INDIAN INSTITUTE OF TECHNOLOGY INDORE

UNDERGRADUATE THESIS

Localized Multiple Kernel Learning for Anomaly Detection

Author:
SUDHARSAN K
ID No. 140001014

Supervisors:
Dr. ARUNA TIWARI
Dr. KAPIL AHUJA

Thesis submitted in fulfillment of the requirements for the degree of Bachelor of Technology

in the

Department of Computer Science and Engineering

December 7, 2017

Declaration of Authorship

I, SUDHARSAN K declare that this thesis titled, "Localized Multiple Kernel Learning for Anomaly Detection" and the work presented in it is my own. I confirm that:

- This work was done wholly or mainly while in candidature for the BTP project at IIT Indore.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Certificate

This is to certify that the thesis entitled, "*Localized Multiple Kernel Learning for Anomaly Detection*" and submitted by Sudharsan K ID No 140001014 in partial fulfillment of the requirements of CS 493 B.Tech Project embodies the work done by him under my supervision.

Supervisor

Dr.ARUNA TIWARI
Associate Professor,
Indian Institute of Technology Indore
Date:

Supervisor

Dr.KAPIL AHUJA
Associate Professor,
Indian Institute of Technology Indore
Date:

"I love deadlines. I like the whooshing sound they make as they fly by. "

Douglas Adams

INDIAN INSTITUTE OF TECHNOLOGY INDORE

Abstract

Department of Computer Science and Engineering

Bachelor of Technology

Localized Multiple Kernel Learning for Anomaly Detection

Multi-kernel learning has been well explored in the recent past and has exhibited promising outcomes for multi-class classification and regression tasks. In this project, I present a multiple Kernel learning approach for the One-Class Classification (OCC) task and employ it for anomaly detection. Recently, the basic multi-kernel approach has been proposed to solve the OCC problem, which is simply a convex combination of different Kernels with equal weights. This paper proposes a localized multiple Kernel learning approach for anomaly detection (LMKAD) using OCC, where the weight for each Kernel is assigned locally. Proposed LMKAD approach adapts the weight for each Kernel using a gating model. The parameters of the gating model and one class classifier are optimized simultaneously through a two-step optimization process. We present the empirical results of performance of LMKAD on 20 benchmark datasets from various disciplines. This performance is evaluated against existing Multi Kernel Anomaly detection (MKAD) algorithm, and other existing one class classifiers to showcase the credibility of our approach. Our algorithm achieves significantly better Gmean scores while using a lesser number of support vectors compared to conventional OCSVM and MKAD. Friedman test is also performed to verify the statistical significance of the results claimed in this project.

Acknowledgements

I would like to thank my B.Tech Project supervisors **Dr. Aruna Tiwari** and **Dr. Kapil Ahuja** for their guidance and constant support in structuring the project and their valuable feedback throughout the course of this project. Their overseeing the project meant there was a lot that I learnt while working on it. I thank them for their time and efforts.

I am grateful to **Mr. Chandan Gautam** without whom this project would have been impossible. He provided valuable guidance with the Mathematics involved in the project and also taught me how to write a scientific paper.

Most importantly, I am thankful to **Ramesh Balaji**, my BTP partner who was the catalyst and the driving force of the project. From waking me up in the mornings to keeping me from losing focus he was there through it all. Also, I am thankful to my other friends who were a constant source of both motivation and light hearted humour.

I am really grateful to the Institute for the opportunity to be exposed to systemic research especially Dr. Tiwari's Lab for providing the necessary hardware utilities to complete the project. Lastly, I offer my sincere thanks to everyone who helped me complete this project, whose name I might I have forgotten to mention.

