Principles and Rules Governing the Construction of the Kernel Functions in Support Vector Machine

Orazanbox

orazanbox@gmail.com

Abstract: Support Vector Machine (svm) is a favorable method for categorizing different types of data. The main problem with this method is significant decrease of classification speed against increase in the problem magnitudes, that the magnitude of the problem is positively correlative to the number of the support vectors. Thus changing, selecting, or modifying the Kernel function can account for a part of this speed reduction, and pave the way of solving the problem. In this article, we introduce the common kernel functions and kernels derived from orthogonal polynomials, and investigate the principles and rules governing the construction of the kernels.

[Orazanbox. **Principles and Rules Governing the Construction of the Kernel Functions in Support Vector Machine.** *Academ Arena* 2017;9(5):111-115]. ISSN 1553-992X (print); ISSN 2158-771X (online). http://www.sciencepub.net/academia. 6. doi:10.7537/marsaaj090517.06.

Keywords: support vector machine, categorization, kernel function

1. Introduction

A Russian researcher named Vladimir Vapnik took an important step in classifiers establishing firmly the theory of statistical learning, and proposing SVM on this basis [1]. SVM is one of relatively new methods, which has shown good efficiency compared to more conventional methods of classification like Perception Neural Network.

This algorithm belongs to supervised classification algorithms, which predicts the class or group for a particular sample, which is often used for binary classification. SVM can be employed wherever recognizing pattern or classification of things is needed. In order to discriminate between two classes, this algorithm uses one page (plate) in a way that the page has the maximum distance to the both classes. The nearest train samples to this page are called support vectors [2].

2. Statement of the problem and aims of the research

Recently, the application of support vector machine (SVM) for solving classification problems has grown substantially. In SVM problems, consider a case where the data are intricate, and classification by one line is not possible for the data. (For example, in recognition of individuals' faces where the face, eyes, ears, and eyebrows have different, nonlinear patterns).

If linear discriminator does not account for our problem, what solution do we have? Therefore nonlinear linear discriminator would be proposed which contains the important subject of Kernel functions.

Note the nonlinear pattern for discriminating data below.

It is now obvious that the aim of proposing Kernel function is decreasing support vectors and improving the accuracy of classification. However, it should be noted that the decrease in the number of support vectors can result in lesser accuracy of classification. Since the accuracy of classification resulting from SVM varies depending on the type of Kernel function, optimization of every kind results in making the way for solving the problem [3].

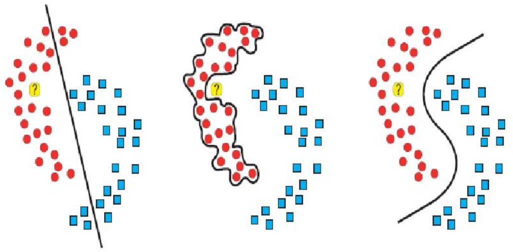

Figure 1. Data differentiation in on linear and 2 nonlinear cases

3. kernel function

Kernel methods have gained increased popularity in the machine learning community in recent years. Basically, one of the privileges of SVM is nonlinear mapping of the vector entering into a feature space with large dimensions which is hidden from ingoing and outgoing perspectives. This is implemented by Kernel functions. Kernel functions play an important role in their classifying capability. The way we modify Kernel-function parameters for the best is of utmost importance.

Appropriate adjustment of kernel parameters can play an important part in its accuracy and validity. Because the input set affects kernel parameter selection. Ideal Kernel function determines the result of similarity between two objects which depends on a class with the size of the two objects from different classes. Implicit mapping by kernel functions causes close similarity among objects, and dissimilar objects are separate from each other in the provided feature space. Nevertheless, appropriate selection of input features is also an important issue in classification procedure [1], [2].

3-1. common kernel functions [4], [5]

Kernel functions can be used in many applications. Several types of kernel functions are commonly used: Linear, Radial Basis Function, polynomial and Multi-Layer Perceptron.

3-1-1. linear kernel

The linear kernels is defined as follows:

$$k(x_i, x_j) = < x_i, x_j > \quad (1)$$

3-1-2. Radial Basis Function (RBF) or Gaussian Kernel

The RBF kernels take this form:

$$k(x_i, x_j) = e^{-\frac{1}{2\sigma^2}\|x_i - x_j\|^2} \quad (2)$$

Where σ is the width of the radial basis function.

3-1-3. polynomial kernel

The Polynomial kernels take this form:

$$k(x_i, x_j) = (x_i^T x_j + c)^d \quad (3)$$

Where *d is* the degree of the polynomial.

