Generalized Chebyshev Kernels for Support Vector Classification

Sedat Ozer[1], Chi Hau Chen[2]

[1] Department of Electrical & Computer Engineering, Medical Imaging Research Center
Illinois Institute of Technology, Chicago, IL
sozer1@iit.edu

[2] Department of Electrical & Computer Engineering
University of Massachusetts, Dartmouth, N. Dartmouth, MA
cchen@umassd.edu

Abstract

In this paper, a method to generalize previously proposed Chebyshev Kernel function is presented for Support Vector Classification in order to obtain more robust and higher classification accuracy. By introducing the generalized Chebyshev polynomials for vector inputs, we increase the performance of this kernel function. The simulation results show that the proposed generalized Chebyshev Kernel has better performance than the previously proposed kernel for Support Vector Classification. Early simulation results show that the proposed kernel function yields the best classification results for a Breast Cancer dataset.

1. Introduction

Support Vector Machine (SVM), is a well known learning algorithm that has been commonly used in many pattern recognition applications including face recognition, cancer detection, and target tracking [1], [2], [3].

Although SVM is considered a "state of the art" learning algorithm, its performance highly depends on the kernel function to transform input data into a higher dimensional space and the cost function that needs to be minimized in order to find the best coefficients [4].

The kernel function plays an important role for SVM, as it maps the nonlinear data structure into a higher dimensional space where the data can be linearly separable, [4], [5]. An ideal kernel function should not require any parameter while giving good performance for all types of data. Gaussian kernel function is widely being used in many applications by requiring only one parameter. However, choosing this kernel parameter for optimal performance is another problem and there are various approaches proposed to find this optimal parameter [6]. Finding this optimal kernel parameter can require additional computation including optimization which is time and power consuming.

Recently Chebyshev kernel function is proposed for SVM and it is proved that it is a valid kernel function for scalar valued inputs [7]. However, in almost any pattern recognition application, the inputs are in vector form. And generally, instead of applying kernel functions on each element of the vectors, applying them to the input vectors directly yields better result, as the kernel functions are supposed to give a measure of how given two vectors are similar to each other by transforming the inner product of these two vectors into a higher dimensional space [8], [9].

Therefore in this study we propose generalized Chebyshev kernel functions by introducing the vector Chebyshev polynomials. The resulting new kernel function gives its best performance, based on the simulation results, for certain values of kernel parameters between 3 and 6. This preliminary result of generalized Chebyshev kernel, can make this kernel function to be called semi parametric kernel function.

In simulation results we show that, the resulting new kernel function works better than previously proposed Chebyshev kernel function and interestingly it always approaches the minimum support vector (SV) number in every test when compared to the other common kernel functions. This property of the generalized Chebyshev kernel function can also be useful for the applications where SV number is important.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2. Support Vector Machines

SVM is a linear classifier which classifies nonlinear data in a higher dimensional space where the data can be linearly separable. In order to classify the data SVM uses the following formula [5], [8]:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \quad (1)$$

where k is the support vector number, b the bias value, α_i the corresponding Lagrange multiplier for the support vector $\mathbf{x_i}$, y_i the class label for the corresponding support vector, and $K(\mathbf{x},\mathbf{x_i})$ the kernel function to map the input vectors onto a higher dimensional feature space where the data can be linearly separable.

In training step, the nonzero α_i can be found by maximizing the formula $w(\alpha)$:

$$w(\alpha) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (2)$$

subject to $\sum_{i=1}^{s} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, where s is the number of training samples. Thus the input vectors $\mathbf{x_i}$, with nonzero α_i values, are called support vectors.

3. Generalized Chebyshev Polynomials

The orthogonal set of Chebyshev polynomials is denoted by $T_n(x)$ n=0,1,2,3,… for the x values between [-1,1]. The first kind of Chebyshev polynomials of order n, $T_n(x)$, is defined as:

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \quad (3)$$

The Chebyshev Polynomials of the first kind are orthogonal with respect to the weighting function $\frac{1}{\sqrt{1-x^2}}$, therefore, integrating two given Chebyshev polynomials, with respect to the weighting function within the interval [-1,1], yields:

$$\int_{-1}^{1} T_i(x) T_j(x) \frac{1}{\sqrt{1-x^2}} dx = \begin{cases} 0 & i \neq j \\ \pi/2 & i = j \neq 0 \\ \pi & i = j = 0 \end{cases} \quad (4)$$

Thus if we need to rewrite the left hand side of equation (4), for discrete case, we have:

$$\left\langle T_i(x), T_j(x) \frac{1}{\sqrt{1-x^2}} \right\rangle \quad (5)$$

where <***a*,*b***> is defined as the inner product of vector ***a*** and vector ***b***.

