Kernel Based Learning

MOHAMMAD ASHRAFUZZAMAN SIDDIQI[1]

Contents

Abstract: Kernel-based learning has emerged as an effective machine learning method for dealing with complicated and nonlinear data. It permits the mapping of input data into high-dimensional feature spaces by utilizing kernel functions, allowing linear models to handle intricate interactions quickly. Furthermore, kernel-based learning provides a versatile framework for adding past information, which improves the performance of learning algorithms. This paper discusses kernel-based learning, including its motivation, methodologies, applications, and problems. The goal is to demonstrate the adaptability and efficacy of kernel-based learning in a variety of disciplines, including computer vision, natural language processing, bioinformatics, and finance. In addition, the paper examines the fundamental issues connected with kernel-based learning, such as kernel function selection, the curse of dimensionality, and processing costs.

[1] mohammad-ashrafuzzaman.siddiqi@stud.hshl.de

1 Motivation

Kernel-based learning is a popular machine learning approach that has been extensively employed in numerous domains, including speech recognition, picture recognition, bioinformatics, and natural language processing. Many real-world issues are complicated and nonlinear, making it challenging for typical machine learning methods to develop useful models [HSS08].

However, kernel-based learning provides a compelling solution to this challenge because it uses the power of kernel functions to implicitly map input into high-dimensional feature spaces, allowing linear models to conduct sophisticated nonlinear computations quickly [Sh04]. Kernel-based learning, by adding the kernel trick, enables the building of meaningful models capable of effectively capturing intricate relationships within data, making it a valuable technique for tackling real-world difficulties [Sh04].

In figure 1 Kernel functions translate the dataset from 2 dimensions to 3 dimensions, creating a distinct decision surface that divides the classes.

Overall, the necessity to develop efficient models for complicated, nonlinear data is what drives kernel-based learning. Kernel-based learning algorithms can learn complex functions and properly classify data by converting the input into a high-dimensional feature space.

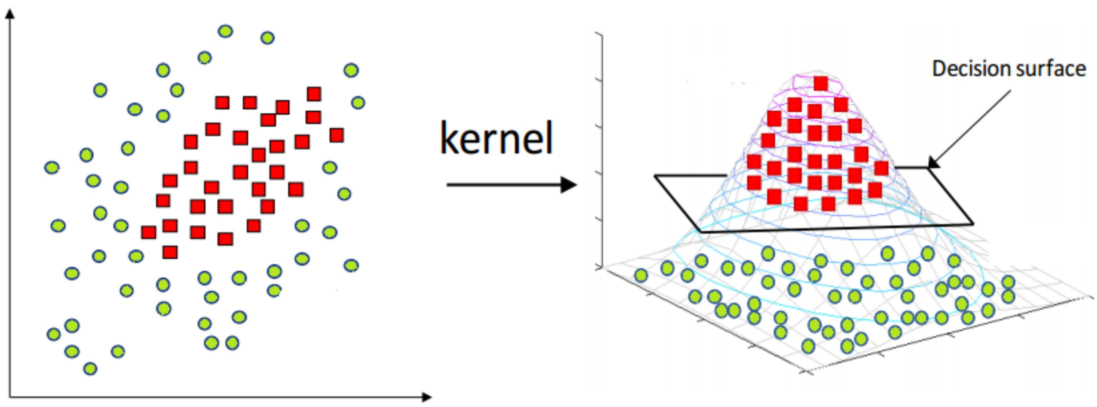

Fig. 1: Kernel trick [zh18].

2 Kernel Based Learning

Kernel-based learning is based on the mathematical idea of a kernel function, which is a positive semi-definite function that evaluates the similarity of two data points. The kernel function maps the data points into a higher-dimensional feature space, where linear algorithms can separate the data[Sc02].

The linear kernel, polynomial kernel, Gaussian radial basis function (RBF) kernel, and sigmoid kernel are the most often utilized kernel functions in kernel-based learning. Each kernel function has unique qualities and is best suited to specific types of data. The linear kernel, for example, is used for linearly separable data, whereas the RBF kernel is utilized for nonlinear data [Sh04].

