THINK INDIA JOURNAL

ISSN: 0971-1260

Vol-22-Issue-3-July-2019

Optical Character Recognition Using Support Vector Machine

Surbhi,

School of Computer Science & Engineering, Lovely Professional University, Phagwara, India
surbhi.23540@lpu.co.in

Shivraj Puggal,

School of Mechanical Engineering, Lovely Professional University, Phagwara, India
shivraj.16125@lpu.co.in

ABSTRACT: In the field of electronics and image processing; computer vision, artificial intelligence and pattern recognition has a great significance. Optical character recognition (OCR) has evolved a lot since the beginning and is one of the important aspects of pattern recognition. OCR converts the optical data, which is recognized from the readable characters into the digital form. Different approaches are clubbed with various methodologies for this purpose. In this paper, we are going to recognize optical characters using a very famous machine learning technique, Support Vector Machine. SVM is a very powerful black box testing algorithm which classifies data by creating a hyperplane or line. The main reason for such programming is to process paper based records by changing over printed or written by hand message into an electronic structure to be spared in a database. In this paper we are going to recognize optical character using SVM for different kernel parameters.

KEYWORDS: Recognition, Support Vector, Kernel functions, Hyperplane.

1. INTRODUCTION

The Support Vector Machine (SVM) was first proposed by Vapnik and has since attracted a high degree of interest in the machine learning research community [2]. Several recent studies have reported that the SVM (support vector machines) generally are capable of delivering higher performance in terms of classification accuracy than the other data classification algorithms. SVM have been employed in a wide range of real world problems such as text categorization, hand-written digit recognition, tone recognition, image classification and object detection, micro-array gene expression data analysis, data classification. It has been shown that SVM is consistently superior to other supervised learning methods. However, for some datasets, the performance of SVM is very sensitive to how the cost parameter and kernel parameters are set. As a result, the user normally needs to conduct extensive cross validation in order to figure out the optimal parameter setting. This process is commonly referred to as model selection. One practical issue with model selection is that this process is very time consuming. We have experimented with a number of parameters associated with the use of the SVM algorithm that can impact the results. This

paper is organized as follows. In next section, we introduce some related background including some basic concepts of SVM, classification with hyperplanes, and kernel selection. In Section 3, the dataset is described. In Section 4, we detail all experiments results. Finally, we have some conclusions in Section 5.

2. SUPPORT VECTOR MACHINE

A Support Vector Machine is envisioned as a surface that creates a limit between the information plotted in multidimensional area that speaks to model and its component.

The objective is to make a straight limit called a hyperplane that separates the space to make genuinely uniform parcels on either side.

Fortunately in spite of the fact that the math might be troublesome, the essential ideas are reasonable. SVMs can be adjusted for use with about any learning task, including both order and numeric forecast. Along these lines, in the rest of the areas, we will concentrate just on SVM classifiers.

2.1 Hyperplanes

SVMs utilize a limit called a hyperplane to parcel information into gatherings of comparable classes. At the point when the information is isolated flawlessly by the straight line or flat surface, they are said to be directly distinct. From the outset, we'll consider just the straightforward situation where this is valid, yet SVMs can likewise be stretched out to issues where the focuses are not directly distinct.

In two dimensions, the assignment of the SVM calculation is to recognize a line that isolates the two classes. To locate the best parcel, a quest is accomplished for the Maximum Margin Hyperplane (MMH) that makes the best partition between the two classes. The line that prompts the best partition will sum up the best to the future information. The maximum margin will improve the chance that, in spite of random noise, the points will remain on the correct side of the boundary.

The support vectors are the points from each class that are the closest to the MMH; each class must have at least one support vector, but it is possible to have more than one. Using the support vectors alone, it is possible to define the MMH. This is a key feature of SVMs; the support vectors provide a very compact way to store a classification model, even if the number of features is extremely large.