In this study we extend the use of Chebyshev polynomials for vector inputs by defining generalized Chebyshev Polynomials. For a given row vector **x**, we define Chebyshev polynomials as:

$$T_0(\mathbf{x}) = 1$$
$$T_1(\mathbf{x}) = \mathbf{x} \quad (6)$$
$$T_n(\mathbf{x}) = 2\mathbf{x}T_{n-1}^T(\mathbf{x}) - T_{n-2}(\mathbf{x})$$

where $T_n(\mathbf{x})$ is the n^{th} order Chebyshev polynomial for a given input vector **x**, and $T_{n-1}^T(\mathbf{x})$ is the transpose of $T_{n-1}(\mathbf{x})$.

4. Generalized Chebyhsev Kernels

By using kernel functions, we can construct nonlinear SVM classifiers. As the SVM algorithm is a linear classifier, we need to transform the nonlinear data into another space, where the data can be linearly separable in order to obtain satisfactory classification results with SVM. If this transformation function is defined as $\Phi(\mathbf{x})$, then a kernel function $K(\mathbf{x},\mathbf{x_i})$, yields the inner product of given two vectors in this higher dimensional space as:

$$K(\mathbf{x}, \mathbf{x_i}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x_i}) \rangle \quad (7)$$

In this paper we present the generalized Chebyshev kernel functions by extending the idea behind the previously proposed Chebyshev kernel functions [7].

Thus by using generalized Chebyshev polynomials, a generalized n^{th} order Chebyshev kernel function can be defined as:

$$K(\mathbf{x}, \mathbf{x_i}) = \frac{\sum_{j=0}^{n} T_j(\mathbf{x}) T_j^T(\mathbf{x_i})}{\sqrt{a - \langle \mathbf{x}, \mathbf{x_i} \rangle}} \quad \text{where } a=m. \quad (8)$$

In equation (8), $T(\mathbf{x})$ and $T(\mathbf{x_i})$ are defined as:

$$T(\mathbf{x}) = [T_0(\mathbf{x}), T_1(\mathbf{x}), ..., T_n(\mathbf{x})] \quad (9)$$
$$T(\mathbf{x_i}) = [T_0(\mathbf{x_i}), T_1(\mathbf{x_i}), ..., T_n(\mathbf{x_i})] \quad (10)$$

where **x** and $\mathbf{x_i}$ are m dimensional vectors.

As an instance, *the 4th order Generalized Chebyshev Kernel* can be written as:

$$K(\mathbf{x}, \mathbf{x_i}) = \frac{1 + c + (2a-1)(2b-1)}{\sqrt{m-c}} + \frac{c(4a-3)(4b-3)}{\sqrt{m-c}} + \frac{(8a^2-8a+1)(8b^2-8b+1)}{\sqrt{m-c}} \quad (11)$$

where a=<**x**, **x**>, b= <$\mathbf{x_i}$, $\mathbf{x_i}$> and c=< **x**, $\mathbf{x_i}$ >.

5. SVM Kernel Construction

In order to be a valid SVM kernel, the kernel function needs to satisfy Mercer Condition [5], [8]. If we write the generalized Chebyshev Kernel function in the form of

$$K(\mathbf{x},\mathbf{x_i}) = K_{(1)}(\mathbf{x},\mathbf{x_i})K_{(2)}(\mathbf{x},\mathbf{x_i}) \quad (12)$$

where

$$K_{(1)}(\mathbf{x},\mathbf{x_i}) = \sum_{j=0}^{n} T_j(\mathbf{x})T_j^T(\mathbf{x_i}) \quad (13)$$

then, we can easily prove that the kernel $K_{(1)}(\mathbf{x},\mathbf{x_i})$ satisfies the Mercer conditions [7], [9].