The kernel trick, kernel functions, and regularization provide the cornerstone of kernel-based learning. Kernel-based learning methods can develop efficient models for complicated, nonlinear data by translating the data into a higher-dimensional feature space using a kernel function and maximizing the margin between various classes.

2.1 Application

1. Kernel-based learning is essential in computer vision, especially deep learning. It permits the effective capturing of complicated and nonlinear patterns in visual data by using kernel functions [O'19]. Using kernel-based learning in computer vision increases representation learning and performance in tasks such as picture classification, object recognition, and semantic segmentation, making it a potent strategy for expanding computer vision capabilities [O'19].

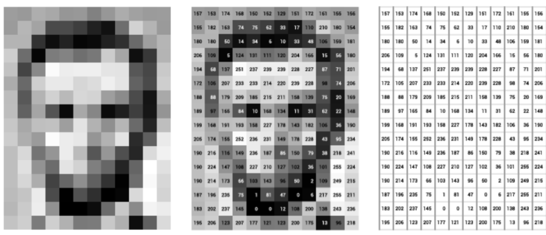

Fig. 2: Computer vision shown,kernel method used [Ba22].

2.2 Challenges

- The Curse of Dimensionality: When the number of features (dimensions) is considerable in comparison to the number of training samples, kernel approaches, like other machine learning algorithms, might suffer from overfitting. When dealing with a high-dimensional feature space, the model may match the training data quite well but generalize poorly to unobserved data [Gé22].

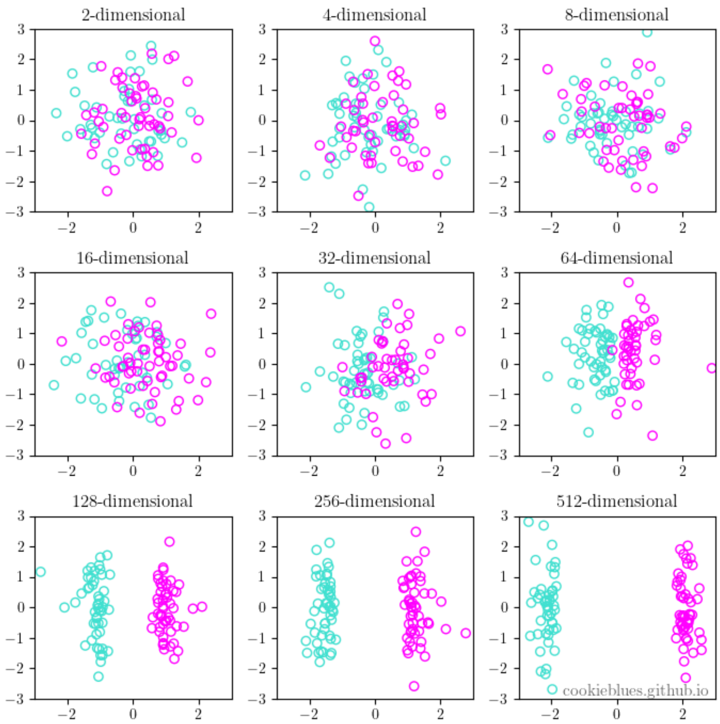

Fig. 3: The feature space becomes sparser and the data easier to separate as the dimensionality of the data grows. In a word, this is the curse of dimensionality.[HR21].

3 Implementation of SVM classification with RBF Kernel

3.1 Dataset

Consider the Iris dataset to demonstrate the practical implementation of kernel-based learning. The Iris dataset includes measurements of sepal length, petal length, petal width, and sepal width for three different species of Iris flowers. We can develop a classification model to differentiate between these species based on the available features by combining the Support Vector Machine (SVM) algorithm with the radial basis function (RBF) kernel.

