CHAPTER - V

Support Vector

Machine Approach
CHAPTER 5

SUPPORT VECTOR MACHINE APPROACH

5.1 Support Vector Machines

Support vector machine is the machine learning classifier widely used in biometrics, it is a supervised learning technique that highly supported for classification easily and accurately from the given unique patterned image templates [100]. It has been extended to a number of tasks such as regression, principal component analysis and so on [101], [102]. Selecting a specific kernel and parameters are usually done in a try-and-see manner [103]. The nut shell of the support vector machine is the inner product kernel between support vector and a vector drawn from the input data space [104].

The support vector consists of small sub set of data points extracted through the learning algorithm from the training sample itself [105]. This method of constructing support vector machine work is based on its kernel. The three types of kernels are used in our research work are linear, polynomial and quadratic kernels. In this three types of methods are used Sequential Minimal Optimization (SMO), Quadratic Programming (QP) and Least Square (LS).
The separating hyperplane gives decision surface for acceptance or rejection of given iris image of authentic and imposter. Support vector machine is the best method for separating complex and nonlinear separable patterns [106]. It is a superior technique for image classification used in our research work. The support vector machine is used to classify the required feature inputs derived by multi block local binary pattern. A supervised machine learning technique supports classification easily and accurately[107]. The support vector machine model which is feed forward neural networks and uses kernel is shown in fig. 5.1.

FIGURE: 5.1 Support Vector Machine Architecture
Input vector is named as \( x \) has 1 to \( m_0 \) values. Single output \( y \) and nut shell is the kernel. The hidden layer forms kernel operation. The kernel function has been specified as \( k(x, y) \) [105]. Synaptic weights multiplied with the output of kernels have been marked as \( u \) vector. The outputs of the kernel have been summed along with bias to make decision as output. A kernel is a function used to identify similarity among the input image vectors, and it is used to update learning weights [108]. For the image classification kernel requires binary vector which possess the feature of image and the label.

The output of the kernel classifier is to reduce the complexity and size of feature vector. In short kernels are shortcuts used to classify images of high dimension space. Mathematically kernel is represented as \( k(x, y) \) which uses inner product of feature vectors \( x \) and \( y \), which simplifies nonlinear boundary mapping into linear learning model. The three types of kernels used in our research work are linear, polynomial and quadratic.
5.2 Kernels in Support Vector Machine

In the kernels of support vector machine dot-products of non-linear features, or similarity measurements are called as kernel functions. Kernel methods map the data into higher dimensional spaces [109]. There are also no constraints on the form of this mapping. This mapping function, however, hardly needs to be computed because of a tool called the Kernel trick. It analyze the relationship between input data and the corresponding output of a function. It encapsulates the properties of functions in a computationally efficient way [110].

5.2.1. Linear kernel

The linear kernel function $k(x, y)$ is defined in the equation 5.1. It is the equation of straight line used to separate the hyper plane for classification.

$$k(x, y) = x^T y + c$$

(5.1)

Linear kernel model is suitable in the place where data can be separate perfectly and it is optimized as minimization problem as follows,

Minimize $\|x\|^2$  

(5.2)
SubJECTED to,

\[ x \cdot y_i + c \geq 1, \quad \text{if } y_i = 1 \]  
(5.3)

\[ x \cdot y_i + c \leq -1, \quad \text{if } y_i = -1 \]  
(5.4)

The kernel function is the straight line function which separates the regions using the simple straight line as shown in the fig.5.2. The information are scattered from negative to positive regions and need to separate two different colours. The two image information given in blue and red colour is separable using simple straight line.

**FIGURE: 5.2 Linear Kernel Function**
The green line separates two regions and gives decision condition such that the region below the line is red region and above the region is blue region. In case of iris image classification it have been considered that the accept region and reject region as shown in the fig.5.3. It is simple to start and learn but the real world problems are not linear separable. The information are scattered from negative to positive region and quit difficult to draw such linear line. Hence nonlinear feature vectors exist in iris region has been required nonlinear separable polynomial kernel classifier.

