Optimal Kernel Selection in Kernel Fisher Discriminant Analysis: An Analysis

Somesh Kumar Gupta, Student Number: 20817245, University of Waterloo

1. Introduction

Kernel Fisher Discriminant Analysis (KFDA) uses a Kernel function that operates on a higher dimension without implicitly calculating the coordinates of each data point. This method is popular mainly because it reduces the cost of computation by a significant amount. KFDA finds the direction in feature space as defined implicitly by the Kernel. The performance of the KFDA varies with the choice of the kernel. The popular choices of Kernels include the Gaussian or polynomial family of Kernels. To enhance the performance of the Kernel, the Kernel parameters are tuned with cross-validation or generalized-cross-validation method [1]. However, the choice of an optimal Kernel requires a careful analysis keeping in mind the maximum achievable Fisher Discriminant Ratio (FDR). This choice of Optimal kernel selection considering the FDR is called the Optimal Kernel Selection problem in Kernel Fisher Discriminant Analysis.

There have been many studies conducted for the Optimal Kernel selection problem. Fung et al.[2] formulated an Optimal Kernel Selection problem which used the quadratic programming formulation of Fisher Linear Discriminant Analysis from the study of Mika et al.[3]. As mentioned in [4] that the problem formulation by Fung et al.[2] is not jointly convex in the variables. The method of Fung et al. [2] is developed on an iterative method that alternates between optimizing the Weight vector and the Gram matrix.

The study by Micchelli et al.[5] has shown that the problem of Optimal Kernel selection is a Convex optimization problem and in the study conducted by Boyd et al.[4] it has independently formulated the Optimal Kernel problem as a Convex optimization problem.

1.1. Notations and Definitions

The input is defined by χ which is a subset of $\mathbb{R}^n$ and the output or the class label is defined as $Y = \{-1, 1\}$. An input-output pair (x, y) is called an example set where $x \in \chi$ and $y \in Y$. An example is positive if the class label associated with it is $+1$ and negative if it is -1.

Now consider a symmetric function $K : \chi \times Y \to \mathbb{R}$ such that it satisfies positive semi-definite property. Then the function K is called a Kernel function, considering any $x_1, x_2, ..., x_m \in \chi$, the Gram matrix $G \in \mathbb{R}^{m \times m}$ is defined as

$$G_{ij} = K(x_i, x_j) \tag{1}$$

which is also a positive semi-definite matrix. In the study of [6] it is evident that any Kernel function K maps a set of input χ to a higher dimensional Hilbert space $\mathscr{H}$ by using the inner product $\langle ., . \rangle_{\mathscr{H}}$. The mapping function $\phi : \chi \to \mathscr{H}$ is a function which maps the lower dimensional input values to the Hilbert space $\mathscr{H}$. As from the study of [4] the function ϕ is written as:

$$K(x, z) = \big\langle \phi(x), \phi(z) \big\rangle, \forall\, x, z \in \chi \tag{2}$$

Product $\langle \phi(x), \phi(z) \rangle$ can also be written as $\phi(x)^T \phi(z)$. As indicated in [4] this space is called the

feature space and the mapping is called the feature mapping. The feature space and feature mapping depend on the Kernel K and can be denoted as ϕ_K and $\mathscr{H}_{\mathscr{K}}$ respectively.

The Kernel function defined in Boyd et al.[4] is the Convex combination of 10 different Gaussian Kernels and the formulation is as shown.

$$K(x,z) = \sum_{i=1}^{10} \theta_i e^{-\|(x-z)\|^2/\sigma_i^2} \tag{3}$$

where θ_i is the weight of the Kernels to be determined using the Numerical Optimization method. As given in [4] that the values of σ_i were chosen uniformly over the interval $[10^{-1}, 10^2]$ on a logarithmic scale. In addition, the regularization parameter chosen in KFDA is fixed to 10^{-8}.

