

## EXPLORING EFFICIENT KERNEL FUNCTIONS FOR SUPPORT VECTOR CLUSTERING

Furkan Burak BAĞCI, Faculty of Engineering, Department of Electrical and Computer Engineering, Ankara Yıldırım Beyazıt University, Turkey, D100000832@ybu.edu.tr

([ID](https://orcid.org/0000-0002-8666-809X) <https://orcid.org/0000-0002-8666-809X>)

Ömer KARAL\*, Faculty of Engineering, Department of Electrical and Computer Engineering, Ankara Yıldırım Beyazıt University, Turkey, karal@ybu.edu.tr

([ID](https://orcid.org/0000-0001-8742-8189) <https://orcid.org/0000-0001-8742-8189>)

Received: 14.03.2020, Accepted: 23.08.2020

\*Corresponding author

Research Article

DOI: 10.22531/muglajsci.703790

### Abstract

Clustering is an effective tool that divides data into different classes to reveal internal and previously unknown data schemes. However, in conventional clustering algorithms such as the  $k$ -means,  $k$ -NN, fuzzy  $c$  tool, the selection of the appropriate number of clusters for each data set is uncertain and varies with the data sets. Furthermore, the data sets to which the clustering algorithm is applied generally have nonlinear boundaries between clusters. Determining these nonlinear boundaries in the input space causes a complex problem. To overcome these problems, kernel-based clustering methods have been developed in recent years, which automatically determine the number and boundaries of clusters. In particular, the Support Vector Clustering (SVC) algorithm has received great attention in data analysis because of its features such as automatically determining the number of clusters and recognizing nonlinear boundaries based on the Gaussian kernel parameter. The number of clusters and region boundaries produced by SVC may show variation depending on the choice of the kernel function and its parameters. Therefore, the choice of kernel function plays a significant role. In this study, for the first time, the implementation of two different kernel (Cauchy and Laplacian) functions and evaluation of their performances have been realized within the framework of SVC. It was observed that the Laplacian kernel function performed better than Gauss and Cauchy kernel functions.

**Keywords:** Clustering, Unsupervised Learning, Support Vector Clustering, Cauchy Kernel, Laplacian Kernel

## ELİPSOİT DESTEK VEKTÖR ÖBEKLEME ALGORİTMASI İÇİN FARKLI KERNEL FONKSİYONLARININ İRDELENMESİ

### Özet

Öbekleme verideki bilinmeyen desenleri açığa çıkararak farklı sınıflara ayıran etkili bir araçtır. Ancak,  $k$ -ortalama,  $k$ -NN, bulanık  $c$ -ortalama gibi geleneksel öbekleme algoritmalarında, veriye göre değişken olan öbek sayısının seçimi belirsizdir. Dahası, öbekleme algoritmalarının uygulanacağı veri setleri genellikle öbekler arası doğrusal olmayan sınırlara sahiptir. Bu doğrusal olmayan sınırları giriş uzayında belirlemek karmaşık bir problemdir. Bahsi geçen sorunları çözmek için, son yıllarda öbek sayısını ve sınırlarını otomatik olarak belirleyen kernel tabanlı öbekleme yöntemleri geliştirilmiştir. Özellikle, Destek Vektör Kümeleme (DVK) algoritması öbek sayısını otomatik olarak belirleme ve Gauss kernel parametresine göre doğrusal olmayan sınırları ortaya çıkarma gibi özellikleriyle veri analizinde büyük ilgi görmektedir. DVK tarafından belirlenen öbek ve öbekler arası sınırlar, kernel fonksiyonunun seçimine ve parametrelerine bağlı olarak değişiklik gösterebilir. Bundan dolayı, kernel fonksiyonunun seçimi önemli bir rol oynar. Bu çalışmada, ilk kez, DVK çatısı altında iki farklı kernel (Cauchy ve Laplacian) fonksiyonunun uygulanması ve performanslarının değerlendirilmesi gerçekleştirilmiştir. Elde edilen sonuçlardan Laplacian kernel fonksiyonunun Gauss ve Cauchy kernel fonksiyonlarından daha iyi performans gösterdiği gözlemlenmiştir.

