The Study of Support Vector Machine to Classify the Medical Data

Ghasem Farjamnia[*,1],Mehdi Zekriyapanah Gashti[2] ,Hamed Barangi[2], Yusif S. Gasimov[3,4]
[1]Institute of Applied Mathematics Baku State University, Baku, Republic of Azerbaijan
[2]Faculty of Engineering, Department of Computer Engineering, Payame Noor University, Tehran, I.R.IRAN
[3]Institute of Mathematics and Mechanics ANAS, Baku, Republic of Azerbaijan
[4]Azerbaijan University, Baku, Republic of Azerbaijan

Summary
Abstract In this article, we are going to study the linear support vectors and their performance in the related classification issues. Using the linear support vectors (SVM's) in the classification issues is a new approach that in recent years is considered by many scientists. It was used in a wide range of applications including OCR, Handwriting recognition, guidance signs diagnosis and etc. SVM approach is in a way that in the training phase, it is tried to choose the limit of decision-making (Decision Boundary) is such a way that its minimum distance to each of the considered categories stays maximum. This kind of choice helps our decision in practice to tolerate the noisy condition very well and has a good response. This way of selecting the boundary is based on the points that are named as support vectors. At first we study the concepts such as generalization of a pattern recognition machine and then the VC dimension that has a great application in the concept of classification machines. And then we describe the linear and non-linear support vectors and Kernel functions. And eventually, we will study the VC dimension for some of these functions.

Key words:
Support Vector Machine, VC Dimension, Mercer, Linear SVM, Nonlinear SVM RBF Kernel, Medical Data

1. Introduction

There is a large family of borders that specify the extent and generalizability and application of a learning machine. Now we will specify some of these measures.
Assume that we have l number of samples that each of them is linked as a pair of one vector and one category. These elements in every issue can represent something. For example, in the issue of tree recognition the vector can represent a pixel vector and the category 1 for (the cases that the picture includes the tree) the category -1 for (the cases that the picture does not include the tree), we use -1 instead of zero so we can perform easier in the calculations of the formulas [1, 2]. Assume that there is function of unknown probability distribution P(x,y) and the data are distributed according to this function. Our assumption is that the data are independent and evenly distributed. Now suppose that we have a machine that its task is to learn the $X_i \longmapsto y_i$ mapping. This machine in fact is defined by a complex of $X \longmapsto f(x,a)$ maps that the functions of f (x,a) are set by the parameter a . This machine is assumed as certain machine [3, 4]. For the input x and a clear choice of a it will always give the same output. A choice of special a , gives us a machine that we call it the taught machine [5]. For example, a neural network with a fixed architecture and structure, when a in it is correspond with the weights and biased values this is a learning machine. Therefore, the mean error for a trained machine is:

$$R(a) = \int \frac{1}{2} |y - f(\boldsymbol{X}, a)| dP(\boldsymbol{X}, y) \qquad (0)$$

2. VC Dimension

VC dimension is a feature of series of functions of {f(a)} that (in this function we have defined a as a general parameter that choosing it, will specify a special function) and can be defined for different classes of f function. Here we consider only the functions that are associated with the two-class pattern recognition $f(x,a) \in \{-1, 1\}$ $\forall$ x,a so that now a set of l points can be labeled as 2^l and for every per of labeling a member of the {f(a)} category can to be found that identifies the labels correctly. So we that set of points are made crushed by the set of functions [5]. The VC dimension for the set of functions of {f(a)} is defined as the maximum number of points which are crushed by these set of functions. Note that if the VC is equal to h, this means that there is at least one set of h points which can be crushed. But generally we cannot be sure that any set of h points can be crushed [6].

2.1. Friable points with the planes in the R^n environment

Assume that the studying environment is the R^2 environment and the set of {f(a)} functions include straight lines in this space [5]. That is for a straight line, all the located points on one side of the line belong to the class 1 of the opening and on the other side the points belong to the class -1. The direction of this attachment is

Manuscript received December 5, 2017
Manuscript revised December 20, 2017

also showed by an arrow on the line. However, three points that can be crushed by the set of lines will be found, but we cannot find four points with this feature. So the VC dimension of the straight line series in the R^2 environment equals to three (Figure 1).
Now consider the hyper planes that are located in the R^2 environment. The following theorem about that is true:

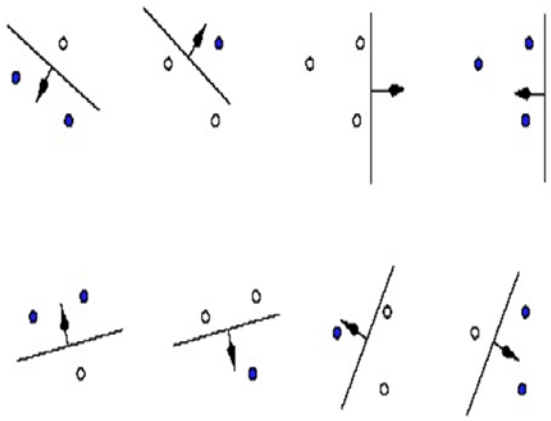

Fig. 1 A sample of studying the VC dimension in the R^n environment

Theorem1: Consider a set of m points in the R^n environment. Consider every arbitrary point as a source. In this case m points can be crushed by the hyper planes. If and only if the vector of remaining points is linearly independent.
From the above theorem we can conclude that the VC dimension in the R^n environment equals to n+1, because we can select n+1 point and put one of them as a source, and the remaining n points can be linearly independent from each other. But can never find n+2 points with this characteristic because on n+1 point is linearly independent in the Rn environment [6, 7].

2.2. VC Dimension and the Number of Parameters

VC dimension in fact shows the capacity of a set of functions [7]. You may expect a learning machine with a large number of parameters to have a high VC dimension and also a learning machine with a few numbers of parameters to have a low VC dimension but here for this issue we will show you a counterexample. We introduce a function that have infinite VC dimension (a set of classified functions with infinite VC dimension if for any l even very large one, could be able to crush the l point) $\theta(x)$,$x \in R$:{$\theta(x)=1 \ \forall \ x>0$; $\theta(x)=-1 \ \forall \ x \leq 0$} The step function is defined. Consider the class of the following one-parameter functions:

$$f(x,a) \equiv \theta(\sin(ax)), x, a \in \boldsymbol{R} \qquad (1)$$

Now if we are asked to choose l points that can be crushed, we can choose them as follows:

$$X^i = 10^{-i}, \ i=1,\ldots,l.'$$

And we can pick the tags in any way that we want:

$$y_1, y_2, \ldots, y_i, \quad y_i \in \{-1,1\}$$

Then, by choosing a as follows, f(a) will give us the way of labeling:

$$a = \pi\left(1 + \sum_{i=1}^{l} \frac{(1-y_i)10^i}{2}\right) \qquad (2)$$

The VC dimension of this machine is indefinite. Interestingly, although we can make the points that can be crushed very large, but there are four points that cannot be crushed [6, 7]. It is enough that these four points have equal distances and are in one line as shown in the figure 2.

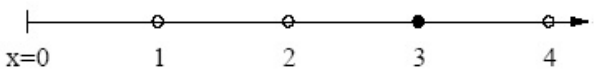

Fig. 2 The points with equal distance on a line.

3. Linear Support Vectors Machines

Begin with the simplest issue [8]: The linear machines that are trained on the separable data [9]. Again we label the training data in this way: $\{X_i, y_i\}, i=1,\ldots,l$, $y_i \in \{-1,1\}$, $X_i \in R^d$. Assume that we have hyper plane, which separates the positive samples from the negative samples. The x points that are located on this hyper plane satisfy the w.x+b=0 condition in which w is the normal vector of the hyper plane, |b| / ||w|| is the vertical distance from the hyper plane and ||w|| is the Euclidean normal w. suppose that d+(d-) is the least distance of the positive (negative) points from the hyper plane. The margin of a separator hyper plane is defined in this way. For the separable linear case, algorithm of the support vector will find the hyper plane with the largest margin. This issue can easily be formulated as follows: Suppose that all the training data satisfy the following constraints:

$$x_i.w+b \geq +1 \text{ for } y_i = +1 \qquad (3)$$

$$x_i.w+b \leq -1 \text{ for } y_i = -1 \qquad (4)$$

And the equations (3) and (4) can be summarized in the equation (5):

$$y_i(x_i.w+b)-1 \geq 0 \ \forall I \qquad (5)$$

Now consider the points that apply in the unequal equation (3). These points are on the H1: x_i . w+b=1 hyper plane with the normal vector w and |1 – b| / ||w|| vertical distance from the source. Similarly the points that apply in the unequal equation (4), are on the H2: x_i . w+b= -1 with the normal vector w and |-1 – b| / ||w|| distance from the source. So we have d+ = d- = 1 / ||w|| and the margin value equals to 2/||w||. Note that H1 and H2 are parallel (have equal norms) and no training point will stay between them. So we can find the paired hyper plane that gives us the biggest margin value that will be done by minimizing ||w||^2 according to the limits.