1.2. Optimal Kernel Selection using Convex Optimization

The Optimal Kernel Selection problem can be formulated as a Convex optimization problem as discussed in [4]. The problem formulated by Boyd et al.[4] is as follows:

$$minimize \quad f_\lambda^*\Big(\sum_{i=1}^{p} \theta_i G_i\Big) \quad subject-to \quad \theta \succcurlyeq 0, \quad \mathbf{1}^T\theta = 1. \tag{4}$$

where $G_i = G_i^T \in \mathbb{R}^{m\times m}$ is the Gram matrix computed with the kernel K_i. The matrix G is a positive semidefinite matrix as it is a convex combination of the positive semidefinite matrices $G_1, ..., G_p$. The problem in equation 3 minimizes the convex function over p non-negative variables, with an equality constraint. The drawback of equation.4 is that it has higher computation costs mainly due to the calculation of the matrix G. The total cost for the equation grows like $O(m^3)$, with the additional hidden larger cost of constant in $O(.)$ notation. To reduce the computation cost of the convex optimization problem Boyd et al.[4] reformulated the equation using Schur complement technique. The equation is given by:

$$minimize \quad \Big(1/\lambda\Big)\Big(t - \sum_{i=1}^{p} \theta_i a^T G_i a\Big) \quad subject-to \quad H(t,\theta) \succcurlyeq 0, \quad \theta \succcurlyeq 0, \quad \mathbf{1}^T\theta = 1. \tag{5}$$

where the variable $t \in \mathbb{R}$ and $\theta \in \mathbb{R}^m$, and $H(t,\theta) \in \mathbb{R}^{n+1\times n+1}$ is defined as

$$H(t,\theta) = \begin{bmatrix} \lambda I + \sum_{i=1}^{p} \theta_i J G_i J & \sum_{i=1}^{p} \theta_i J G_i a \\ \sum_{i=1}^{p} \theta_i a^T G_i J & t \end{bmatrix} \tag{6}$$

Where

$$\mathrm{a} = \mathrm{a}_+ - a_-,$$

$$\mathrm{a}_+ = \begin{bmatrix} (1/m_+)\mathbf{1}_{m_+} \\ 0 \end{bmatrix},$$

$$\mathrm{a}_- = \begin{bmatrix} 0 \\ (1/m_-)\mathbf{1}_{m_-} \end{bmatrix},$$

$$\mathrm{J} = \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix},$$

$$\mathrm{J}_+ = \frac{1}{\sqrt{m_+}}\Big(I - \frac{1}{m_+}\mathbf{1}_{m_+}\mathbf{1}_{m_+}^T\Big),$$

$$\mathrm{J}_- = \frac{1}{\sqrt{m_-}}\Big(I - \frac{1}{m_-}\mathbf{1}_{m_-}\mathbf{1}_{m_-}^T\Big),$$

The problem reformulated by Boyd et al.[4] in equation.5 is a semidefinite program (SDP). The SDP problem can be solved by the interior-point methods with the same complexity as that of equation.4. Therefore the advantage of the SDP formulation in equation.5 is that we can find the optimal kernel using the Standard SDP solvers.

2. Literature Review

The process of Optimal Kernel Selection in Kernel-based classification is called Kernel Learning. The term is given by different researchers [7], [8], [9], [10], [2], [11], [12], [13], [14]. Kernel-Based learning methods were studied in greater detail by [11], [6], these algorithms work by searching a linear relationship among the input samples after transforming the data into Hilbert space which is represented as $\mathscr{H}$. As mentioned in [10] this transformation of the data into Hilbert space is also called embedding the data into the higher dimensional space and this embedding of the dataset into a Higher dimension or Hilbert space is done implicitly by the Kernel function. To reduce computation cost the Kernel function computes the inner product between each pair of points rather than computing their coordinates explicitly as the inner product in the embedding space can often be computed more easily than the computation of coordinates of these points themselves.

Let us take an input set denoted by χ and let the embedding space be denoted by $\mathscr{F}$ then Lanckriet et al.[10] maps $\phi : \chi \rightarrow \mathscr{F}$. For any two points $x_i \in \chi$ and $x_j \in \chi$, the function ϕ returns the inner product between their images in the embedding space $\mathscr{F}$ and this function ϕ is known as the Kernel function. Here ϕ is just a mapping function that maps χ to a feature space $\mathscr{F}$. A Kernel matrix is a square matrix of dimension $n \times n$ and is denoted as $K \in \mathbb{R}^{n \times n}$, where n is the total number of inputs. The Kernel matrix is defined as $K_{ij} = k(x_i, x_j)$ for some $x_1, ...x_n \in \chi$ and the Kernel function k. The Kernel matrix defined above is also known as the Gram matrix. As mentioned in [10], the Gram matrix is symmetric, positive semidefinite matrix as it specifies the inner product between all the pairs of input point given by $\{x_i\}_{i=1}^{n}$. In addition to the inner product it also completely determines the relative positions of all the points in the embedding space $\mathscr{F}$.