**Anahtar Kelimeler:** Öbekleme, Denetimsiz Öğrenme, Destek Vektör Öbekleme, Cauchy Kernel, Laplacian Kernel

### Cite

Bağcı, F. B., Karal, Ö., (2020). "Exploring efficient kernel functions for support vector clustering", *Mugla Journal of Science and Technology*, 6(2), 36-42.

### 1. Introduction

As an unsupervised approach, clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set [1]. Clustering algorithms aim to

understand the general structure and characteristics of the limited data by finding the similarities between the data samples [2]. These algorithms are applied in many scientific fields like medicine [3-7], engineering [8], finance [9], internet [10].

Despite the widespread use and popularity of clustering algorithms, there are some disadvantages. The parameterization level in conventional clustering algorithms is one of the most important issue. For example, in algorithms such as k-means and fuzzy c-means, choosing the number of clusters suitable for each data set has an important effect on the result. Generally, this low-level parameterization is not obvious and varies according to the data sets. Moreover, the data sets to which the clustering algorithm will be applied generally have nonlinear boundaries between clusters. Identifying these non-linear boundaries in the input space is a complex problem. To resolve this problem, firstly, the data in the low dimensional input space is mapped into a high-dimensional space called the feature-space by means of a non-linear transformation function. The clustering process is then performed in this high-dimensional space by finding a hyper-plane to separate the clusters and then mapping this hyper-plane back to the input space. The algorithms using this approach are called kernel-based methods [11] in general. According to the Mercers Theorem, there is a mapping function for any positive definite kernel function [12]. In other words, any valid kernel function can be expressed as the dot product of the data samples in the feature space. Therefore, there is no need to explicitly calculate the mapping function that moves the data from the input space to the higher dimensional space. Instead, the kernel functions are used in place of the dot product of the mapping functions. For this purpose, it is checked whether the kernel function is positive definite rather than what the mapping function is. For this reason, kernel-based clustering algorithms has an advantage over classical clustering algorithms and these methods over performed others in the literature by performing clustering operation in high dimensional space instead of data space. Furthermore, as mentioned above, by means of the kernel function, these algorithms calculate similarities of the data samples in high dimensional space without mapping the data samples to high dimensional space. With this process called kernel trick, kernel-based algorithms perform with higher accuracy rate than conventional algorithms, but do not have high computational complexity by performing all operations in the data space.

Kernel-based, non-parametric Support Vector Clustering (SVC) method [13] was developed inspired by one of the well-known supervised classification algorithms named Support Vector Machines (SVM). SVC has pointed significant interest due to its ability to generate arbitrary cluster shape by using Gaussian kernel parameter, automatically determination of the number of clusters, no need prior information etc. This method is very interesting with its aspects and used in fields like, electrical engineering [14], economics [15], and medicine [16, 17] and performance analysis is done with several synthetic and real data sets [18-23].

According to our recent search, there isn't any study that implements SVC algorithms with different kernel function other than Gaussian in literature. However, SVM based classification algorithms have been implemented with different kernel functions. Kavzoğlu and Colkesen [24] compared Maximum likelihood classifier and SVM classifier (using polynomial and radial basis function kernels) for land cover classification. It has been observed that SVM with radial based function (RBF) kernel performs better than the other in terms of general and individual class accuracy. Chen et al. [25] compared the six GIS-based prediction methods such as frequency ratio, index of entropy, and SVM with four kernel functions (linear, polynomial, RBF, and sigmoid) for landslide susceptibility mapping. It has been observed that SVM with RBF kernel gives the best performance. Sharafi et al. [26] applied the SVM technique using six different kernel functions (polynomial, sigmoid, exponential, Gaussian, Laplacian and rational quadratic) to predict scour depth around bridge piers. It has been observed that SVM with polynomial kernel function (SVM-P) produce the most accurate results compared with other kernel functions. Then, it has been compared with two artificial intelligence (AI) methods, namely artificial neural networks (ANN) and adaptive neuro-fuzzy inference systems (ANFIS). Clearly, SVM-P was found to be more accurate than the two AI techniques. Zhang et al. [27] presented the Optimal Relaxation Factor (ORF), which is a set of new SVM kernel functions for speech recognition. In the SVM framework, the ORF kernel was compared with RBF, Exponential RBF, and the Kernel with Moderate Decreasing, and higher recognition accuracy was achieved with the ORF-based SVM. Feizizadeh et al. [28] compare the predictive performance of GIS-based landslide susceptibility mapping (LSM) using four different kernel functions in SVM. RBF was observed to be the most efficient kernel among other kernels for LSM. They also stated that SVM would generally be an effective method for LSM, provided that the kernel function is evaluated carefully. Fadel et al. [29] studied the effect of different kernel functions on the performance of SVM for recognizing Arabic characters with eleven different kernel functions. From the results, it was seen that Exponential and Laplacian kernels showed excellent performance and others such as multi-quadratic kernel could not recognize the characters, especially with increased level of noise.