Contents

List of Tables

List of Abbreviations

OCC	**O**ne **C**lass **C**lassifier
SV	**S**upport **V**ector
SVM	**S**upport **V**ector **M**achine
SVDD	**S**upport **V**ector **D**ata **D**escription
MKL	**M**ultiple **K**ernel **L**earning
LMKL	**L**ocalised **M**ultiple **K**ernel **L**earning
MKAD	**M**ultiple **K**ernel **A**nomaly **D**etection
LMKAD	**L**ocalised **M**ultiple **K**ernel **A**nomaly **D**etection
OCSVM	**O**ne **C**lass **S**upport **V**ector **M**achine
UCI	**U**niversity **C**alifornia **Irvine**
PCA	**P**rinciple **C**omponent **A**nalysis
SOM	**S**elf **O**rganising **M**aps
KRR	**K**ernel **R**idge **R**egression
KNN	**K**-**N**earest **N**eighbour
AD	**A**nomaly **D**etection
SMC	**S**oft **M**argin **C**lassifier
Gmean	**G**eometric **M**ean
FRank	**F**riedman **R**ank
MGmean	**M**ean **G**eometric **M**ean
MSE	**M**ean **S**quare **E**rror

Dedicated to all the first graders who are getting introduced to math.

Chapter 1

Introduction

1.1 Background

The problem of Anomaly Detection is the problem of finding instances of the input data that do not conform to the general pattern or behavior exhibited by majority of the data points. This problem has been well explored and addressed in the past using One-Class Classification [1]. The One-Class Classification (OCC) problem is unlike the conventional binary and multi-class classification problems in that in OCC, one has data about only one of the many classes that could constitute the input space. Outliers on the other hand could belong to any class or even be isolated anomalies.

Imagine a factory type of setting; heavy machinery under constant surveillance of some advanced system. The task of the controlling system is to determine when something goes wrong; the products are below quality, the machine produces strange vibrations or something like a temperature that rises. It is relatively easy to gather training data of situations that are OK; it is just the normal production situation. But on the other side, collection example data of a faulty system state can be rather expensive, or just impossible. If a faulty system state could be simulated, there is no way to guarantee that all the faulty states are simulated and thus recognized in a traditional two-class problem.

To cope with this problem, one-class classification problems (and solutions) are introduced. By just providing the normal training data, an algorithm creates a (representational) model of this data. If newly encountered data is too different, according to some measurement, from this model, it is labeled as out-of-class.

Various models to handle OCC problems have been developed [1].Tax [2] has developed three models which are density based viz., (i) Gaussian model (ii) mixture of Gaussians and (iii) Parzen density estimator, three models are boundary based viz., (i) k-centers method (ii) KNNdd and (iii) svdd and three models are reconstruction based viz., (i) k-mean clustering (ii) self-organizing maps (iii) PCA and mixtures of PCA's. Scholkopf [3] and Tax and Duin [4] have developed methods based on Support Vector Machine proposed by Vapnik [5]. One-class k-nearest neighbor (KNN) based approach has been applied by Munroe and Madden [20] for vehicle model recognition from images.

Out of these, SVM based methods have gotten more attention from researchers due to their efficiency, kernel learning ability and generalization capability. In SVM based methods, the OCC problem is redefined as the task of devising a boundary around the given data of target class points, such that most of the target class points lie within the defined boundary, while at the same time minimizing the chance that a given input outlier is accepted into the target class. Two types of SVM based methods have been developed viz., Support Vector Data Description (SVDD) [2] and OCSVM [3]. Tax and Duin [2] developed SVDD by finding a hyper-sphere of minimum radius around the target class data such that it encloses almost all points in the target

class data set. Scholkopf et al. [3] extend the idea of SVM for binary classification [6] to the domain of anomaly detection by proposing one class SVM. They construct a hyper-plane such that it separates all the data points from the origin and the hyper-plane's distance from the origin is maximum.

The general idea of kernel based methods such as SVM is to project the input space to a higher dimension where they become linearly separable. Kernel based methods have received considerable attention over the last decade due to their success in classification problems. An advance in kernel based methods for classification problems is to use many different kernels or different parameterization of kernels instead of a single fixed kernel [7] to get better performance. Using multiple kernels gives two advantages. Firstly, this provides flexibility to select for an optimal kernel or parameterizations of kernels from a larger set of kernels, thus reducing bias due to kernel selection and at the same time allowing for a more automated approach. Secondly, multiple kernels are also reflective of the need to combine knowledge from different data sources (such as images and sound in video data). Thus they accommodate the different notions of similarity in the different features of the input space. A further advance in multiple kernel learning for binary classification [8] and image recognition problems [9] is to localize the kernel selection process. Gonen et al. [8] propose a multiple kernel approach using a gating model to select the appropriate kernel locally.