The study conducted by Lanckriet et al. in [10] set the theoretical foundation for Kernel learning using transduction. Transduction in [10] is defined as the problem of completing the labelling of a partially labelled dataset. Here there is no learning function instead it only learns a set of labels. Lanckriet et al address the problem of finding the unknown function of a dataset for which the labels are known but the function for each of the sample features is not known. This problem of unknown function is addressed by Lanckriet et al. in [10] by learning the Kernel matrix corresponding to the entire dataset. This matrix optimizes the cost function by depending on the available labels. In other words, the author uses the available labels to learn a good embedding and this is then applied to both the labelled and unlabeled data. The Kernel matrix is then used in combination with already existing algorithms that use kernels. In the new transduction method proposed by Lanckriet et al. in [10], it scales the SVMs polynomially with the number of test points. This method proposed in [10] offers a solution to optimize the 2-norm soft margin parameter for the SVM learning algorithms which was an important open problem.

All the implementation done in [10] is done using Semidefinite programming (SDP), SDP is a branch of Convex Optimization that majorly deals with finding the optimal solution of the Convex function over the Convex cone of positive semidefinite matrices. Besides, many natural cost functions which are motivated by error bounds are indeed Convex in the Kernel matrix. Lanckriet et al. present a solution to the problem of combining the data from multiple sources by assuming that each of the sources is associated with a Kernel function in which each training set produces a set of Kernel matrices. The tool that is developed in [10] makes it possible to optimize the coefficients in a linear combination of kernel matrices. The coefficients of the kernel matrices are then used to form the linear combination of the Kernel function in the classifier. The main advantage of this approach is that it allows the heterogeneous combinations of data sources which further aids in the reduction of heterogeneous data types to a common framework of Kernel matrices.

Lanckriet et al. [10] uses four different datasets in the evaluation of its new method. In its new

method of learning a Kernel matrix from the data Semidefinite programming (SDP) solutions are used. It takes into account the theorem that every symmetric, positive semidefinite matrix can be considered as a Kernel matrix corresponding to a certain embedding of a finite set of data. Besides, it also considers the fact that the Semidefinite programming method uses optimization of Convex cost function over the Convex cone of positive semidefinite matrices. The main focus of [10] is the transductive setting in which the labelled data is used to learn an embedding and as a result, this learned embedding is then applied to the unlabeled part of the dataset. From the developed generalization bound in [10] for transduction, it is shown that imposing Convex constraints effectively controls the capacity of the search space of possible Kernels and produces an efficient learning algorithm that can be implemented by Semidefinite programming. Furthermore as mentioned in [10] this approach gives a solution to a convex method to learn the 2-norm soft parameter in Support Vector Machines (SVMs) thus this solution leads to solving an important open problem in SVMs. The empirical results reported by Lanckriet et al. in [10] on the standard datasets show that the new approach developed by Lanckriet et al. provides a principle way of combining multiple Kernels to produce a classifier that is equivalent to the best performing individual classifier and can also perform better than any individual Kernel. As mentioned above that for obtaining better results Kernel hyperparameters is tuned with cross-validation however the approach used in [10] doesn't require any hyperparameter tuning with cross-validation. This result reported by Lanckriet et al. is very important and lays the basis for Kernel learning.

Micchelli et al.[5] established the general result on the convexity of the Kernel learning. The author consider the general Optimal Kernel selection problem of the form:

$$minimize \quad inf_{w\in\mathscr{H}} \sum_{i=1}^{m} \psi\Big(y_i, w^T\phi_K(x_i)\Big) + \lambda\,\|(w)\|^2 \quad subject-to \quad K\in\mathscr{K} \tag{7}$$

Here the Kernel function is $K:\chi\times\chi\rightarrow\mathbb{R}$. The positive regularization parameter is given by λ and its value is fixed to 10^{-8} as given in [4]. The training set for the problem is given by $\{(x_i,y_i)\}_i^m$, the set of Kernel functions is given by $\mathscr{K}$. The function $\psi:\mathbb{R}^2\rightarrow\mathbb{R}_+$ is the loss function or the hinge loss. As mentioned in [4] the Kernel problems that arise in many kernel-based problems including 1-norm soft margin and 2-norm soft margin SVMs have this form.