Although many kernel functions have been previously proposed and used for SVM-based classification, a comparative study of the SVC method using two different kernel functions has not been considered in the literature. For this reason, this research focuses on the implementation of Laplacian and Cauchy kernel functions and comparing their performances in the SVC framework. For this purpose, a real world benchmark data set, namely the Iris data set, is classified with SVC using three different kernel functions. In addition, a synthetic two-dimensional blobs data set from scikit-

learn library in Python programming language is used for visual inspection of the results.

Rest of the article contains the following sections: Method, Experimental Results, Conclusion, and References.

## 2. Method

In this section, the SVC algorithm is explained in detail and then the theory of kernel-based methods and kernel functions used in this study are examined.

### 2.1. Support Vector Clustering

Kernel-based SVC algorithm is first proposed by Ben-Hur et. al. [13] inspired by Support Vector Machines (SVM). SVC can provide arbitrary set of clusters, can determine the number of clusters not previously given to the system.

In SVC theory, first data points mapped from data space to the feature space using a non-linear mapping function. Then minimal feature space sphere that encloses most of the data points is found. Support Vectors (SVs) are on the boundary of this sphere, bounded support vectors are outside and all other points are inside of the sphere. After that, this sphere mapped backed to the data space so, it can be separated into the components. Points inside of each separate component are in the same cluster. Kernel parameter has high level control on the number of clusters by changing the mapping scale to the feature space. But, it does not directly control the number of clusters.

Consider a dataset, containing  $N$  points  $\{x_1, x_2, \dots, x_N\}$ ,  $x_i \in \mathbb{R}^d$  in  $d$  dimensional data space ( $X \subseteq \mathbb{R}^d$ ). Utilizing the non-linear transformation  $\phi$  from data space  $X$  to the high dimensional feature-space ( $\mathbb{R}^f$ ), a hypersphere that encloses the all mapped data points in the feature-space with minimum volume is generated.

This definition is described by the representation in Equation 1.

$$\begin{aligned} \min \quad & R^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & \|\Phi(x_i) - a\|^2 \leq R^2, R > 0, i = 1, \dots, N \end{aligned} \quad (1)$$

As shown in Equation 2, adding slack variables ( $\{\xi_1, \xi_2, \dots, \xi_N\}$ ,  $\xi_j \in \mathbb{R}$ ) that enables soft boundary calculation provides more robust and accurate representation.

$$\begin{aligned} \min \quad & R^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & \|\Phi(x_j) - a\|^2 \leq R^2 + \xi_j \\ & R > 0, \xi_j > 0, i = 1, \dots, N \end{aligned} \quad (2)$$

To solve problem, Lagrange multipliers are used in Equation 3.

$$\begin{aligned} L = R^2 - \sum_j (R^2 + \xi_j - \|\Phi(x_j) - a\|^2) \beta_j - \\ \sum_j \xi_j \mu_j + C \sum_j \xi_j \\ \beta_j \geq 0, \mu_j \geq 0 \end{aligned} \quad (3)$$