2.2 Linearly separable data

It is easiest to understand how to find the maximum margin under the assumption that the classes are linearly separable. In this case, the MMH is as far away as possible from the outer boundaries of the two groups of data points. These outer boundaries are known as the convex hull. The MMH is then the perpendicular bisector of the shortest line between the two convex hulls. Sophisticated computer algorithms that use a technique known as quadratic optimization are capable of finding the maximum margin in this way.

ISSN: 0971-1260

Vol-22-Issue-3-July-2019

An alternative (but equivalent) approach involves a search through the space of every possible hyperplane in order to find a set of two parallel planes that divide the points into homogeneous groups yet themselves are as far apart as possible. To comprehend this pursuit procedure, we have to characterize precisely what is hyperplane. In an n-dimensional space, the accompanying condition is utilized:

w' * x' + b = 0 **(1)**

The 'over the letters demonstrate vectors as opposed to static variables. Specifically, w is a vector of different loads and b is a static variable called as the predisposition.

Utilizing this recipe, the objective of the procedure is to locate a lot of loads that determine two hyperplanes:

w' * x' + b > = +1 **(2)**

w' * x' + b < = -1 **(3)**

The data is directly divisible. The vector geometry characterizes the separation between two planes as:

2/||w'|| **(4)**

Here ||w|| demonstrates Euclidean standard. Since ||w|| is in the denominator, to boost separation, we have to minimize||w||. The undertaking is regularly re communicated as a lot of requirements, as follows:

min 1/2 * $||w'||^2$ **(5)**

s.t. y_i (w' * x_i' - b) > = 1, for all x_i

The primary line suggests that we have to limit the Euclidean standard (squared and isolated by two to make the figuring simpler). The subsequent line takes note of this is dependent upon the condition that every one of the yi information focuses is effectively ordered. Note that y shows the class esteem.

Similarly, as with the other strategy for finding the most extreme edge, finding an answer for this issue is an undertaking best left for quadratic streamlining programming.

2.3 Non-linearly separable data

Imagine a scenario in which the data is not directly distinguishable. The answer for this is the utilization of a slack variable which makes a delicate edge that enables a few to fall on the wrong side of edge. A cost factor is applied to all focuses that damage the requirements, and instead of finding the MMH, the calculation tries to limit the total expense.

We can in this way overhaul the enhancement issue to:

min 1/2 $||w'||^2$ + c $\sum £_I$ **(6)**

s.t. y_i (w' * x' - b) > = 1 - $£_I$, for all x_i',$£_i$>=0

The significant piece to comprehend is the addition of cost parameter. The more prominent the cost factor is the harder the enhancement will attempt to accomplish full separation.

2.4 Kernels for non-linear spaces

In some genuine applications, the connections between variables are nonlinear. As we simply found, a SVM can in any case be prepared on such information through the slack variable which enables a few guides to be misclassified. In any case, this isn't the best way to move toward the issue of nonlinearity. A key element of SVMs is their capacity to map issue into a higher measurement. In doing as such, a nonlinear relationship may abruptly seem, by all accounts, to be very direct. This is conceivable in light of the fact that we have acquired another point of view on the information. SVMs with nonlinear bits add extra dimensions to the information so as to make partition along these lines. The kernel $\phi(x)$ is a mapping of the information into other space. Therefore, the function applies some transformation to the support vectors *xi* and *xj* , combines them using the dot product, which takes two vectors as input and returns a single number as result.