The main aim of the study conducted by Micchelli et al. in [5] is to broaden the theoretical understanding of how the study of Optimal Kernels is effected by the minimization of the regularization function. The main analysis in [5] is derived from the work of Lanckriet et al.[9], [10]. In the analysis by Micchelli et al.[5] it is established that the regularization function is a Convex function with respect to the Kernel function. In addition, it is also established that any Optimizing Kernel can be formulated or expressed as the Convex combination of at most $m+2$ basic Kernels. The paper has also described in detail the characterization of the resulting minimax problem of Square loss regularization function. Micchelli et al.[5] has shown that if the loss function ψ is convex, then the above problem is a Convex Optimization problem. In the study conducted by Micchelli et al. in [5] it has only marginally discussed about the algorithms for the search of Optimal Kernels and its implementation. However the theorems presented in the same study as drawn out by Micchelli et al. the formulation in developing a practical algorithm for learning an Optimal Kernel with a mixture of Gaussian or polynomial Kernel is possible. Micchelli et al.[5] also throws some light into a previously unexplored direction of deriving the error bounds which is discussed in greater detail in a different study conducted by Micchelli et al. in [15].

The problem formulated by [4] is not a direct consequence of the result in equation.7. The main reason for this being that the FDR does not satisfy the convexity condition. Therefore the formulation of the Optimal Kernel selection in [4] is independent of [5] which is given in the equations.4 and 5.

3. Proposed Solution

The problem formulated by Boyd et al.[4] as a Convex function is solved by the interior point method using SDP solver with a classifier that has formulations shown below.

$$h(x) = sgn(w^T \phi_K(x) + b) \tag{8}$$

Here we learn the classifier $h : \chi \rightarrow \{-1, +1\}$ from inputs to obtain the optimal weights from the training datasets whose decision boundary between the two classes of the target is affine in the feature space $\mathscr{H}_K$. The weight $w \in \mathscr{H}_K$ is a feature weight vector and the intercept b $\in \mathbb{R}$ is the bias term,

$$sgn(u) = \begin{cases} +1, & if \quad u > 0 \\ -1, & else \quad u < 0 \end{cases} \tag{9}$$

The classifier predicts the class labels once it obtain a scalar value from the equation.8 and then if the value is less than 0 the classifier function puts that sample into negative class and if the value is greater than 0 then the classifier puts the sample into the positive class. In [4] the author reformulated the representation of the optimal decision boundary using the kernel function. For a input data point $x \in \chi$ the inner product $\langle w^*, \phi_K(x) \rangle_{\mathscr{H}_K}$ in [4] is written as:

$$\langle w*, \phi_K(x) \rangle = \sum_{i=1}^{m} \alpha^* \phi_K^T(x_i) \phi_K(x) \tag{10}$$

and from equation.2 we can write equation.10 as

$$\langle w*, \phi_K(x) \rangle = \sum_{i=1}^{m} \alpha^* K(x_i, x) \tag{11}$$

This is known as kernel trick as this computes the inner product of the pair $(x_i, x), i = 1, ..., m$. In this paper we deduce the classifier function $h(x)$ from the classifier equation formulated by Body et al.[4] without the weight vector and the intercept b which is as shown

$$h(x) = \sum_{i=1}^{m} \alpha^* \sum_{i=1}^{p} \theta_i^* K(x_i, x) \tag{12}$$

where α^* is given as:

$$\alpha^* = \frac{1}{\lambda} \Big[I - J(\lambda I + J G_K J)^{-1} J G_K \Big] a \tag{13}$$

where G is the Gram matrix which is defined as

$$G_K = \begin{bmatrix} K(x_1, x_1) & ... & K(x_1, x_m) \\ . & . & . \\ . & . & . \\ . & . & . \\ K(x_m, x_1) & ... & K(x_m, x_m) \end{bmatrix} \tag{14}$$