$\beta_j$  and  $\mu_j$  are Lagrange multipliers.  $C$  is a constant and  $C \sum_j \xi_j$  is error term. Setting the derivative of  $L$  to zero with respect to  $R$ ,  $a$  and  $\xi_j$  respectively gives the Equation 4, 5 and 6.

$$\sum_j \beta_j = 1 \quad (4)$$

$$a = \sum_j \beta_j \Phi(x_j) \quad (5)$$

$$\beta_j = C - \mu_j \quad (6)$$

Karush-Kuhn-Tucker conditions give Equation 7 and 8.

$$\xi_j \mu_j = 0 \quad (7)$$

$$(R^2 + \xi_j - \|\Phi(x_j) - a\|^2) \beta_j = 0 \quad (8)$$

Support Vectors are on the boundary of the sphere with  $0 < \beta_i < C$ , bounded support vectors are outside of the sphere with  $\xi_i > 0$ ,  $\beta_j = C$  and all other points are inside of the sphere.

Using Lagrange equation, the Wolfe dual form that is used in the optimization case is generated (Equation 9).

$$\begin{aligned} W = \sum_j \Phi(x_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(x_i) \Phi(x_j) \\ \text{s.t.} \quad 0 \leq \beta_i \leq C, i = 1, \dots, N \end{aligned} \quad (9)$$

According to the kernel trick, dot product of mapped samples is equal to the value of the kernel function with the data values in data space. This is described in Equation 10.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (10)$$

Gaussian kernel used in SVC is written in Equation 11.

$$K(x_i, x_j) = e^{-q \|x_i - x_j\|^2} \quad (11)$$

Hence, the optimization problem is written as in Equation 12.

$$\begin{aligned} W = \sum_j K(x_j, x_j) \beta_j - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \\ \text{s.t.} \quad 0 \leq \beta_i \leq C, i = 1, \dots, N \end{aligned} \quad (12)$$

To find the test function, first, the distance between an image of the sample and center of the sphere is found in Equation 13.

$$R^2 = \|\Phi(x_j) - a\|^2 \quad (13)$$

Putting Equation 4 into Equation 13 gives test function named trained kernel support function (TKSF) as in Equation 14.

$$f(x) := \|\Phi(x) - a\|^2 = K(x, x) - 2 \sum_j \beta_j K(x_j, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \quad (14)$$

Radius of the sphere is described in Equation 15.

$$R = \{f(x_i) \mid x_i \text{ is a support vector}\} \quad (15)$$

Cluster boundaries are defined as in Equation 16.

$$\{x | R(x) = R\} \quad (16)$$

The cluster assignment algorithm, which does not vary according to the elements in different clusters, includes a geometric approximation based on the distances from the center of sphere of the elements in the feature space. This approach is constructed on the following observation: a line that combines two points in the data space must exit the sphere in feature space. This is accomplished by the determining points, forming line that joining two points, randomly extracted from the data set and by comparing the distance to the center of sphere for each point in the feature space with the radius of sphere. If a point with a distance greater than the radius is found, it is understood that these two points are in different clusters and the value that represents these two points in the adjacency matrix ( $A$ ) is zero. However, if there isn't any point forming line has greater distance than radius of sphere, it is understood that these two points are in the same cluster and the value of the adjacency matrix is set to 1 (Equation 17). All values of the adjacency matrix are found by above-mentioned method. After determination of the adjacency matrix, cluster assignment process is performed by using the depth first search algorithm.

$$A_{i,j} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j, R(y) \leq R \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

## 2.2. Kernel Method

Kernel methods are ones that, by replacing the inner product with positive definite function, implicitly perform a non-linear mapping of input data into a high dimensional feature space (Equation 10).

Thanks to the kernel functions, the mapping process from input to high-dimensional space is carried out without the need to calculate the non-linear mapping function. It plays an important role in SVC. The kernel function is used in the optimization equation of the SVC algorithm where the dot product of the data in the feature space is expressed. This process is achieved by generating a kernel matrix of  $N \times N$  size for a data set with  $N$  instances. Each element of this kernel matrix shows the similarity of the data samples corresponding to the row and column index. Until this time, the Gaussian kernel function has been used with SVC. The values in the kernel matrix for Gaussian kernel function refer to the similarity of the data samples calculated by Gaussian distribution in the feature space. Using different kernels from the Gauss kernel function will induce different metric measurements and result in different similarities between the data samples. However, the SVC algorithm has not been implemented in the literature with any other kernel functions except Gauss kernel function.