Note

- If c=0, the kernel is called homogeneous.
- If c=1, the kernel is called nonhomogeneous.
- If d=1. We obtain linear kernel.

3-1-4. Multi-Layer Perceptron kernel (MLP)

The MLP kernels is defined as follows:

$$k(x_i, x_j) = \tanh(\beta_0 + \beta_1 x_i^T x_j) \quad (4)$$

Note that using this kernel is not simple. Because β_0, β_1 is not implement for every value.

3-2. producing new kernel

Kernels can combine through specific operators to make more complicated kernels as well. But for producing kernel we encounter the crucial issue of Gramian matrix. This matrix, under certain circumstances mentioned below, verifies the credibility of produced kernels. There are various way to verify kernel validity [4]:

- Prove its positive definiteness (difficult).
- Find out a corresponding feature map.
- Use kernel combination properties (we'll see).
- Use Mercer's theorem.

3-2-1. Gramian matrix [4], [8]

Given a set V of m vectors (points in R^n), the Gram matrix G is the matrix of all possible inner products of V, i.e. $G_{ij} = V_i^T V_j$. Let $V = \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ a set of input vectors, then the Gram Matrix K is defined as:

$$k = \begin{pmatrix} \langle \phi(v_1).\phi(v_1) \rangle & \ldots & \langle \phi(v_1), \phi(v_n) \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi(v_n).\phi(v_1) \rangle & \cdots & \langle \phi(v_n).\phi(v_n) \rangle \end{pmatrix}$$

For example, the Gram matrix of (1, 2) and (1, -1) is:

$$\begin{pmatrix} (1,2)(1,2) & (1,2)(1,-1) \\ (1,2)(1,-1) & (1,-1)(1,-1) \end{pmatrix} = \begin{pmatrix} 5 & -1 \\ -1 & 2 \end{pmatrix}$$

3-2-2. Necessary Definitions [5], [7]

- A symmetric $m \times m$ matrix K is positive definite (pd), if $\sum_{i,j=1} c_i c_j k_j \geq 0 \quad \forall c \in \mathbf{R}^m$ (5).
- If equality only holds for c = 0, the matrix is strictly positive definite (s.p.d).
- Alternative conditions:
- ▪ All eigenvalues are non-negative (positive for s.p.d.).
- ▪ There exists a matrix B such that $k = B^T B$.
- Positive definiteness is necessary and sufficient condition for a kernel to correspond to a dot product of some feature map Φ .
- A symmetric function $k : X \times X \to \mathbf{R}$ which for all $m \in \mathbb{N}$, $x_i \in X$ gives rise to a positive definite Gram matrix, i.e. for which for all $c_i \in \mathbf{R}$ we have

$$\sum_{i,j=1}^{m} c_i c_j k_{ij} \geq 0$$

Where $k_{ij} := k(x_i, x_j)$, is called a positive definite (p.d) kernel.

- A symmetric function $k : X \times X \to \mathbf{R}$ which satisfies previous relation for all $m \in \mathbb{N}$, $x_i \in X$ and for all $c_i \in \mathbf{R}^m$ with $\sum_i^m c_i = 0$ is called a conditionally positive definite (c.p.d) kernel.
- The Cauchy-Schwarz inequality for kernels is:

$$\begin{aligned} k(x, x')^2 &= \langle \Phi(x).\Phi(x') \rangle \\ &= \|\Phi(x)\|^2 \|\Phi(x')\|^2 \\ &= \langle \Phi(x).\Phi(x) \rangle \langle \Phi(x').\Phi(x') \rangle \\ &= k(x, x) k(x', x') \end{aligned} \quad (6)$$

- Symmetry properties:

$$k(x, z) = \langle \Phi(x).\Phi(z) \rangle = \langle \Phi(z).\Phi(x) \rangle = k(z, x)$$

(7).

- Kernel function as similarity measure between input objects. Gram Matrix (Similarity

/Kernel Matrix) represents similarities between input vectors.

3-2-3. Mercer's theorem [4]

By using this theorem, we can test whether it is a kernel function.