**FIGURE: 5.3 Linear kernel based classification**
5.2.2 Polynomial Kernel

In polynomial kernel function the region has been separated through the curve drawn using the polynomial. It has been commonly used nonlinear model of support vector machine. It represents the similarity of training samples or vectors. This polynomial kernel is similar to polynomial regression but without the combinatorial blowup in the number of parameters to be learned and the kernel function is given in equation 5.5.

\[ k(x, y) = (\alpha^T y + c)^d \] (5.5)

Input vectors \(x\) and \(y\) are from the input space. When the value of \(d\) is greater than zero, the polynomial contains higher order and low order terms. As kernel, \(k\) corresponds to an inner product in a feature space based on some mapping \(\varphi\), \(K(x, y) = (\varphi(x), \varphi(y))\), the nature of \(\varphi\) as shown in the fig.5.4.
The equation 5.6 has been derived by assigning the value $d=2$, using multinomial theorem and regrouped it.

$$K(x, y) = \left( \sum_{i=1}^{n} x_i y_i + c \right)^2$$
$$= \sum_{i=1}^{n} x_i^2 y_i^2 + 2 \sum_{i=2}^{n} \sum_{j=1}^{i-1} (\sqrt{2} x_i x_j)(\sqrt{2} y_i y_j) + \sum_{i=1}^{n} (\sqrt{2} c x_i)(\sqrt{2} c y_i) + c^2$$

(5.6)
From this it follows that the feature map has given as in the equation 5.7.

\[
\varphi(x) = \left\{ x_n^2, \ldots, x_1^2, \sqrt{2}x_n^2x_{n-1}, \ldots, \sqrt{2}x_nx_{n-1}x_1, \ldots, \sqrt{2}x_nx_{n-2}x_1, \ldots, \sqrt{2}x_2x_1, \sqrt{2}x_n, \ldots, \sqrt{2}x_1, c \right\}
\]

(5.7)

The polynomial kernel for input space and feature space are shown in left and right respectively in fig. 5.4. In input space the boundary of separation is shown as ellipse and in the feature space it is converted into hyper plan using the polynomial kernel. This hyper plane separation has required for the decision making in classification. In addition, the iris unique features from 2400 cells have required more efficient nonlinear classifier. In this method the false accept and false reject rate are high. So, the quadratic kernel has preferred to achieve least false accept and false reject rate.

5.2.3 Quadratic Kernel

The quadratic kernel function is given in the equation 5.8.

\[
k(x, y) = \sqrt{\|x - y\|^2 + c^2}
\]

(5.8)

The power of the polynomial used in this case is equal to two. It is very much suitable to construct non trivial function. If the feature points are nonlinear then this quadratic kernel has applied to
separate the feature points. Points of the class \( y=1 \) (circles below) are placed in an inner region surrounded from all sides through points of class \( y=-1 \), again depicted as triangles. There is no single straight (linear) line that can separate both regions. However, it is still possible to find such a separator using transforming the points \( x = \{x_1, x_2\} \) from feature space to a quadratic kernel space with points given in the corresponding square coordinates \( \{x_1^2, x_2^2\} \). The technique of transforming from feature space into a quadratic kernel space allows for a linear separation can be formalized in terms of kernels. Assuming \( \Phi() \) is a vector coordinates transformation function.

A squared coordinate space would be \( \{\Phi(x)_1, \Phi(x)_2\} = \{x_1^2, x_2^2\} \). The support vector machine separation task has now acting on in the transformed space to find the support vectors that generate the equation 5.9.

\[
h \Phi(x) + b = \pm 1
\]  
\[5.9\]

The hypothesis vector \( h = \sum c_i \Phi(x_i) \) provides the sum over support vector points \( x_i \). Putting both expressions together results equation 5.10 with the scalar kernel function \( K(x_i, x) = \Phi(x_i)\Phi(x) \).

\[
\sum c_i K(x_i, x) + b = \pm 1
\]  
\[5.10\]
The kernel has composed out of the scalar product between a support vector $x_i$ and another feature vector point $x$ in the transformed space. In practice the support vector machine algorithm can be fully expressed in terms of kernels without having to actually specify the feature space transformation. Implementing a two dimensional quadratic kernel function allows the algorithm to find support vectors and correctly separate the regions. The nonlinear regions are linearly separated after transforming to the squared kernel space is shown in fig.5.5 and fig.5.6.

**FIGURE: 5.5 Feature Space**  
**FIGURE : 5.6 Quadratic Kernel Space**
5.3 Support Vector Machine Methods

The methods of support vector machine kernels have been applied in the given data set for optimization. Prominent methods have been implemented such as Sequential minimal optimization (SMO), Quadratic programming (QP) and Least square (LS).