As in the literature section it is shown how the Gram matrix is a Convex combination of Kernel function and from equation.13 we can further derive as mentioned in [4] as below

$$F_\lambda^*(K) = \frac{1}{\lambda} \Big[a^T G_K a - a^T G_K J (\lambda I + J G_K J)^{-1} J G_K a \Big] \tag{15}$$

From equation.15 it is clear that the right hand side of the equation is a function of Gram matrix. Let the set of Gram matrices be denoted by $\mathscr{G}$ so this can be represented in the form of a Kernel function as given below

$$\mathcal{G} = \left\{ G_K \quad | \quad K \in \mathcal{K} \right\} \tag{16}$$

The set $\mathcal{G}$ is Convex subset of $\mathbb{S}_+^m$ and here $\mathbb{S}_+^m$ is denoted as the set of all $m \times m$ symmetric positive semidefinite matrices. This Convexity follows directly form the Convexity of the set of Kernel function $\mathcal{K}$. In addition to as given in [4] any Gram matrix $G \in \mathcal{G}$ is positive semidefinite as the Kernel function $K \in \mathcal{K}$ satisfies the positive semidefinite property of as defined in [10] of Kernel function. Based on these definition the cost function for the Optimal Kernel selection for the equation.4 can further be simplified from equation.15 as shown below

$$f_\lambda^*(K) = \frac{1}{\lambda}\left[a^T GJ(\lambda I + JGJ)^{-1} JGa - a^T Ga \right] \tag{17}$$

This derivation of the cost function mainly reduces the cost of computation and it can also be expressed as a composite function $f(h(G), s(G))$.

Using the classifier function given in equation.12 we will classify the samples as per their class labels. In equation.12, there is no Optimal weight(w^*) and the optimal intercept (b^*). The information about the class separation is already included in the term α^* and this value is computed for each sample of the dataset. Besides, equation.12 also uses the Gaussian combination of 10 different Kernels each of which is weighted by the weight of the Kernel function. This deduction from the equation.11 can save both memory and computation costs when we compute the accuracy score for a relatively larger dimension dataset. In addition to the classifier, we also take a different approach to solve the Convex problem of Optimal Kernel selection on different datasets with different numerical methods. The numerical method used by Boyd et al.[4] is the interior point method using SDP solvers such as SeDuMi[16] or the SDPT3[17]. In this paper, we use two different numerical methods the first is the Nelder Mead Method [18] and the second is Conjugate Gradient Method [19] to analyze how the results vary with the datasets when we change the numerical method. For each of the Numerical Optimization method we will first calculate the Optimum weight of the Kernel function and then using the formula as given in equation.12 we will predict the class labels of the dataset for each sample. We use three of the datasets as given in [4] in order to evaluate our classifier function $h(x)$.

4. Experimental Results

The implementation of the Convex Optimization problem in [4] was computed using SDP [16] solver using the interior point method. The classifier used by Boyd et al. [4] is given in equation.8 it has an optimal weight (w^*) along with an optimal intercept term (b^*). After a comprehensive analysis of the classifier equation, we reformulate the classifier equation as mentioned in equation.12. In this section, an evaluation of the classifier function on different datasets using different numerical Optimization algorithm is discussed. For extensive analysis, the complete dataset was divided into Test set with the size of 30% and the remaining 70% into Validation set and then the accuracy score was calculated after computing the Optimal Kernel weight vector (θ_i) using each of the algorithm discussed below. To calculate the accuracy score for the Test set and Validation set, first the Optimal weight vector was calculated using the Nelder Mead method and then the Conjugate Gradient method was used for each of the datasets. In both, the algorithm an initial value of weight vector was passed to both the algorithm to compute the Optimal weight vector. After computing the Optimal weight vector the accuracy for the Test set and Validation set for each of the dataset using both Nelder Mead and Conjugate Gradient algorithm was computed.