Assume that:

- $X=\{x_1,\ldots,x_n\}$ be finite input space
- K (x, z) on X be a symmetric function
- Gram Matrix $K=(K(x_i,x_j))_{i,j=1}^{n}$ (8)
- since K is symmetric there exists an orthogonal matrix V s.t $K=V\Lambda V'$ (9)
- diagonal Λ containing eigenvalues λ_i of K.
- and eigenvectors $v_t=(v_{ti})_{i=1}^{n}$ as columns of V.
- all eigenvalues are non-negative and let feature mapping be

$$\phi : x_i \mapsto (\sqrt{\lambda_i}v_{ti})_{i=1}^{n} \in \mathbf{R}^n, i=1,\ldots,n. \quad (10)$$

Then:

$$\langle \Phi(x_i),\Phi(\mathrm{z_j})\rangle = \sum_{i=1}^{n}\lambda_i v_{ti}v_{tj} = (V\Lambda V')_{ij} = K_{ij}=k(x_i,x_j). \quad (11)$$

- The kernel matrix is symmetric positive definite.
- Any symmetric, positive definite matrix can be regarded as a kernel matrix; that is, there exists a Φ such that: $\mathrm{K(\,x,z)}=\langle\Phi(x),\Phi(\mathrm{z})\rangle$ (12).

Note

- Every Gram Matrix is symmetric and positive semi-definite (s.p.s.d).
- Every s.p.s.d matrix can be regarded as a Kernel Matrix, i.e. as an inner product matrix in some space.
- diagonal matrix satisfies Mercer's criteria, but not good as Gram Matrix.
- Every similarity matrix can be used as kernel (satisfying Mercer's criteria).

3-2-4. Methods of making new kernel function [6], [7]

Simpler kernels can combined using certain operators. Then Kernel combination allows to design complex kernels on structures from simpler ones. Correctly using combination operators guarantees that complex kernels are p.d. The best of known operators are:

3-2-4-1. Kernel Sum

The sum of two kernels corresponds to the concatenation of their respective feature spaces:

$$\begin{aligned}&(k_1+k_2)(x,x')=\\ &=k_1(x,x')+k_2(x,x')\\ &=\Phi_1(x)^T\,\Phi_1(x')+\Phi_2(x)^T\,\Phi_2(x')\\ &=(\Phi_1(x)\ \Phi_2(x))\begin{pmatrix}\Phi_1(x')\\ \Phi_2(x')\end{pmatrix}\end{aligned} \quad (13)$$

The two kernels can be defined on different spaces (direct sum, e.g. string spectrum kernel plus string length).

3-2-4-2. Kernel Product

The product of two kernels corresponds to the Cartesian products of their features:

$$\begin{aligned}&(k_1\times k_2)(x,x')=k_1(x,x')k_2(x,x')=\\ &=\sum_{i=1}^{n}\Phi_{1i}(x)\Phi_{1i}(x')+\sum_{j=1}^{m}\Phi_{2j}(x)\Phi_{2j}(x')\\ &=\sum_{i=1}^{n}\sum_{j=1}^{m}(\Phi_{1i}(x)\Phi_{1i}(x'))(\Phi_{2j}(x)\Phi_{2j}(x'))\\ &=\sum_{k=1}^{nm}(\Phi_{12k}(x)\Phi_{12k}(x'))=\Phi_{12}(x)^T\,\Phi_{12}(x')\end{aligned} \quad (14)$$

Where $\Phi_{12}(x)=\Phi_1(x)\times\Phi_2(x)$ (15), is the Cartesian product. Note that the product can be between kernels in different spaces (tensor product).

3-2-4-3. Kernel Linear combination

A kernel can be rescaled by an arbitrary positive constant: $k_\beta(x,x')=\beta k(x,x')$ (16). We can e.g. define linear combinations of kernels (each rescaled by the desired weight):

$$k_{sum}(x,x')=\sum_{i=1}^{k}\beta_i k_i(x,x') \quad (17)$$

Note that:

- ▪ The weights of the linear combination can be learned simultaneously to the predictor weights (the alphas).
- ▪ This amounts at performing kernel learning.

3-2-4-4. Kernel normalization

Kernel values can often be influenced by the dimension of objects. E.g. a longer string has more substrings higher kernel value. This effect can be reduced normalizing the kernel.

- Cosine normalization:

Cosine normalization computes the cosine of the dot product in feature space:

$$\hat{k}(x,x')=\frac{k(x,x')}{\sqrt{k(x,x)k(x',x')}} \quad (18)$$

Or

$$\mathrm{k}(x_i,x_j)/\left[\mathrm{k}(x_i,x_i)\mathrm{k}(x_j,x_j)\right]^{\frac{1}{2}} \quad (19)$$

3-2-4-5. kernel convex combination

$$k(x_i,x_j)=\lambda_1 k_1(x_i,x_j)+\lambda_2 k_2(x_i,x_j) \quad (20)$$

3-3. introducing new kernels

In mathematics, an orthogonal polynomial sequence is a family of polynomials such that any two different polynomials in the sequence are orthogonal to each other under some inner product. Orthogonal polynomials are important in solving linear equations and the linear least squares fitting.