$$\mathbf{K(x_i', x_j') = \phi(x_i') * \phi(x_j')} \quad \mathbf{(7)}$$

Utilizing this structure, portion capacities have been created for a wide range of areas of data. A couple of the most normally utilized piece capacities are recorded as pursues. Almost all SVM programming bundles will incorporate these parts, among numerous others. The linear kernel doesn't change the data by any means. It can be expressed simply as:

$$\mathbf{K(x_i', x_j') = x_i' * x_j'} \quad \mathbf{(8)}$$

The polynomial kernel with degree *d* adds a non-linear transformation to the data:

$$\mathbf{K(x_i', x_j') = (x_i' * x_j' + 1)^d} \quad \mathbf{(9)}$$

The sigmoid bit brings about a SVM model to some degree closely resembling a neural arrange utilizing a sigmoid enactment work. The Greek letters are used as kernel parameters:

$$\mathbf{K(x_i', x_j') = \tanh(k\ x_i' * x_j' - \delta)} \quad \mathbf{(10)}$$

There is no dependable principle to coordinate kernel to a specific learning. This fit depends on the idea that model has to learn. Experimentation is required via preparing and assessing a few SVMs on an approved dataset.

3. Dataset

The dataset utilized is taken from the UCI Machine Learning Data Repository (http://archive.ics.uci.edu/ml) by W. Frey and D. J. Record. It contains 20,000 instances of 26 English letters in order capital letters as printed utilizing 20 distinctive arbitrarily reshaped and mutilated highly contrasting fonts. The following fig 1, distributed by Frey and Slate, gives a case of a portion of the printed glyphs. The letters are used by a PC to distinguish, yet are effectively perceived by a person:

ISSN: 0971-1260

Vol-22-Issue-3-July-2019

Fig 1

As indicated by the documentation when the glyphs are checked into the PC they are changed over into pixels and 16 measurable traits are recorded. The qualities measure such attributes as vertical measurements of the glyph, the extent of dark pixels, and the normal even and vertical situation of the pixels. This information with the 16 highlights is what characterizes every case of the letter class.

4. RESULTS OF EXPERIMENT

The optical character recognition is done using R programming on Rstudio. To provide a baseline measure of SVM, ksvm() is used from kernlab library is used. In this experiment, different values of kernels parameter are selected to check the efficient recognition of optical character. The kernel function is utilized in preparing and foreseeing. This parameter can be set to any capacity, of class portion, which processes the internal item in include space between two vector contentions. The different kernel functions used to check the most accurate results.

The Table 1 shows the accuracy of each kernel function on OCR dataset:

S. No.	Kernel Function	True	False	Accuracy
1.	Vanilladot	0.83925	0.16075	83.90%
2.	Polydot	0.83925	0.16075	83.90%
3.	Tanhdot	0.0845	0.9155	8%
4.	RBFdot	0.93125	0.06875	93.10%
5.	Laplacedot	0.8905	0.1095	89.05%
6.	Besseldot	0.70575	0.29425	70.5%

Table1

The following plot shows the accuracy of each kernel function in scatterplot:

ISSN: 0971-1260

Vol-22-Issue-3-July-2019

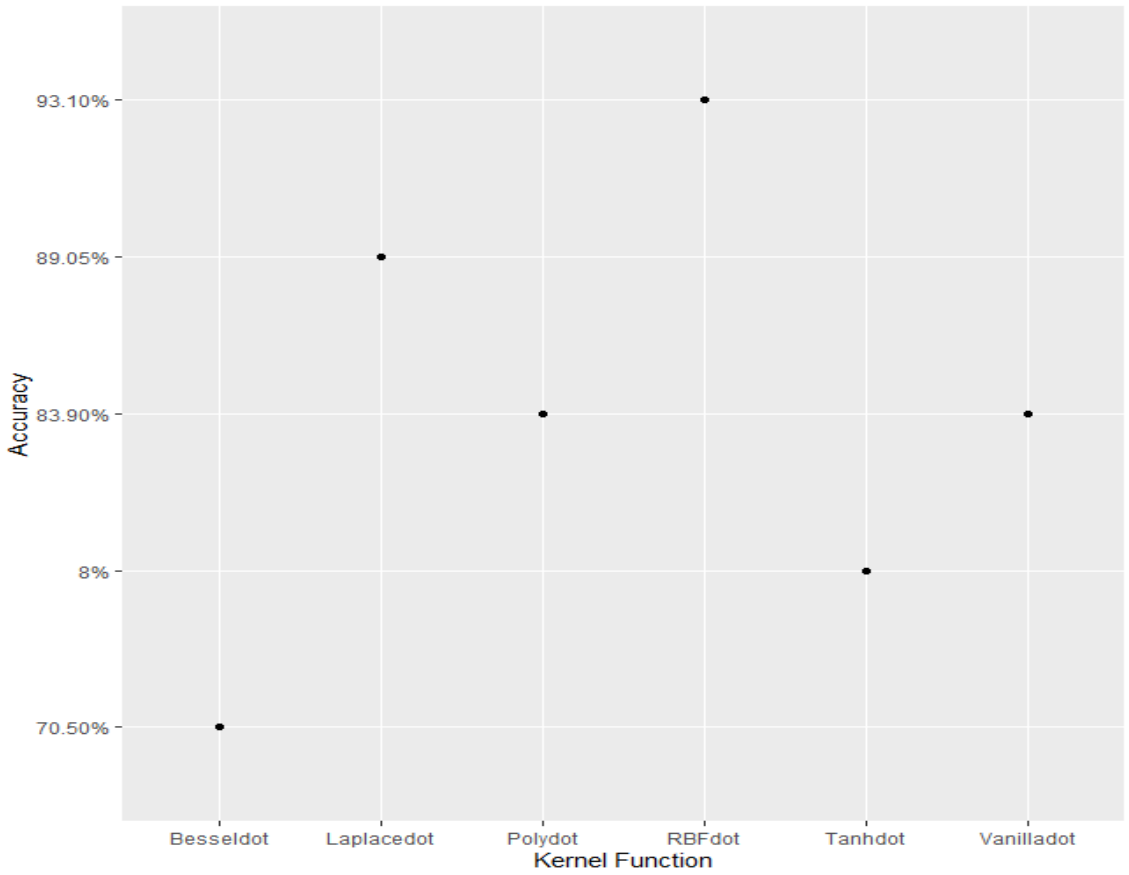

Fig 2: Kernel Accuracy

5. CONCLUSION

In this paper, we have demonstrated the outcomes utilizing different kernel functions. Fig 2 shows the results of information tests using various kernel function. The test outputs are empowering .We can see from the results that the decision of kernel function is basic for a given measure of information. It shows that the best one among all is RBFdot for the recognition of optical character.

REFERENCES:

[1] Boser, B. E., I. Guyon, and V. Vapnik , "A training algorithm for optimal margin classifiers" . In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pages. 144 -152. ACM Press 1992.

[2] V. Vapnik. "The Nature of Statistical Learning Theory". NY: Springer-Verlag. 1995.

[3] C. Hsu, C.C. Chang, and C.J. Lin. "A Practical Guide to Support Vector Classification" . Deptt of Computer Sci. National Taiwan Uni, Taipei, 106, Taiwan http://www.csie.ntu.edu.tw/~cjlin 2007

[4] C.-W. Hsu and C. J. Lin. "A comparison of methods for multi-class support vector machines". IEEE Transactions on Neural Networks, 13(2):415-425, 2002.

[5] C.-C. Chang and C. J. Lin (2001). LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm .

[6] L. Maokuan, C. Yusheng and Z. Honghai, "Unlabeleddata classification via SVM and k-means Clustering". Proceeding of the International Conference on Computer Graphics, Image and Visualization (CGIV04), 2004 IEEE.

[7] Z. Pawlak, "Rough sets and intelligent data analysis", Information Sciences 147 (2002) 1– 12.

[8] RSES 2.2 User's Guide Warsaw University http://logic.mimuw.edu.pl/»rses ,January 19, 2005

[9] E. Kovacs and L. Ignat, "Reduct Equivalent Rule Induction Based On Rough Set Theory", Technical University of Cluj-Napoca.

[10] RSES Home page http://logic.mimuw.edu.pl/»rses