In this article, the effects of non-Gaussian kernel functions such as Cauchy and Laplacian on cluster

results will be examined. The mathematical notations of the Gaussian, Cauchy and Laplacian kernel functions are listed in Equations 18, 19 and 20 respectively.

$$K(x_i, x_j) = e^{-q\|x_i - x_j\|^2} \quad (18)$$

$$K(x_i, x_j) = 1/(1 + (\beta\|x_i - x_j\|^2)) \quad (19)$$

$$K(x_i, x_j) = e^{-q\|x_i - x_j\|} \quad (20)$$

As can be seen from Figure 1, although Gauss, Cauchy and Laplacian kernels have similar appearance, their statistical properties are slightly different from each other. For example, the Laplacian kernel function has a sharper peak and broad tails compared with the Gaussian and Cauchy kernel functions.

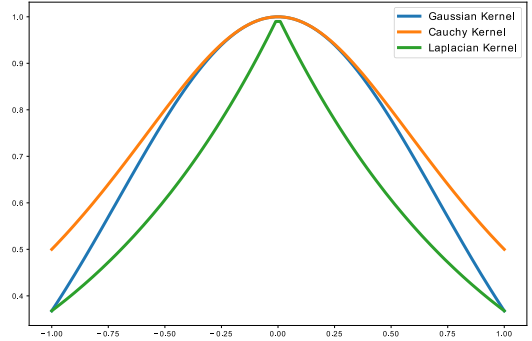


Figure 1. The graph of Gaussian, Cauchy and Laplacian kernel functions with  $q = 1$ ,  $x_i = -1, \dots, 1$  and  $x_j = 0$

## 3. Experimental Results

In this section, Cauchy and Laplacian kernel functions are applied along with the Gaussian kernel function in SVC. The effects of different kernel functions on performance have been demonstrated by the implementation of SVC with synthetic and real-world iris data sets. The two-dimensional blobs data set is generated by using one of the well-known open source machine learning library named scikit-learn in Python programming language.

### 3.1. Synthetic Data Set

The two-dimensional Blobs data set consists of 200 samples with three clusters. One of the sets in the Blobs data set can be easily separated from the others. However, since the other two clusters overlap, it is very difficult to divide them into two different clusters. The data set is shown in Figure 2. Each class is visualized with different colors so that classes can be easily distinguished

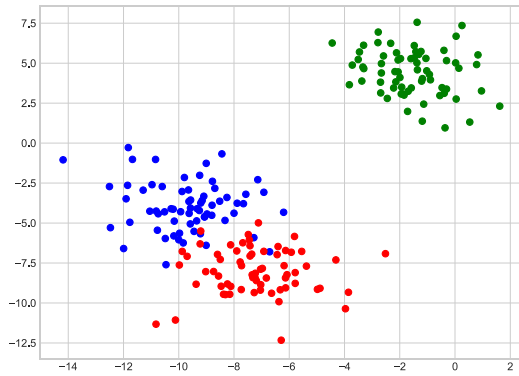
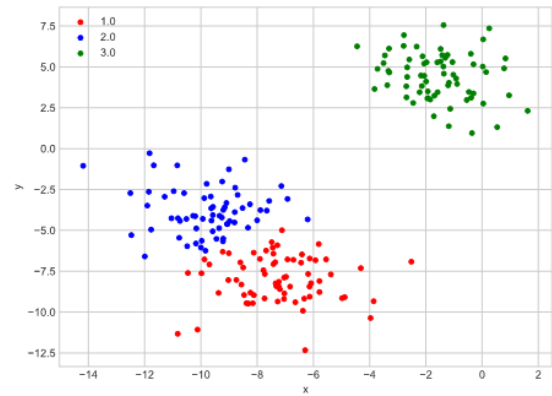