5.3.1. Sequential Minimal Optimization (SMO)

Sequential minimal optimization is the optimization technique based on quadratic programming. It is widely used to train support vector machine. This method is very simple, efficient to implement in support vector machine. Sequential minimal optimization maximization problem for the binary classification with the data set \((x, y)\) consists of, \(x_i\) is an input vector and \(y_i\) (either \(-1\) or \(+1\)) is a binary label corresponding to it. The problem definition is given in the equations 5.11 to 5.13.

\[
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j k(x_i, x_j) \alpha_i \alpha_j \tag{5.11}
\]
Subjected to

$$0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \ldots, n$$ \hfill (5.12)

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$ \hfill (5.13)

where C is an support vector machine hyperplane and $k(x_i, x_j)$ is the kernel function and the variable $\alpha_i$ is Lagrange multiplier. It requires iterative process to solve and need to break into smallest sub problem which are simple to solve analytically. The equality constraint given in the equation 5.14 has further been simplified using another two multipliers $\alpha_1$ and $\alpha_2$ and obtained the equation 5.15.

$$0 \leq \alpha_1, \quad \alpha_2 \leq C$$ \hfill (5.14)

$$y_1 \alpha_1 + y_2 \alpha_2 = k$$ \hfill (5.15)

Where, k is the negative of the sum over the rest of terms in the equality constraint. This reduced problem can be solved analytically to find a minimum of a one-dimensional quadratic function.
5.3.2 Quadratic Programming

The optimization technique to find optimal classification, in support vector machine the quadratic programming problem is stated in the equations 5.16 to 5.18. Equation 5.16 is the objective of optimization problem which is subjected to inequality constraint and equality constraint given in the equation 5.17 and 5.18 respectively.

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{j} \alpha_i - \frac{1}{2} \sum_{i=1}^{j} \sum_{j=1}^{i} y_i y_j k(x_i, x_j) \alpha_i \alpha_j$$  (5.16)

$$0 \leq \alpha_i \leq C$$  (5.17)

$$\sum_{i=1}^{j} y_i \alpha_i = 0$$  (5.18)

The objective of our problem has been optimized if the Karush-Kuhn Tucker (KKT) conditions are fulfilled and $Q_{ij} = y_i y_j k(x_i, x_j)$ is positive semi definite. Such a point may be a non-unique and non-isolated optimum. For solving quadratic programming and Karush Kuhn Tucker conditions are given below in the equations 5.19 to 5.21 for all i.

$$\alpha_i = 0 \Rightarrow y_i f(x_i) \geq 1$$  (5.19)
\[ 0 < \alpha_i < C \Rightarrow y_i f(x_i) = 1 \]  
\[ \alpha_i = C \Rightarrow y_i f(x_i) \leq 1 \]

This formation of quadratic programming for the support vector machine is large in size and if the training examples are more than 4000 requires more space of greater than 128 Megabytes memory. To reduce this size chunking has been used. Chunking is the process of removing the rows and columns of the matrix which corresponds to zero Lagrange multipliers, in which large quadratic programming problem has been split into number of small quadratic programming problem. It reduces the matrix size and training. The size of quadratic sub problems are maintained constant and output of sub problem is added to the previous sub problem till the last one violates the and Karush Kuhn Tucker conditions. Very large quadratic problem has been solved using data structure with Hessian matrix. In this method the inner product or dot products between vectors have been performed on Hessian matrix instead matrix multiplication.

5.3.3 Least Square Method

Least square support vector machine are supervised learning machine used for pattern recognize and classification. It is kernel
based learning method solves set of linear equation to find the solution. The least square support vector machine has been derived from the simple support vector machine as follows.

The input training data set \( \{x_i, y_i\} \) and the input data belongs to \( x_i \in \mathbb{R}^n \) and corresponding binary class labels \( y_i \in \{-1, +1\} \). The support vector machine has satisfied the following conditions given in the equations 5.22 and 5.23

\[
\omega^T \phi(x_i) + b \geq 1, \quad \text{if } y_i = +1 \tag{5.22}
\]

\[
\omega^T \phi(x_i) + b \leq -1, \quad \text{if } y_i = -1 \tag{5.23}
\]

These two equations have been combined and written as in the equation (5.24)

\[
y_i [\omega^T \phi(x_i) + b] \geq 1, \quad i = 1, \ldots, N \tag{5.24}
\]

where, \( \phi(x) \) is the nonlinear map from original space to the high dimensional space. In case of un-separable hyper plane new slack variable \( \xi_i \) has been introduced for the separation. This slack variable converts the support vector machine equation as given in the equation 5.25 and 5.26.

\[
y_i [\omega^T \phi(x_i) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, N \tag{5.25}
\]
\( \xi_i \geq 0 \quad i = 1, \ldots, N \) \hspace{1cm} (5.26)

According to the structural risk minimization principle, the risk bound has been minimized the minimization problem as given in the equations 5.27 to 5.29

\[
\min J_1(\omega, \xi) = \frac{1}{2} \omega^T \omega + c \sum_{i=1}^{N} \xi_i \quad (5.27)
\]

Subject to

\[
y_i[\omega^T \phi(x_i) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, N \quad (5.28)
\]

\[
\xi_i \geq 0 \quad i = 1, \ldots, N \quad (5.29)
\]