So we expect that usual two-dimensional solution case to have shape like the figure3. Those training points that apply in the equation (5) (It means that they are located on the H1 and H2 hyper planes) and removing them will change the founded solution, they are called the support vectors. These points are identified in Figure 3 with additional circles.

Now we go to formulate the LAGRANGE of the issue. We have two reasons for this. First, the limits shown in the equation (5) will be replaced with Lagrange multipliers that this will make our job so much easier; second, in such formulation of the issue the training data will be appeared only as a pointy multiplication of the vectors. This is a vital feature that allows us to extend the problem solution process to a non-linear state.

So we introduce the Lagrange coefficients، i = 1, …, l , a_i that each are for one of the limitations in the unequal equation (5). Recall that rule in this case is for the limitations in the form of Ci >= 0, the constraint equations are multiplied to the positive coefficients of Lagrange and will be subtracted from the objective function. For equality constraints the Lagrange multipliers will not be limited. This will give us the following equation:

$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} a_i y_i (x_i . w + b) + \sum_{i=1}^{l} a_i \quad (6)$$

Now we should minimize the LP according to the w, b and simultaneously it is needed that the LP derivatives disappear due to the (according to the limitations: a_i >= 0). This set of constraints is called C1. Now this is a quadratic convex problem because the objective function is convex itself. And the points that satisfy the restrictions will form a convex set (each linear limitation define a convex set, and a set of simultaneous N linear limitation define N convex set that the impact of this N convex set is a convex set itself). This means that we can deal with these two issues at the same time:

Maximum out the LP, according to the limitations that the gradient of LP with respect to w and b should be removed and also according to this limitation a_i >= 0 (this complex is called the C2 limitation set). This dual formulation of a problem is called Wolf Dual. This issue has a feature that the LP maximum, according to the C2 limitation will happen in the same values of w, b and a, and LP minimum happens according to the C1 limitation.

The need to remove the gradient of LP according to the w and b will give us the following condition:

$$w = \sum_i a_i y_i \boldsymbol{x}_i \quad (7)$$

$$\sum_i a_i y_i = 0 \quad (8)$$

Because of the fact that these limitations are the same in the dual formulation, we can substitute them in equation (6) so have the following equation:

$$L_D = \sum_i a_i - \frac{1}{2}\sum_{i,j} a_i a_j y_i y_j \boldsymbol{x}_i . \boldsymbol{x}_j \quad (9)$$

LP and LD of an objective function, but with different limitations are resulted. And solutions will be obtained by minimizing the LP or maximizing the LD. Note that if we formulate our issue with b = 0, which means that all the hyper planes will pass the source, the limitation (5) won't appear. This is a mild limitation for the high-dimensional spaces; this is because it will reduce the degree of freedom of the unit.

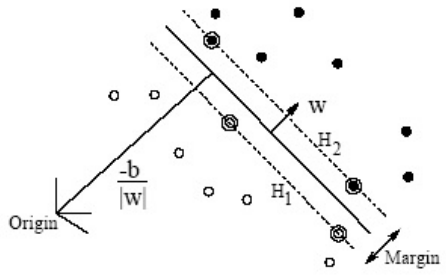

Fig. 3 Support Vector

So training the support vector (for the separable linear cases) with maximizing the LD according to the a_i that have positive values and limitations and according to the above solution will be performed. In the solution, the parts that we have a_i > 0 for them are called support vector. They are located on one of the H1 or H2 hyper planes. The rest of the training parts have the a_i = 0 value. The support vectors are the key elements of training complex for these machines. They are the closest to the decision border and if all the rest of the training points remove and the training repeats again still that mentioned separable hyper plane will be obtained.

4. Nonlinear Support Vector Machines

How the above methods can be extended for the case s that decision's functions is not a linear function of

the data and we could get the answer? We can show that an old trick can help us to do this easily. Firstl y [10, 11], note that the only problem of the data tha t appears in the training issue is only as a point mult iplication of Xi.Xj. Now suppose that we put the data in another environment named H (in some cases wit h infinite dimensions). As following:

$$\Phi : \boldsymbol{R}^d \mapsto H$$

Now the training algorithm by the point multiplication of the written data in the H, only depends on the data of the issue (as a $P(x_i).P(x_j)$). Now if we have kernel function under condition of $K(X_i, X_j) = \Phi(X_i).\Phi(X_j)$so it is only necessary that the kernel function to be used instead of the point multiplication in the learning algorithm. Now we do not even need to know exactly that our written function was what kind of function. A sample is as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\|x_i - x_j\|^2/2\sigma^2}.$$

In this example, H has infinite dimensions, therefore, working with the written function directly is not that much easy. But if we substitute the point multiplication of the written parts in the training algorithm with their kernel function (like the above equation) then we will have the support vector that is located in an environment with infinite dimensions. And to train that function, the same amount that was used to train the primary data (not written) is needed. All the rules and conditions of the last parts are still infeasible, because we are still performing a linear separation in a different environment.