Figure 2. Blobs Data Set



(C)

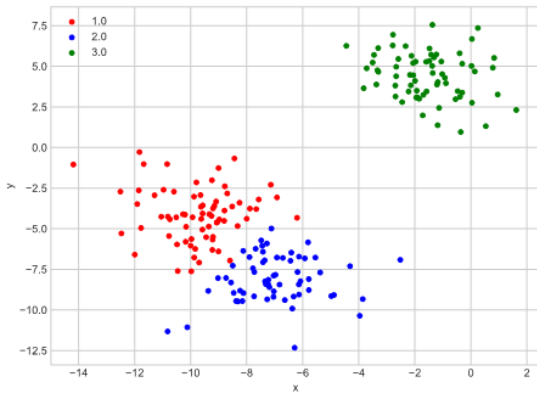
Figure 3. Experimental results of SVC with three different kernel functions in the Blobs synthetic data set  
 A) Gaussian B) Cauchy C) Laplacian

Figure 3 demonstrates the results of SVC with three different kernel functions in the Blobs synthetic data set. As shown in Figure 3 and Table 1, the Blobs data set is divided into 3 clusters with over 96% accuracy with 2 different kernels proposed for the SVC algorithm. The best accuracy was obtained from the SVC algorithm using Laplacian kernel with 97.5%.

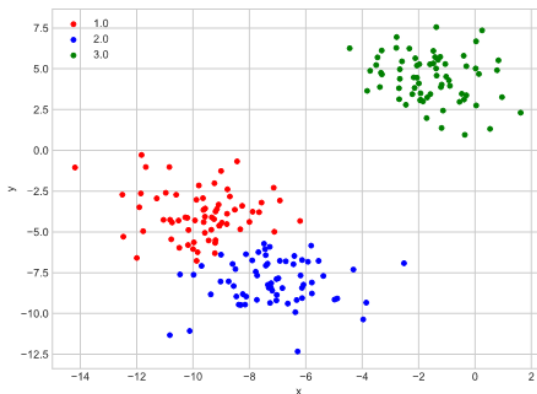
The performance results of the three different kernel functions proposed in the SVC algorithm on synthetic data sets are shown in Table 1.

**Table 1.** Experimental results with Blobs data sets

Accuracy %		Kernel Function		
		Gaussian	Cauchy	Laplacian
Data Set	Blobs	96	96	97.5



(A)



(B)

### 3.2. Real World Iris Data Set

Iris is the standard benchmarking data set in pattern recognition literature. It has 150 instances with 4 measurements (the length and the width of the sepals and petals, in centimeters) of an iris flower. There are 3 types (Setosa, Versicolor and Virginica) of iris flower in data set. Setosa class is linearly separable from the other two classes. Versicolor and Virginica classes are not linearly separable. It has 50 instances for each class. In literature, SVC algorithm was applied to the Iris data set with Gaussian kernel function with %90.66 accuracy [13].

For the iris data set, the confusion matrices of Gaussian, Cauchy and Laplacian kernel functions used in SVC algorithm are given in Table 2, Table 3 and Table 4 respectively. Furthermore, sensitivity, recall and F1 score values obtained for each class using these confusion matrices are given in Table 5, Table 6 and Table 7 respectively. Finally, the accuracy rates obtained in the classification of Iris data determined by Gaussian, Cauchy and Laplacian kernel functions used in SVC algorithm are given in Table 8.

**Table 2.** Confusion Matrix for Gaussian kernel function with iris data set

		Number Of Samples	Classified As		
			Setosa	Versicolor	Virginica
Actual Classes	Setosa	50	0	0	
	Versicolor	0	46	4	
	Virginica	0	11	39	

**Table 3.** Confusion Matrix for Cauchy kernel function with iris data set

		Number Of Samples	Classified As		
			Setosa	Versicolor	Virginica
Actual Classes	Setosa	50	0	0	
	Versicolor	0	41	9	
	Virginica	0	2	48	

**Table 4.** Confusion Matrix for Laplacian kernel function with iris data set

		Number Of Samples	Classified As		
			Setosa	Versicolor	Virginica
Actual Classes	Setosa	50	0	0	
	Versicolor	1	46	3	
	Virginica	0	2	48	