The kernel function, $K_{(2)}(\mathbf{x},\mathbf{x_i})$ is a function of inner product $z=<\mathbf{x},\mathbf{x_i}>$, we can simply find its Maclaurin expansion. If all the Maclaurin coefficients are nonnegative for the expansion, then this kernel will be a valid kernel [8].

$$K_{(2)}(\mathbf{x},\mathbf{x_i}) = \frac{1}{\sqrt{m-<\mathbf{x},\mathbf{x_i}>}} = K(z) = \frac{1}{\sqrt{m-z}} \quad (14)$$

The Maclaurin expansion of $K_{(2)}(\mathbf{x},\mathbf{x_i})$ is:

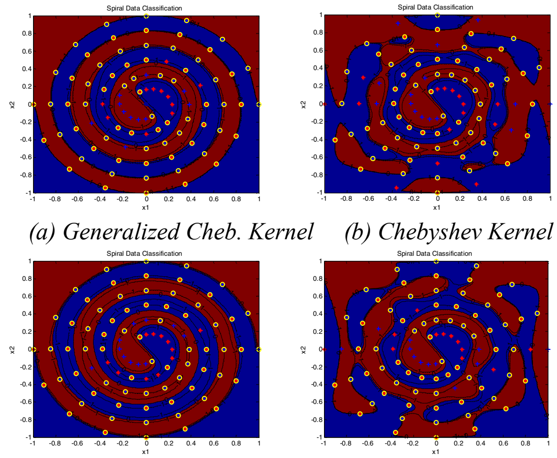
$$K_{(2)}(\mathbf{x},\mathbf{x_i}) = \frac{1}{\sqrt{m}} + \sum_{j=1}^{\infty}\left(\frac{<\mathbf{x},\mathbf{x_i}>^j}{m^{(2j+1)/2}2^j j!}\prod_{k=1}^{j}(2k-1)\right) \quad (15)$$

As $m \geq 1$, all the coefficients are nonzero, and therefore, $K_{(2)}(\mathbf{x},\mathbf{x_i})$ is a valid function. As a result, as multiplication of two valid kernels is also a valid kernel [8], the resulting kernel function, $K(\mathbf{x},\mathbf{x_i})$, is also a valid SVM kernel function.

6. Simulation Results

Here we perform some tests to compare the proposed kernel with its previous counterpart as well as the Gaussian kernel [5], [9] function. For testing we first test the kernel functions on spiral dataset. This toy dataset is good for visualization of the kernel functions' performance. The dataset has 92 data where each data is two dimensional. Each class has 46 data. After normalizing the data between [-1, 1], we used all 92 data for training and created a grid dataset between [-1, 1] to see generalization performance of the kernels.

As can be seen from Figure.1, the Generalized Chebyshev Kernel results show better generalization ability than previously proposed Chebyshev kernel. In Figure.1 (a), the 4th order generalized Chebyshev kernel, found 70 SVs while the 4th order Chebyshev kernel found only 62 SVs as shown in Figure.1 (b). The Gaussian kernel found 72 SVs with σ=0,33. However it can be clearly seen that the generalized Chebyshev kernel shows the best generalization performance with the least SVs. The 6th order Chebyshev kernel, in Figure.1 (d), found 68SVs. However its generalization performance is still not satisfactory, when compared to (a) and (c).



(a) Generalized Cheb. Kernel *(b) Chebyshev Kernel*

(c) Gaussian Kernel *(d) Chebyshev Kernel*

Figure.1 *SVM generalization for various kernel functions*

Another test dataset is the Image segmentation dataset currently available at [10]. The dataset contains 7 different classes of images formed of 210 data for training and another 2100 data for testing. Each data has 18 features. We used one against all approach for multiclass classification.

Table.1 *Image Segmentation Dataset Classification results*

	SV no / Test %	SV no / Test %	SV no / Test %	SV no / Test %	SV no / Test %	Best σ value
	Cheb. Kernel (*n*=6)	Cheb. Kernel (*n*=4)	Gen. Cheb. (*n*=6)	Gen. Cheb. (*n*=4)	Gaussian Kernel	
Window:	43 / 92.00 %	38 / 91.19%	30 / 92.19%	32 / 92.48%	122 / **94.57%**	σ=0.40
Path:	40 / 95.81 %	40 / 95.29%	13 / 98.90 %	11 / 99.19%	106 / **99.71%**	σ=0,43
Sky:	20 / 99.29%	18 / 99.38%	**05 / 100 %**	08 / 100%	18 / 100%	σ=1.45
Foliage:	50 / 93.95%	46 / 93.95%	30 / 96.20 %	30 / **96.90%**	28 / 96.71%	σ=1.41
Cement:	58 / 91.95%	54 / 91.00%	30 / 92.19%	31 / 92.33%	84 / **96.95%**	σ=0.53
Brickface	33 / 97.33%	32 / 97.14%	15 / 98.62%	14 / 98.71%	93 / **99.48%**	σ=0.44
Grass:	24 / 99.57%	20 / 99.52%	**11 / 99.86%**	15 / 99.86%	18 / 99.86%	σ=1.53

On Table1, the best testing performance value with the least SV numbers are shown in bold. Although in every experiment, the generalized Chebyshev kernel yielded a better testing performance than the Chebyshev kernel, the Gaussian kernel showed the best performance on average for this dataset. When both generalized Chebyshev kernel and Gaussian kernel found the same testing performance, the generalized Chebyshev kernel yielded the least SV number.