But how can we apply this machine? We need w. but in the testing phase, SVM is applied by calculating the point multiplication of a part of x test with w, or more precisely, by calculating the sign of the following function:

$$f(x) = \sum_{i=1}^{N_s} a_i y_i \Phi(\boldsymbol{s}_i).\Phi(\boldsymbol{x}) + b = \sum_{i=1}^{N_s} a_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}) + b \quad (10)$$

That the Si are the support vectors. Therefore, we can avoid calculating the written function directly, and instead we can use the kernel function.

Suppose that the environment that our data are located in is called L, note that in addition to the fact that w is in H, totally there is no vector in the L that through our written function is written on the w. the important point is that we can find some Kernels that (For example, Kernels which are the function of point multiplication of the X_i s in the L), so the training algorithm and issue solution are depended from the L and H dimensions.

4.1. Conditions of Mercer

For which of the Kernels the { H ,Φ } pair exists that has the mentioned conditions in the previous parts and for which does not exist? The answer to this question is given by the Mercer condition [12]. A writing Φ and a development of:

$$K(x, y) = \sum_i \Phi(x)_i \Phi(y)_i \quad (11)$$

Exists, if and only if for every g(x) that

$$\int g(\boldsymbol{x})^2 d\boldsymbol{x}$$

Is finite, so

$$\int K(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0. \quad (12)$$

Note that for some specified cases, maybe studying the fact that the Mercer conditions are satisfied or not is not easy that much. But we can easily prove that this condition for the integral of the positive abilities of the point multiplication is infeasible. We should show that the:

$$\int \left(\sum_{i=1}^{d} x_i, y_i\right)^p g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0. \quad (13)$$

Is a phrase that through the development of $\left(\sum_{i=1}^{d} \boldsymbol{x_i y_i}\right)^{\boldsymbol{p}}$ will be obtained, and it is in the following form:

$$\frac{p!}{r_1! r_2! \dots (p - r_1 - r_2 \dots .)!} \int {x_1}^{r_1} {x_2}^{r_2} \dots . {y_1}^{r_1} {y_2}^{r2} \dots \, g(\boldsymbol{x}) g(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}$$

The above equation (12) is going to be as follows:

$$= \frac{p!}{r_1! r_2! \dots (p - r_1 - r_2 \dots .)!} \left(\int {x_1}^{r_1} {x_2}^{r_2} \dots \, g(\boldsymbol{x}) d\boldsymbol{x} \right)^2 \geq 0.$$

A simple conclusion is that every kernel function can be written in this way: $K(x, y) = \sum_{p=0}^{\infty} c_p (x.y)^p$

In which the C_P s are the positive coefficients and also the Seri is convergent. So the Mercer condition is satisfied.

If someone use the Kernel function that does not satisfy the Mercer condition what will happen? The answer is that in this condition we may deal with a second-degree issue of planning that cannot be solved. However, even for the kernel functions that does not meet the Mercer conditions, there is possibility that the trainings algorithms gets converge and we could achieve the answer. But there is no guarantee for this condition.

5. VC Dimension of the Support Vector Machines

Now we show you that the VC dimension of the support vector machines can be very large (even infinite). Then this discussion arises that, despite these features why SVM has good generalization ability? It should be noted that currently there is no guarantee to show that a set of SVM for specific categories of issues have a very high accuracy [13, 14]. Here any Kernel function that satisfies the Mercer conditions is positive Kernel and the linked H environment of that is called improviser embedding space. Also, the embedded space with minimum dimensions for the Kernel function is called minimal embedding space. Now we introduce the following theorem:

Theorem: Suppose that K is a positive kernel function that is corresponding to the minimum space improviser of H. in this case, the VC dimension of the support vector machine corresponding with that equals to dim(H) + 1.

Proof: if the minimal space improviser has the d dimension, then d points in picture L can be found under the written function of L to H that the positive vectors in H are linearly independent. These vectors can be crushed in the hyper planes in H space. Therefore, by limiting the SMV to the separable issue situation, or for the case where the penalty parameter C can get any values, the support vector machines family with K Kernel can crush these points so its VC dimension is d + 1.

5.1. VC Dimensions Linked to the RBF Kernels

In this case, we only talk about the related theorem to the VC dimension in the RBF (Radial Basis Function) Kernels and we won't discuss its proof.