**Table 5.** Classification Metrics for Gaussian kernel function with iris data set

	Precision	Recall	F1-Score
Setosa	1	1	1
Versicolor	0.81	0.92	0.86
Virginica	0.91	0.78	0.84

**Table 6.** Classification Metrics for Cauchy kernel function with iris data set

	Precision	Recall	F1-Score
Setosa	1	1	1
Versicolor	0.95	0.82	0.88
Virginica	0.84	0.96	0.9

**Table 7.** Classification Metrics for Laplacian kernel function with iris data set

	Precision	Recall	F1-Score
Setosa	0.98	1	0.99
Versicolor	0.94	0.92	0.93
Virginica	0.94	0.94	0.94

**Table 8.** Accuracy values with iris data set

		Kernel Function		
		Gaussian	Cauchy	Laplacian
Data Set	Iris	90	92.6	95.3

In these experiments, SVC algorithm was applied to the iris dataset with 90% accuracy rate with Gaussian kernel function. Afterwards, 92.6% and 95.3% successes were obtained in the experiments with Cauchy and Laplacian kernel functions, respectively. In all experiments using these kernel functions, the data samples of the Setosa class were classified accurately from the other two classes. However, some samples of data in the Versicolor and Virginica classes are incorrectly classified.

#### 4. Conclusion

Clustering is used to find the structure in unlabeled data. In conventional clustering algorithms such as the k-means, k-NN, fuzzy c means, the selection of the appropriate number of clusters for each data set depends on the user. Furthermore, the application of conventional clustering algorithms to the unlabeled data results in nonlinear boundaries between clusters. Determining these boundaries in the input space causes a complex problem. Therefore, kernel-based clustering methods have received great attention in recent years. Especially, SVC algorithm has been applied in many scientific fields due to its features such as automatically determining the number of clusters and recognizing nonlinear boundaries depending on Gaussian kernel parameter. However, there are no studies in the literature that implement an SVC algorithm that uses a kernel function different from Gaussian. In this study, for the first time, implementation of two different kernel functions such as Cauchy and Laplacian kernels and evaluation of their performance were performed within the framework of SVC. The results showed that Laplacian kernel function works better than Gauss and Cauchy kernels. In the future, as an alternative to the commonly used Gauss kernel, performance analysis can be performed using Laplacian and Cauchy kernels.

#### 5. References

- [1] Xu, R. & WunschII, D. Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3), 645-678. 2005.
- [2] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A., Fofou, S. and Bouras, A. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. IEEE Transactions on Emerging Topics in Computing, 2(3), 267-279, 2014.
- [3] Xu R, Wunsch D. Clustering algorithms in biomedical research: A review. Biomedical Engineering, IEEE Reviews in, 3:120 – 154. 2010.
- [4] Wang D, Shi L, Yeung DS, Tsang ECC, Heng PA. Ellipsoidal support vector clustering for functional MRI analysis. Pattern Recognition, 40:2685–2695. 2007.
- [5] A. L. Gamboa, Hybrid Fuzzy-SV Clustering for Heart Disease Identification, CIMCA-IAWTIC, 6. 2006.