A1 | fx sepal length (cm)

	A	B	C	D	E	F
1	sepal lengt	sepal widtl	petal lengt	petal widtl	target	
2	5.1	3.5	1.4	0.2	setosa	
3	4.9	3	1.4	0.2	setosa	
4	4.7	3.2	1.3	0.2	setosa	
5	4.6	3.1	1.5	0.2	setosa	
6	5	3.6	1.4	0.2	setosa	
7	5.4	3.9	1.7	0.4	setosa	
8	4.6	3.4	1.4	0.3	setosa	
9	5	3.4	1.5	0.2	setosa	
10	4.4	2.9	1.4	0.2	setosa	
11	4.9	3.1	1.5	0.1	setosa	
12	5.4	3.7	1.5	0.2	setosa	
13	4.8	3.4	1.6	0.2	setosa	
14	4.8	3	1.4	0.1	setosa	
15	4.3	3	1.1	0.1	setosa	
16	5.8	4	1.2	0.2	setosa	
17	5.7	4.4	1.5	0.4	setosa	
18	5.4	3.9	1.3	0.4	setosa	
19	5.1	3.5	1.4	0.3	setosa	

Fig. 4: Preview of data full set foundhere.

3.2 Why RBF?

The RBF (Radial Basis Function) kernel is used in the SVM (Support Vector Machine) classifier. The RBF kernel is a popular choice for SVMs as it allows for nonlinear decision boundaries. The kernel parameter in the SVC class is set to 'rbf', indicating that the RBF kernel should be used. The RBF kernel calculates the similarity between two samples as a function of the Euclidean distance between them. It transforms the input space into a higher-dimensional feature space, where the samples can be linearly separable.

3.3 Code

The implementation includes importing relevant libraries and the Iris dataset , splitting it into training and testing sets, fitting the SVM classifier with the RBF kernel to the training data, and making predictions on the test data. The outcomes are depicted visually by putting the decision border and data points on a graph, with different classes indicated by different colors. This is all done in Python, importing libraries like Sci-kit to enable the process.

```python
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

# Load the Iris dataset
iris = datasets.load_iris()
X = iris.data[:, :2]  # Consider only the first two features (sepal length and sepal width)
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create an SVM classifier with the RBF kernel
svm = SVC(kernel='rbf')

# Fit the classifier to the training data
svm.fit(X_train, y_train)

# Make predictions on the test data
y_pred = svm.predict(X_test)
```

Fig. 5: Preview of code,full code found here.

3.4 Results

The data points in plot 6 are scattered across the graph, with each point representing an Iris flower sample. The plot's background is colored in various colours to represent the SVM model's decision boundary. The decision boundary divides the feature space into distinct regions, each of which corresponds to a different class. These separate locations are represented by the plot areas colored blue, red, and brown.

Furthermore, the data points are colored according to their various classes. The blue points represent one species of Iris, the red points another species, and the brown points a third species. This color-coded depiction helps us to see how the SVM model sorted the data points into their respective classes depending on the specified features.

We can determine the accuracy of the SVM classifier with the RBF kernel in correctly identifying the Iris flowers based on their sepal length and width by studying the output plot. The decision boundary assists us in understanding the regions where each species is most common. We can visually evaluate the data point separation and see how the SVM algorithm successfully differentiates between the various Iris species based on these two features.