4.1. Nelder Mead Method

Nelder Mead algorithm [18] is also known as the downhill simplex method that is commonly applied to minimize or maximize an objective function in a multidimensional space. The Nelder Mead algorithm is a direct search method that can be applied to nonlinear Optimization problems whose derivative

may not be known and the method can converge to non-stationary points. We used three different datasets of different dimensions and sizes in order to evaluate that the analyzed classifier function separates the samples according to their class labels (target). For analyzing that the classifier works on different datasets of different dimensions three separate datasets were used. PIMA[20] has a dimension of (768,8), Sonar[21] has a dimension of (208,60) and Ionosphere dataset's dimension is (351,34) [22]. The table below shows the accuracy score for the classifier after computing the Optimal kernel weights using the Nelder Mead algorithm. As we can infer from Table I that among all the dataset this algorithm gives the best result for Ionosphere dataset[22] with a mean of 63.85% whereas for the sonar dataset[21] the accuracy score is lowest among both the algorithms with a mean accuracy score of 54.61% for this algorithm. Therefore in this paper we have successfully implemented a different Classifier function that predicts datasets of different dimensions using two different Numerical Optimization methods.

TABLE I
ACCURACY SCORE USING NELDER MEAD ALGORITHM

Dataset	Test Set Accuracy(%)	Validation Set Accuracy(%)
PIMA	65.37	61.94
Sonar	52.08	57.14
Ionosphere	64.49	63.21

4.2. Conjugate Gradient Method

The Conjugate gradient method [19] is an algorithm that finds numerical solutions for a particular system of equations whose matrix is symmetric and positive definite. The Conjugate gradient method is often applied to problems by implementing it as an iterative algorithm. It is applied to systems that are sparse and to the problems that are too large to be Optimized using the direct method or direct implementation methods such as Cholesky decomposition. When we consider a problem with large inputs often the Optimization problem leads to a large sparse system. One of the main advantages of the Conjugate gradient method is that it can be used to solve unconstrained optimization problems. In this method three different datasets are used namely PIMA [20], Sonar [21] and Ionosphere [22]. The main reason for using different datasets is to analyze and see if the classifier is able to predict the classes of different and large dimension dataset. From Table II we can infer that the overall accuracy score is increased from that of the Nelder Mead algorithm. The accuracy score is best for the Ionosphere dataset [22] with a mean score of 69.3% for both the sets. Whereas the Sonar dataset [21] has the least accuracy score.

TABLE II
ACCURACY SCORE USING CONJUGATE GRADIENT ALGORITHM

Dataset	Test Set Accuracy(%)	Validation Set Accuracy(%)
PIMA	69.70	65.86
Sonar	57.96	63.42
Ionosphere	69.73	68.87

The results from Table I shows the accuracy score of Nelder mead algorithm [18] on three different datasets and the Table II shows the accuracy score of Conjugate Gradient algorithm [19] on the above-mentioned dataset. The analysis of the Test set and Validation set for each of the algorithms is discussed as per the dataset in the below subsections.

4.3. PIMA Dataset

4.3.1. Nelder Mead Algorithm

PIMA dataset [20] is a diabetes dataset with 8 different features for 768 patients. The data from Table I shows the accuracy score after finding the Optimal $\theta(\theta_i)$ vector using the Nelder Mead algorithm.

The Optimal theta vector was used to calculate the accuracy score for both the Test set as well as the Validation set. For the Test set, the accuracy is 65.37% whereas for the Validation set it is 61.94%. The mean accuracy for this dataset is 63.66%, which is slightly below the Ionosphere dataset. For the Test set, the accuracy score is highest among all the datasets for this algorithm.

4.3.2. Conjugate Gradient Algorithm

The accuracy score for this method is given in Table II and for this dataset, the accuracy score was obtained after computing the Optimum θ vector for both the Test set and Validation set. The accuracy for the Test set is 69.70% whereas for the Validation set it is 65.86%, which is lower to that of the Ionosphere dataset [22]. The score is better than that of the Nelder Mead algorithm by approximately 5% for the same dataset. The mean accuracy score for this method is 67.78% which is approximately 2% lower than that of the Ionosphere dataset.

4.4. Sonar Dataset

4.4.1. Nelder Mead Algorithm

Sonar Dataset [21] is a different dataset when compared to the other two datasets as this has more number of features, it has a dimension of (208,60). This dataset has 60 features with 208 different samples. The accuracy score for the Test set after computing the Optimal θ vector using the Nelder Mead method is close to 52% whereas for the Validation set the accuracy increases to 57.14%. This dataset has the least accuracy score from three datasets both in the Test set as well as the Validation set. The mean accuracy score given by the deduced classifier function is 54.61%. The accuracy score is approximately 11% lower in the Test set as compared to other datasets whereas the accuracy score in the Validation set is approximately 5% lower as compared to others.