We choose Scholkopf's one-class SVM as the classifier for our developments because of its robust performance as reported by other researchers [10], guaranteed convergence and the flexibility of kernel methods in general [11, 12].

Chapter 2

Literature Survey

The following chapter discusses literature pertaining to previously known methods of one class classification, focussing mainly on SVM based methods. It describes the conventional One class SVM method and Multiple Kernel Anomaly Detection method in detail.

2.1 One Class Classification

Conventional multi-class classification algorithms aim to classify an unknown object into one of several pre-defined categories. A problem arises when the unknown object does not belong to any of those categories. In one-class classification one of the classes (referred to as the positive class or target class) is well characterized by instances in the training data. For the other class (nontarget), it has either no instances at all, very few of them, or they do not form a statistically-representative sample of the negative concept.

2.1.1 One-Class SVM

One-Class SVM was proposed by Scholkopf et al. [3] for extending the utility offered by SVMs to One-Class classification. Given a set of training vectors $x_i \in \mathbb{R}^n, i = 1, \ldots, l$ without class labels, One-Class SVM constructs a hyperplane that basically separates all the target class data points from the origin and maximizes the distance of this hyperplane from the origin. This is done by solving the following optimization problem.

$$\begin{aligned} \min_{\omega, \xi, \rho} \quad & \frac{1}{2}\omega^T\omega - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & \omega^T\phi(x_i) \geq \rho - \xi_i \quad i = 0, \ldots, l, \\ & \xi_i \geq 0, \qquad\quad i = 0, \ldots, l \end{aligned} \tag{2.1}$$

The dual of which can be written as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q\alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \tfrac{1}{\nu l} \quad i = 0, \ldots, l, \\ & e^T\alpha = 1 \end{aligned} \tag{2.2}$$

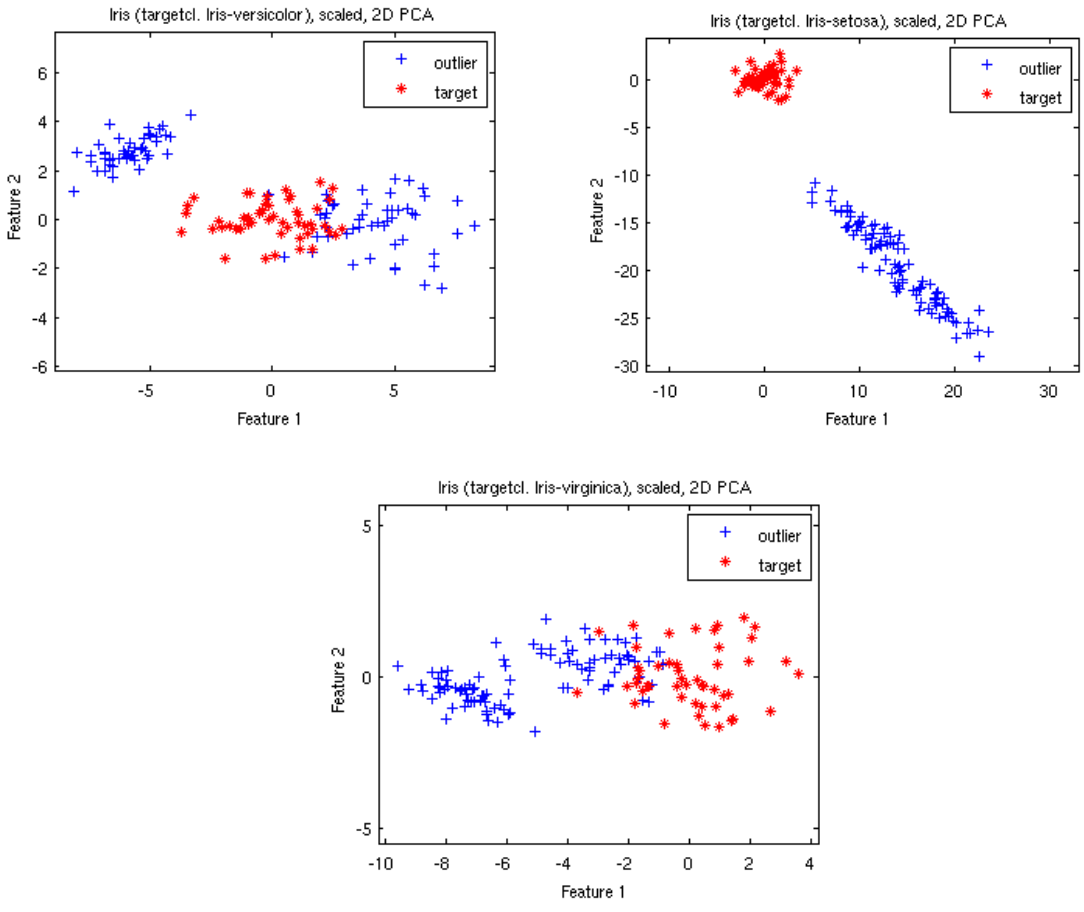
where $Q_{ij} = K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$