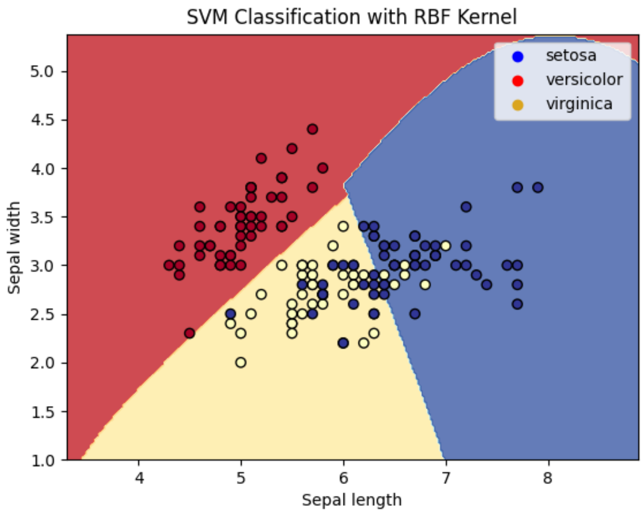

Fig. 6: SVM PLOT

4 Conclusion

Finally, kernel-based learning provides a compelling method for dealing with complicated and nonlinear data in machine learning. This approach, which makes use of kernel functions and the kernel trick, enables the creation of models capable of capturing intricate relationships within data, surpassing the constraints of classic linear algorithms. The adaptability of kernel-based learning allows for the insertion of existing knowledge, domain-specific information, or structural limitations, which improves the learning process even further. Furthermore, the diverse set of applications, which includes computer vision, natural language processing, bioinformatics, and finance, proves the versatility and efficacy of kernel-based learning in a variety of disciplines. However, issues like : selecting proper kernel functions, the curse of dimensionality, and processing costs remain. Future research should concentrate on overcoming these difficulties in order to increase the performance and scalability of kernel-based learning algorithms. Overall, kernel-based learning is an important technique in the machine learning toolkit, allowing academics and practitioners to address real-world challenges associated with complicated, nonlinear data.

5 Declaration of Originality

I, MOHAMMAD ASHRAFUZZAMAN SIDDIQI, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

13/05/2023&Dortmund - MOHAMMAD ASHRAFUZZAMAN SIDDIQI

Bibliography

[Ba22] Bandyopadhyay, Hmrishav: , What is Computer Vision? https://www.v7labs.com/blog/what-is-computer-vision, 22. Article with extensive definition of computer vision and relevant images.

[Gé22] Géron, Aurélien: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. Ö'Reilly Media, Inc.", 2022. The Python programs Scikit-Learn and TensorFlow are used in this book to provide a hands-on introduction to machine learning.

[HR21] Hrouda-Rasmussen, Stefan: , The Curse of Dimensionality. https://towardsdatascience.com/the-curse-of-dimensionality-5673118fe6d2, 2021. Detailed explanation of curse of dimensionality.

[HSS08] Hofmann, Thomas; Schölkopf, Bernhard; Smola, Alexander J: Kernel methods in machine learning. 2008. This article covers the most essential kernel methods used in machine learning, such as kernel SVMs, kernel PCA, and kernel clustering. They also go through the theoretical underpinnings of kernel methods as well as their practical implementations. This site is helpful in providing a thorough understanding of kernel approaches and their numerous applications in machine learning.

[O'19] O'Mahony, Niall; Campbell, Sean; Carvalho, Anderson; Harapanahalli, Suman; Hernandez, Gustavo Velasco; Krpalkova, Lenka; Riordan, Daniel; Walsh, Joseph: Deep Learning vs. Traditional Computer Vision. arXiv preprint arXiv:1910.13796, 2019. The article discusses deep learning in comparison with traditional computer vision.It talks about CNNS which make use of kernels.

[Sc02] Schölkopf, Bernhard; Smola, Alexander J; Bach, Francis et al.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002. This book delves deeply into the mathematical foundations of kernel methods and is an essential contribution to the field of kernel-based learning. The comprehensive introduction to kernel-based learning that is provided includes support vector machines. Regularization, optimization, and multiple kernel types are among the subjects covered in the book.

[Sh04] Shawe-Taylor, John; Cristianini, Nello et al.: Kernel methods for pattern analysis. Cambridge university press, 2004. The book discusses a variety of topics related to kernel approaches, including the theory behind them, how they might be applied in the real world, and how they work on a computer. In addition, the book informs me about the benefits and cons of various kernel approaches.

[zh18] zhang, Grace: , What is the Kernel Trick? Why is it Important? `https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d`, 2018. discusses kernel tricks and importance of kernel tricks , contains relevants diagrams.