To solve this problem, Lagrangian function has been constructed as given in the equation 5.30

\[
L_1(\omega, b, \xi, \alpha, \beta) = \frac{1}{2} \omega^T \omega + c \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \alpha_i[y_i[\omega^T \phi(x_i) + b] - 1 + \xi_i] + \sum_{i=1}^{N} \beta_i \xi_i \quad (5.30)
\]

Where, \( \alpha_i \geq 0, \beta_i \geq 0 \) \( i = 1, \ldots, N \) are the Lagrangian multipliers. The optimal point has been in saddle point of the Lagrangian function and the equation has been derived as given in 5.31 to 5.33.

\[
\frac{\partial L_1}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \quad (5.31)
\]
\[
\frac{\partial L_1}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0
\]  
\[ (5.32) \]

\[
\frac{\partial L_1}{\partial \xi_i} = 0 \rightarrow 0 \leq \alpha_i \leq C, \ i = 1, \ldots, N
\]  
\[ (5.33) \]

The least squares based support vector machine classifier has been formulated based on minimization technique as given in 5.34 to 5.35.

\[
\text{min } J_2(\omega, b, e) = \frac{\mu}{2} \omega^T \omega + \frac{\varepsilon}{2} \sum_{i=1}^{N} e^2_i
\]  
\[ (5.34) \]

Subjected To

\[
y_i[\omega^T \phi(x_i) + b] = 1 - e_{i,j}, \quad i = 1, \ldots, N
\]  
\[ (5.35) \]

The equality constraint given in the equation 5.34 of least square support vector machine implicitly corresponds to a regression with target \( y_i = \pm 1 \) using \( y_i^2 = 1 \) then the second part of the objective function may expand as given in the equation 5.36

\[
\sum_{i=1}^{N} e^2_{i,j} = \sum_{i=1}^{N} (y_i e_{i,j})^2 = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - (\omega^T \phi(x_i) + b))^2
\]  
\[ (5.36) \]

Where, \( e_i = y_i - (\omega^T \phi(x_i) + b) \). From this it is clear that least square data fitting is always present with error so that the same end result holds for the regression.
The least square support vector machine classifier formulation has been stated as in the equation 5.37.

\[ J_2(\omega, b, e) = \mu E_w + \zeta E_D \]  
(5.37)

Where both \( \mu \) and \( \zeta \) have be considered as hyper parameters to tune the amount of regularization versus the sum squared error. The solution depends only on the ratio \( \gamma = \frac{\zeta}{\mu} \) therefore the original formulation uses only \( \gamma \) as tuning parameter. In the ratio \( \mu \) and \( \zeta \) are used for Bayesian interpretation to least square support vector machine. The solution of least square support vector machine has been derived with Lagrangian function as given in the equation 5.38

\[
L_2(\omega, b, e, \alpha) = J_2(\omega, e) - \sum_{i=1}^{N} \alpha_i \{[\omega^T \phi(x_i) + b] + e_i - y_i \} \\
= \frac{1}{2} \omega^T \omega + \frac{\zeta}{2} \sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i \{[\omega^T \phi(x_i) + b] + e_i - y_i \}
\]  
(5.38)

Where \( \alpha_i \in \mathbb{R} \) are the Lagrange multipliers. The conditions for optimality are given in the equation 5.39 to 5.42.

\[
\frac{\partial L_2}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^{N} \alpha_i \phi(x_i) \]  
(5.39)

\[
\frac{\partial L_2}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0
\]  
(5.40)
\[
\frac{\partial L_2}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \quad i = 1, \ldots, N
\] (5.41)

\[
\frac{\partial L_i}{\partial \alpha_i} = 0 \rightarrow y_i = \omega^T \phi(x_i) + b + e_i, \quad i = 1, \ldots, N
\] (5.42)

5.4 Iris Recognition using Support Vector Machine

The support vector machine is a best binary classifier. The required input values are obtained in preprocessing stage. The accurate spectral features values are extracted in feature extraction stage. The input parameter are same as in artificial neural network. These input are passed as inputs to input layer of support vector machine. The inbuilt kernel function \( k(x,y) \) in each hidden layer has chosen the best features automatically from the given inputs and the bias function computes to provide the output. The classification process provides the output wherever ‘1’ occurred that is authorized and ‘0’ occurred unauthorized.

In our research work the three supervised support vector machine kernel linear, polynomial and quadratic kernels have been implemented. The least square method of quadratic kernel function has resulted in optimal performance with maximum true rejection and least false rejection rate at maximum speed for the given data set. It supports both authentication and recognition. The classification
can be done on the basis of both identity and without identity that means classification process of both one to one or among gallery. The classification speed is enhanced due to the minimal and unique features that are passed to the support vector machine classifier with 94.5% accuracy.