W. Yan, W. Wie, *Sleep Staging by Bidirectional Long Short-term Memory Convolution Neural Network,* Future Generation Computer Systems 109 (2020) 188-196.

[6] J. K. Appati, G. K. Gogovi, G. O. Fosu, *On the Selection of Appropriate Kernel Function for SVM in Face Recognition,* International Journal of Advanced Research in Computer Science and Software Engineering 4(3) (2014) 6-9.

[7] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, P. A. Torres-Carrasquillo, *Support Vector Machines for Speaker and Language Recognition,* Computer Speech and Language 20(2-3) (2006) 210-229.

[8] S. Bellamkonda, N. P. Gopalan, *A Facial Expression Recognition Model Using Support Vector Machines,* IJ Mathematical Sciences and Computing 4 (2018) 56-65.

[9] M. Rezaei, E. Zereshki, H. Sharini, M. Gharib Salehi, F. Naleini, *Detection of Alzheimer's Disease Based on Magnetic Resonance Imaging of The Brain Using Support Vector Machine Model,* Tehran University Medical Journal TUMS Publications 76(6) (2018) 410-416.

[10] Ö. Y. Akşehirli, H. Ankaralı, D. Aydın, Ö. Saraçlı, *An Alternative Approach in Medical Diagnosis: Support Vector Machines,* Türkiye Klinikleri Journal of Biostatistics 5(1) (2013) 19-28.

[11] F. E. Tay, L. Cao, *Application of Support Vector Machines in Financial Time Series Forecasting,* Omega 29(4) (2001) 309-317.

[12] P. Wang, R. Mathieu, J. Ke, H. J. Cai, *Predicting Criminal Recidivism with Support Vector Machine,* International Conference on Management and Service Science (2010) Wuhan, China.

[13] J. Karia, *Stock Market Prediction Using Machine Learning,* International Journal of Emerging Technology and Computer Science 3(2) (2018) 159-162.

[14] J. Wainer, P. Fonseca, *How to Tune the RBF SVM Hyperparameters? An Empirical Evaluation of 18 Search Algorithms." arXiv preprint arXiv:2008.11655* (2020).

[15] D. Fradkin, I. Muchnik, *Support Vector Machines for Classification,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science 70 (2006) 13-20.

[16] S. R. Gunn, *Support Vector Machines for Classification and Regression,* ISIS Technical Report 14(1) (1998) 5-16.

[17] C. F. Lin, S. D. Wang, *Fuzzy Support Vector Machines,* IEEE Transactions on Neural Networks 13(2) (2002) 464-471.

[18] C. Savas, F. Dovis, *The Impact of Different Kernel Functions on The Performance of Scintillation Detection Based on Support Vector Machines,* Sensors 19(23) (2019) 1-16.

[19] D. Dua, C. Graff, *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]*, Irvine, CA: University of California, School of Information and Computer Science, (2019).