The most widely used orthogonal polynomials are the classical orthogonal polynomials, consisting of the Hermite polynomials, the Laguerre polynomials, the Chebyshev polynomials, and the Legendre polynomials. By using kernels based orthogonal polynomials, the number of support vectors has decreased, and the accuracy of classification has increased [9]. Now we introduce new kernels based orthogonal polynomials:

3-3-1. Hermite polynomials kernel [10]

The Hermite kernel for the given scalar valued inputs x and z is defined as:

$$k(x,z)=\sum_{i=0}^{n} H_i(x)H_i(z) \qquad (21).$$

Where $H_{n+1}(x)=xH_n(x)-n\,H_{n-1}(x)$ and $H_0(x)=1, H_1(x)=x$ (22).

For vector inputs x, z:

$$k(x,z)=\prod_{k=1}^{m}\sum_{i=0}^{n} H_i(x_k)H_i(z_k) \qquad (23).$$

3-3-2. Chebyshev polynomial kernel [11]

The Chebyshev kernel for the given scalar valued inputs x and z take this form:

$$k(x,z)=\frac{\sum_{i=0}^{n} T_i(x)T_i(z)}{\sqrt{1-xz}} \qquad (24).$$

Where $T_{n+1}(x)=2xT_n(x)-T_{n-1}(x)$ and $T_0(x)=1, T_1(x)=x$ (25).

As the Chebyshev polynomials are orthogonal only within the region [-1, 1] the input data needs to be normalized within this region according to the following formula:

$$x^{new}=\frac{2(x^{old}-\min)}{\max-\min}-1$$

(26). Where Min and Max are the minimum and maximum values of the entire data, respectively. For vector inputs x, z, we have:

$$k(x,z)=\prod_{j=1}^{d}\frac{\sum_{i=0}^{n} T_i(x_j)T_i(z_j)}{\sqrt{1-x_j z_j}}$$

(27). where d is the dimension of the training vectors x and z.

3-3-3. Legendre polynomial kernel [12]

For scalar inputs x and z, Legendre kernel is defined as follows:

$$k(x,z)=\sum_{i=0}^{n} P_i(x)P_i(z)=<\Phi_n(x),\Phi_n(z)> \qquad (28).$$

Where $p_{n+1}(x)=\frac{2n+1}{n+1}x\,p_n(x)-\frac{n}{n+1}p_{n-1}(x) \quad n\geq 1$ and $p_0(x)=1,\ p_1(x)=x$ (29).

The same as Chebyshev kernel function for vector input, each feature of the input vector for Legendre kernel function lies in [-1,1] So we have to normalize the input data to [-1,1] via the formula:

$$x_i^{new}=\frac{2(x_i^{old}-\min_i)}{\max_i-\min_i}-1 \qquad (30).$$

Where x_i is the i-th feature of the vector x, $\max_i$ and $\min_i$ are the minimum and maximum values along the i-th dimensions of all the training and test data, respectively.

3-3-4. Laguerre polynomial Kernel [13]

By using generalized Laguerre polynomials, we define generalized n-th order Laguerre kernel as:

$$k(x,z)=\sum_{i=0}^{n} L_i(x)L_i^T(z) \qquad (31).$$

Where $L_{n+1}(x)+(x-2n-1)L_n(x)+n^2L_{n-1}(x)=0$ and $L_0(x)=1\,,\ L_1(x)=1-x$ (32).

x and z are m-dimensional vectors.

3-4. the choice of kernel function [14]

A good choice of Kernel function is very important for effective SVM based classification. An appropriate Kernel function provides learning capability to SVM.

The most serious issue in the sphere of SVM is kernel function selection. It means that, in order to solve an encountered problem, how can we decide which kernel function is more optimized for this particular problem, and which one should we select out of existing functions.

It is not quite clear which kernel function offers the best result of a series of data. Therefore the best kernel function should be selected. Generally, even if a theoretical method of kernel selection is duly developed, its validity cannot be trusted until tested on a large number of problems.

Thus kernel selection is manually performed and if some of the points were not able to be differentiated, it should be continually performed to the extent of accurate differentiation with other kernels.

However, several methods and principles have been introduced for this purpose, which is selecting optimized kernel function, but they are still incomplete and have to be modified; of these we can name:

- diffusion kernel.
- fisher kernel.