- [6] Singh N. & Mohapatra A. Breast cancer mass detection in mammograms using k-means and fuzzy c-means clustering. *Int. J. Com. Appl.* 22 (2), 15-21. 2011.
- [7] Khanmohammadi, S., Adibeig, N., & Shanehbandy, S. An Improved overlapping k-means clustering method for Medical applications. *Expert Systems With Applications*, 67, 12-18. 2017.
- [8] Chicco G. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy*, 42 (1), 68–80. 2012.
- [9] Kou, G. Peng, Y. Wang, G. Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Inf. Sci.* 275, 1–12. 2014.
- [10] Bochkaryov P. V. & Guseva A. I. The Use of Clustering Algorithms Ensemble with Variable Distance Metrics in Solving Problems of Web Mining. 5<sup>th</sup> International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 41-46. 2017.
- [11] Scholkopf, B., Mika, S., Burges, C. J., Knirsch, P., Muller, K. R., Ratsch, G., & Smola, A. J. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5),1000-1017. 1999.
- [12] Minh, H. Q., Niyogi, P., & Yao, Y. Mercer's theorem, feature maps, and smoothing. In *International Conference on Computational Learning Theory Springer, Berlin, Heidelberg*, 154-168. 2006.
- [13] Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. Support vector clustering. *Journal of machine learning research*, 125-137. 2001.
- [14] Chicco G. & Ilie I.-S., Support vector clustering of electrical load pattern data. *IEEE Trans. on Power Systems*, 24, no. 3, 1619- 1628, 2009.
- [15] Huang J.-J., Tzeng G.-H., & C.-S. Ong, Marketing segmentation using support vector clustering, *Expert Systems with Applications*, 32 (2), 313–317. 2007.
- [16] Wang D., Shi L., Yeung D. S., Heng P. A., T. T. Wong, & Tsang E. C. C.. Support vector clustering for brain activation detection. In *Proc. of the 8<sup>th</sup> International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 572–579, Oct. 2005.
- [17] Garcia, C. & Moreno, J. Application of support vector clustering to the visualization of medical images. *IEEE International Symposium on Biomedical Imaging*, 1553–1556. 2004.
- [18] Lee, J., & Lee, D. Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11), 1869–1874. 2006.
- [19] Lee J. & Lee D. An Improved Cluster Labeling Method for Support Vector Clustering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27, no. 3, 461-464, Mar. 2005.
- [20] DonGiovanni, D. & Vaina, L. Select and Cluster: A Method for Finding Functional Networks of Clustered Voxels in fMRI. *Computational Intelligence and Neuroscience*, 1-19. 2016.
- [21] Yin, Z. & Zhang, J. Identification of temporal variations in mental workload using locally-linear-embedding-based EEG feature reduction and supportvector- achine-based clustering and classification techniques. *Computer Methods and Programs in Biomedicine*, 115(3), 119-134. 2014.
- [22] Villazana, S., Seijas, C., & Caralli, A. Lempel-Ziv complexity and Shannon entropy-based support vector clustering of ECG signals. *Revista Ingeniería Uc*, 22(1). 2015
- [23] Karal, Ö., & Bağcı, F. B. Elipsoit Destek Vektör Öbekleme Algoritmasının Biyomedikal Veri Setleri Üzerinde Karşılaştırmalı Performans Analizi. *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, 7(1), 140-148, 2019.
- [24] Kavzoglu, T., & Colkesen, I.A kernel functions analysis for support vector machines for land cover classification. *International Journal of Applied Earth Observation and Geoinformation*, 11(5), 352-359. 2009.
- [25] Chen, W., Wang, J., Xie, X., Hong, H., Van Trung, N., Bui, D. T., ... & Li, X. Spatial prediction of landslide susceptibility using integrated frequency ratio with entropy and support vector machines by different kernel functions. *Environmental Earth Sciences*, 75(20), 1344. 2016.
- [26] Sharafi, H., Ebtehaj, I., Bonakdari, H., & Zaji, A. H. Design of a support vector machine with different kernel functions to predict scour depth around bridge piers. *Natural Hazards*, 84(3), 2145-2162. 2016.
- [27] Zhang, X., Liu, X., & Wang, Z. J. Evaluation of a set of new ORF kernel functions of SVM for speech recognition. *Engineering Applications of Artificial Intelligence*, 26(10), 2574-2580. 2013.
- [28] Feizizadeh, B., Roodposhti, M. S., Blaschke, T., & Aryal, J. Comparing GIS-based support vector machine kernel functions for landslide susceptibility mapping. *Arabian Journal of Geosciences*, 10(5), 122. 2017.
- [29] Fadel, S., Ghoniemy, S., Abdallah, M., Sorra, H. A., Ashour, A., & Ansary, A. Investigating the effect of different kernel functions on the performance of SVM for recognizing Arabic characters. *IJACSA) International Journal of Advanced Computer Science and Applications*, 7(1), 446-450. 2016.