We also tested on the Wisconsin breast cancer dataset from the UCI repository [10]. There are two classes in the dataset and the entire dataset contains 569 data, where each data contains 30 features. We used the first 50 data from each class for training and the remaining data for testing.

Table2: *Breast Cancer Dataset Classification Results*

Kernel Function:	Chebyshev	Gen. Cheby.	Gaussian
Breast Cancer % :	95.95	**97.23**	95.52
Best Kernel Param.:	n=0	n=3	σ=12
SV no:	30	30	30

On table 2, only the best performance values with the corresponding kernel parameters are shown for each kernel. The generalized Chebyshev kernel found the highest percentage at overall.

7. Conclusion

In this paper we propose a new kernel function set by introducing the vector Chebyshev polynomials. When compared to the previously proposed Chebyshev kernel, generalized Chebyshev kernel function yields better performance in general as it works on the vector inputs directly, instead of being applied onto each element.

Normalization takes an important role for generalized Chebyshev kernel, and therefore the entire data, must be normalized between [-1,1] before using the kernel function. Also, in order to avoid division by zero, a small ε value can be used at the denominator. Based on the simulation results, we can say that choosing n, from an integer set between 3 and 6 is usually enough to obtain a good classification result from the generalized Chebyshev kernel function.

According to the test results, the generalized Chebyshev kernel finds the least number of SVs on almost every test. Intuitively, we think this comes from the orthogonality property of the Chebyshev polynomials. This property of the kernel function can be important and useful in some applications where the SV number is highly important as in feature selection.

Test results show that the proposed kernel function yields the best performance for Breast Cancer dataset. These preliminary results indicate that the generalized Chebyshev kernel separates the malignant and benign tissues the best in the high dimensional feature space. Thus generalized Chebyshev kernel functions can be considered as a good alternative to the Gaussian kernel function for some specific datasets as in breast cancer dataset.

A more comprehensive test can be done in the future work by using different datasets.

8. Acknowledgement

The authors would like to thank to Mr. Ye Ning and Mr. Sun for providing the Chebyshev kernel MATLAB code, which they used in [7] for simulations.

9. References

[1] Z. Li; X. Tang, “Using Support Vector Machines to Enhance the Performance of Bayesian Face Recognition”, *Trans. on Information Forensics and Security*, IEEE, June 2007, Vol. 2, no: 2, pp.174 - 180.

[2] I. El-Naqa; Y. Yang; M.N. Wernick; N.P. Galatsanos; R.M. Nishikawa, “A support vector machine approach for detection of microcalcifications”, *Trans. on Medical Imaging*, IEEE, Dec. 2002, Vol. 21, no: 12, pp. 1552 - 1563.

[3] S. Avidan, "Support vector tracking", *Trans. on Pattern Analysis and Machine Intelligence*, IEEE , Vol. 26, no: 8, 2004, pp. 1064 - 1072.

[4] P.H. Chen; C.J. Lin; B. Schölkopf, “A tutorial on nu-support vector machines.”, *Applied Stochastic Models in Business and Industry*, 2005, pp. 111-136.

[5] V.N. Vapnik, “The Nature of Statistical Learning Theory”, 2nd edn, Springer-Verlag, New York, 1998.

[6] L.P. Bi; H. Huang; Z.Y. Zheng; H.T. Song, “New heuristic for determination Gaussian kernels parameter”, *Machine Learning and Cybernetics*, pp. 4299 - 4304 Vol. 7, 2005.

[7] N. Ye; R. Sun; Y. Liu; L. Cao, "Support vector machine with orthogonal Chebyshev kernel", *The 18th International Conference on Pattern Recognition (ICPR'06),* 0-7695-2521-0/06, 2006.

[8] B. Scholkopf; A. J. Smola, “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, MIT Press, 2001.

[9] S. Ozer, “On the Classification Performance of Support Vector Machines using Chebyshev Kernel Functions”, M.Sc. Thesis, University of Massachusetts, Dartmouth, MA, 2007.

[10] A. Asuncion; D.J. Newman, “UCI Machine Learning Repository”, Irvine, CA: University of California, School of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html].