Theorem: if we have the Mercer Kernels class and instead of $\|x1 - x2\| \rightarrow \inf$ we have $K(x_1,x_2) \rightarrow 0$ and also we have K(x,x) of the order of 1, assume that the data are randomly chosen from the R^d environment. In this case, the classifier families including the support vector machines that use these Kernels, and they can get the C penalty amount that gets any value, they have the infinite VC dimension.

6. Conclusion

When the solution of a problem is the comprehensive support vector training and when it is unique?

When we say comprehensive, we mean that there is no points in our decision-making area. In which our objective function can take a smaller amount. We will show you two ways in which you may not get a unique answer: Solutions that {w,b} are unique in it, but w is not unique for their development. And solutions in which {w,b} are different (not unique). Both of these cases are interesting. Even if the pairs {w,b} were unique and a_i s are not unique, it is possible that there will be a equational developments of the w which requires less support vectors. And therefore it will require a smaller number of operations in the testing phase. This means that any local solution can be a comprehensive solution. This feature is one of the characteristics of any convex programming problem. Moreover, if the objective function is clearly convex the solution also will be definitely unique. Finding the solutions that because of the lack of uniqueness of a_i in the development of w, are not unique is very easy [6]. For example, consider the issue about the 4 inseparable points on four vertices of a square in a two-imensional space. Assume that these points respectively are $x_1=[1,1]$, $x_2=[-1,1]$, $x_3=[-1,-1]$, $x_4=[1,-1]$ and with polarity of [+,-,-,+].

A solution to this is that; w=[1,0] , b=0 ,

a=[0.25,0.25,0.25,0.25]. Another solution can have the same w and b, but we have a=[0.5,0.5,0,0].

The problem of optimization the support vectors can be solved only through analytical methods if the number of the data is too small or for the separable cases, we should already know that which data are support vectors. We should note that, this issue only happens when the problem has the symmetry. For the case of separate analysis, computational complexity in the worst case is proportional to Ns to the power of 3 in which Ns is the number of support vectors.

In many real problems of the previous equations (Where the point coefficients have been replaced with kernel functions) must be solved using numerical methods. For the small problems, any optimization method that can solve the quadratic convex planning issues with linear limitation is suitable.

Now we show you that the VC dimension of the support vector machines can be very large (even infinite). Then this discussion arises that, despite these features why SVM has good generalization

References

[1] V. Vapnik, Statistical Learning Theory, John Wiley, 1998.
[2] B. Schölkopff, A.J. Smola, Learning with Kernels, The MIT Press, 2002.
[3] BURGES, Christopher J. C., A Tutorial on Support Vector Machines for Pattern Recognition, 1998.
[4] C.J.C. Burges, "Simplified Support Vector Decision Rules," Proc. Int'l Conf. Machine Learning, 1996.
[5] RIFKIN, Ryan, Current Topics of Research III: Theory And Implmentation Of Support Vector Machines RifkinKanski, J.J., Clinical ophthalmology. 6th edition, London: Elsevier Health Sciences (United Kingdom). 2007.
[6] R.N. Karasev, Transversals for families of translates of a two-dimensional convex compact set, Discrete Comput. Geom. 24 (2–3) (2000) 345–353. The Branko Grünbaum birthday issue.
[7] Eric Martina, Arun Sharmab, Frank Stephan "On ordinal VC-dimension and some notions of complexity", Theoretical Computer Science 364 (2006) 62 – 76.

[8] T. Joachims, “Training Linear svms in Linear Time,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, 2006.
[9] Y. Shao, C. Zhang, X. Wang, N. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.
[10] K.Q. Shen, C.J. Ong, X.P. Li, Einar P.V. and Wilder-Smith, Feature selection via sensitivity analysis of SVM probabilistic outputs. Machine Learning, 2008, 70:1-20.
[11] B. Scho¨lkopf, A. Smola, and K. Mu¨ ller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” Neural Computation, vol. 10, no. 5, pp. 1299-1319, 1998.
[12] Binbin Pana, Jianhuang Laib, Wen-Sheng Chenc, “Nonlinear nonnegative matrix factorization based on Mercer kernel construction “,Pattern Recognition , Semi-Supervised Learning for Visual Content Analysis and Understanding, Volume 44, Issues 10–11, October–November 2011, Pages 2800–2810.
[13] Ujwala Ravale, Nilesh Marathe, Puja Padiya, “Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function”, International Conference on Advanced Computing Technologies and Applications ICACTA-2015. Procedia Computer Science 45 (2015) 428 – 435.
[14] Mehdi Zekriyapanah Gashti, “A novel hybrid support vector machine with decision tree for data classification”, International Journal of Advanced and Applied Sciences, Volume 4, Issue 9, September 2017, Pages 138-143.