4.4.2. Conjugate Gradient Algorithm

This method gives a slightly better accuracy score when compared to the Nelder Mead method, however, for this dataset the accuracy score is lowest among all the dataset. The accuracy score for Test set increases by approximately 6% to 58% as compared to Nelder Mead's 52% when Optimal θ vector is used after computing from this algorithm. Besides, the accuracy score for the Validation set also increases by 6.28% to 63.42% as compared to Nelder Mead's 57.14%. The mean accuracy for this algorithm increases by approximately 6% to 60.69%. However, as compared to other datasets the accuracy score is lower by 11% in the Test set and approximately 3% in the Validation set which is increased by 2% as compared to the Nelder Mead method.

4.5. Ionosphere Dataset

4.5.1. Nelder Mead Algorithm

Ionosphere Dataset [22] has a dimension of (351,34) where the 351 represents the total number of different data samples and 34 represent the total number of features that each sample contains. The accuracy score for the Test set is 64.49% wich slightly less than the PIMA dataset [20]. However, for Validation set the accuracy score is 63.21% which is the highest among all the dataset in the Validation set. The mean accuracy score for this dataset is 63.85% which highest among all the dataset classified by the classifier using the Optimal θ vector computed by this algorithm.

4.5.2. Conjugate Gradient Algorithm

The mean accuracy score of the Ionosphere dataset [22] is the highest that the classifier computes among all the datasets when the Optimal θ vector is calculated using this method. The mean accuracy for the dataset is 69.3%. The accuracy score of the Test set is higher than that of the Validation by approximately 1% to 69.73%. This score is also the highest among all the Test set in both the algorithm. The accuracy score of the Validation set is 68.87% which very close to that of the Test set. As compared to the Nelder Mead algorithm, the accuracy in this algorithm is increased by approximately 6%. In addition, the mean score for this dataset is the highest among all the datasets including both the algorithms.

5. Discussion and Conclusions

In the Numerical results presented by Boyd et al. [4] the mean Test set accuracy for the Ionosphere dataset [22] was the highest among all the datasets with the accuracy score of 94.1%. The second highest accuracy is for Sonar dataset [21] with a mean accuracy score of 84.4% approximately 10% lower than that of the Ionosphere dataset. The mean accuracy score for PIMA dataset [20] was the lowest amongst all the datasets with a score of 74.9%. These scores are calculated after finding the Optimal weight of the Kernel function using the SDP solver with the Interior point method. The classifier used for this accuracy score is given in equation.8 which is the original Classifier function. The accuracy score that is presented in Table I and Table II using the deduced classifier equation.12 is lower than the results in [4]. These accuracy scores were calculated using the Nelder Mead algorithm and the Conjugate Gradient algorithm.

The highest accuracy score is given when the Optimal weight of the Kernel function is computed using Conjugate Gradient method and the score for Ionosphere dataset [22] is 69.73% which is lower than that in [4] which verifies that the deduced classifier can predict on any dataset. However, the accuracy score for the datasets is decreased in each case from that in [4]. One of the main reasons for this could be that the deduced classifier function doesn't involve any intercept and the optimum weight vector in the formula, therefore, the accuracy score is lower for the datasets as mentioned in Table I and Table II. Besides, in the Numerical result of [4] the difference between the highest accuracy score of the Ionosphere dataset [22] and that of PIMA dataset [20] is approximately 20%. This difference is highest among all the datasets. However, in the case of the deduced classifier function, this is reduced to approximately 11% in the Test set and further reduced in Validation set to 5%. This suggests that the deduced classifier function performs better for datasets with a larger number of input samples. However, as [4] points out that the interior point method might not work as expected for datasets with a larger number of input samples which is not the case in this implementation of the project as this uses Conjugate Gradient method for computation of the Optimal weight of the Kernel function.