- string kernel.

There are also studies being performed for obtaining kernel matrix out of existing data. But in the majority of problems, usually the Polynomial kernel and the linear kernel are the first which are used.

3-5. other kernels

In addition to the kernels mentioned in sections 3.1 and 3.3, there are other, already-produced kernels which are proposed by different studies, but they are not used frequently nowadays. These kernel functions are mentioned in the table below [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]:

Table 1. List of other kernel functions

Kernel function	Formula
Exponential Kernel	$k(x,y) = exp\left(-\frac{\Vert x-y \Vert}{2\sigma^2}\right)$
Laplacian Kernel	$k(x,y) = exp\left(-\frac{\Vert x-y \Vert}{\sigma}\right)$
Anova Kernel	$k(x,y) = \sum_{k=1}^{n} exp(-\sigma(x^k - y^k)^2)^d$
Rational Quadratic Kernel	$k(x,y) = 1 - \frac{\Vert x-y \Vert^2}{\Vert x-y \Vert^2 + c}$
Multiquadric Kernel	$k(x,y) = \sqrt{\Vert x-y \Vert^2 + c}$
Inverse Multiquadric Kernel	$k(x,y) = \frac{1}{\sqrt{\Vert x-y \Vert^2 + c}}$
Circular Kernel	$k(x,y) = \frac{2}{\pi}\arccos\left(-\frac{\Vert x-y \Vert}{\sigma}\right) - \frac{2}{\pi}\left(\frac{\Vert x-y \Vert}{\sigma}\right)\sqrt{1-\left(\frac{\Vert x-y \Vert}{\sigma}\right)^2}$
Spherical Kernel	$k(x,y) = 1 - \frac{3}{2}\left(\frac{\Vert x-y \Vert}{\sigma}\right) + \frac{1}{2}\left(\frac{\Vert x-y \Vert}{\sigma}\right)^3$ $if\ \Vert x-y \Vert < \sigma\ ,\ zero\,othrewise$
Wave Kernel	$k(x,y) = \left(\frac{\theta}{\Vert x-y \Vert}\right)\sin\left(\frac{\Vert x-y \Vert}{\theta}\right)$
Power Kernel	$k(x,y) = -\Vert x-y \Vert^d$
Log Kernel	$k(x,y) = -\log(\Vert x-y \Vert^d + 1)$
Spline Kernel	$k(x,y) = \prod_{i=1}^{d} 1 + x_i y_i + x_i y_i \min(x_i, y_i) -$ $\frac{x_i + y_i}{2}\min(x_i, y_i)^2 + \frac{\min(x_i, y_i)^3}{3}$ $where\ \ x, y \in R^d$
B-Spline kernel	$k(x,y) = \prod_{p=1}^{d} B_{2n+1}(x_p - y_p)$ $B_n(x) = \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\left(x + \frac{n+1}{2} - k\right)_+^n$ $x_+^d = \begin{cases} x^d, if\ x > 0 \\ 0,\ ow \end{cases}$
Bessel Kernel	$k(x,y) = \frac{J_{v+1}(\sigma\Vert x-y \Vert)}{\Vert x-y \Vert^{-n(v+1)}}$ $where\ J\ is\ the\ Bessel\ function\ of\ first\ kind$
Cauchy Kernel	$k(x,y) = \frac{1}{1 + \frac{\Vert x-y \Vert^2}{\sigma}}$
Chi-Square Kernel	$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i - y_i)}$
Histogram Intersection Kernel	$k(x,y) = \sum_{i=1}^{n} \min(x_i, y_i)$
Wavelet Kernel	$k(x,y) = \prod_{i=1}^{N} h\left(\frac{x_i - c}{a}\right) h\left(\frac{y_i - c}{a}\right)$ *Where* **a** *and* **c** *are the wavelet dilation and translation coefficients*

4. Conclusion

In this paper, the most widely used kernel functions, as well as new kernel functions derived from orthogonal polynomials have been introduced. Since using kernel functions can decrease support vectors and affect the speed, accuracy, and function of classification, depending on the proposed conditions for the production of the kernels, we can work on the combination of these functions in order to obtain more optimized functions with more accuracy of classification. Among them, we can work on combination of kernel functions based on the combination of orthogonal polynomials known as hybrids. But it should be remembered that still the most important issue in this sphere is how the kernel function is selected.

References:

1. http://www.google.com. 2017.
2. http://www.sciencepub.net. 2017.
3. http://www.yahoo.com. 2017.
4. https://www.wikipedia.org. 2017.

5/25/2017