In [4] the difference in the accuracy score between the Ionosphere dataset [22] and PIMA dataset [20] is approximately 20% whereas the accuracy score from the deduced classifier function is very similar. The accuracy score between the two datasets is approximately the same and close to 70% in the test set and a mean of 68% for the Validation set. It is clear that for the Original classifier function is given in equation.8 training is required to effectively compute the Optimal weight vector for each dataset along with computation of Optimal intercept variable which in turn adds towards the cost of computation along with more requirements for memory. However, one of the main advantages of this deduced classifier function given in equation.12 is that it doesn't require any form of training as there is no Optimal weight vector along with the Optimal intercept variable, this intern reduces the cost of computation and memory as the overall computation requirement is reduced. However, the penalty for this is paid in terms of accuracy score as the accuracy from this classifier is less than that of [4].

The detailed analysis in the Numerical Results section shows that the Classifier equation.12 deduced from the original Classifier function successfully predicts the class labels (targets) for data of different dimensions. The derived Classifier function doesn't have any Optimal weight and the Optimal intercept variable. The results presented in the above section doesn't compute optimal weight (w^*) and the intercept variables to predict the class labels (target). Therefore this classifier requires less memory and the cost of computation is relatively lower. Besides, from the results of different Optimization Numerical methods as described in the Experimental Results section, it is evident that Conjugate Gradient Method outperforms the Nelder Meads method for this classifier for the datasets described in the Experimental Results section. The accuracy score in the dataset by the Conjugate Gradient method is approximately 5% higher than that of the Nelder Mead method for both the Test set and Validation set. However, the time complexity for computing the Optimal weight of Kernel function is lower for the Nelder Mead algorithm whereas for the Conjugate Gradient method the time taken to compute the Optimal weight of the Kernel function is slightly higher with an additional cost of computing the gradient for each dataset.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*, 02 2009.

[2] G. Fung, M. Dundar, J. Bi, and R. Rao, "A fast iterative algorithm for fisher discriminant using heterogeneous kernels," 01 2004.

[3] S. Mika, G. Rätsch, and K.-R. Müller, "A mathematical programming approach to the kernel fisher algorithm," vol. 13, 01 2000, pp. 591–597.

[4] S.-J. Kim, A. Magnani, and S. Boyd, "Optimal kernel selection in kernel fisher discriminant analysis," vol. 2006, 01 2006, pp. 465–472.

[5] C. Micchelli and M. Pontil, "Learning the kernel function via regularization." *Journal of Machine Learning Research*, vol. 6, pp. 1099–1125, 07 2005.

[6] J. Shawe-Taylor, *Kernel methods for pattern analysis*. Cambridge, UK ;: Cambridge University Press.

[7] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," 01 2004.

[8] K. Bennett, M. Momma, and M. Embrechts, "Mark: A boosting algorithm for heterogeneous kernel models," 08 2002.

[9] G. Lanckriet, T. Bie, N. Cristianini, M. Jordan, and W. Noble, "A statistical framework for genomic data fusion," *Bioinformatics (Oxford, England)*, vol. 20, pp. 2626–35, 12 2004.

[10] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the kernel matrix with semi-definite programming." vol. 5, 01 2002, pp. 323–330.

[11] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment." vol. 1, 01 2001, pp. 367–373.

[12] K. Crammer, J. Keshet, and Y. Singer, "Kernel design using boosting," 06 2003.

[13] C. S. Ong, A. Smola, and R. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 07 2005.

[14] H. Xiong, M. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *Neural Networks, IEEE Transactions on*, vol. 16, pp. 460 – 474, 04 2005.

[15] A. Argyriou, C. Micchelli, and M. Pontil, "Learning convex combinations of continuously parameterized basic kernels," *Learning Theory, Proceedings*, vol. 3559, pp. 338–352, 2005.

[16] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," 1998.

[17] R. H. Tütüncü, K.-C. Toh, and M. J. Todd, "Sdpt3 — a matlab software package for semidefinite-quadratic-linear programming, version 3.0," 2001.

[18] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, vol. 7, pp. 308–313, 01 1965.

[19] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal cf Research of the National Bureau of Standards Vol*, vol. 49, 12 1952.

[20] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus." Publ by IEEE, 1988, pp. 261–265.

[21] R. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Networks*, vol. 1, no. 1, pp. 75–89, 1988.

[22] V. Sigillito, S. Wing, L. Hutton, and K. Baker, "Classification of radar returns from the ionosphere using neural networks." *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989. [Online]. Available: http://search.proquest.com/docview/25142006/