and α_i are the Lagrange multipliers, l is the total number of training samples provided, ν is a parameter that lets the user define the fraction of target class points rejected, K is the kernel matrix and ρ is the bias term. This results in a binary function which returns $+1$ or -1 for target class and outliers respectively and is called the decision function. The decision function thus obtained is

$$f(x) = sign(\sum_{i=1}^{l} \alpha_i K(x_i, x) - \rho) \tag{2.3}$$

2.1.2 Non-OCSVM

Ridder et al. conduct an experimental comparison of various OCC algorithms, including: (a) Global Gaussian approximation; (b) Parzen density estimation; (c) 1-Nearest Neighbor method; and (d) Gaussian approximation (combines aspects of (a) and (b)). Manevitz and Yousef [13] trained a simple neural network to filter documents when only positive information is available. To incorporate the restriction of availability of positive examples only, they used a three-level feed forward network with a "bottleneck". DeComite et al. [14] modify the C4.5 decision tree algorithm [15] to get an algorithm that takes as input a set of labeled examples, a set of positive examples, and a set of unlabeled data, and then use these three sets to construct the decision tree. Letouzey et al. [16] design an algorithm which is based on positive statistical queries (estimates for probabilities over the set of positive instances)and instance statistical queries (estimates for probabilities over the instance space). They design a decision tree induction algorithm, called POSC4.5, using only positive and unlabeled data. They present experimental results on UCI data sets that are comparable to the C4.5 algorithm. Wang et al. [17] investigate several one-class classification methods in the context of Human-Robot interaction for face and non-face classification. Some of the noteworthy methods used in their study are: (a) SVDD; (b) Gaussian data description; (c) KMEANS-DD; (d) Principal Component Analysis-DD. In their experimentation, they observe that SVDD attains better performance than the other OCC methods they studied.



2.2 Kernel Method

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation.

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "kernel trick"[1]. Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors.

Algorithms capable of operating with kernels include the kernel perceptron, support vector machines (SVM), Gaussian processes, principal components analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters and many others. Any linear model can be turned into a non-linear model by applying the kernel trick to the model: replacing its features (predictors) by a kernel function.

Most kernel algorithms are based on convex optimization or eigenproblems and are statistically well-founded. Typically, their statistical properties are analyzed using statistical learning theory

In the next section we describe anomaly detection using One-Class SVM and more than one kernel.

2.3 Multiple Kernel Anomaly Detection

Das et al. [18] propose MKAD to detect anomalies in aviation data. Aviation data consists of features that can be grouped into two categories - (i) Real valued data such as flight velocity, altitude, flap angle, etc and (ii) Binary valued data such as cockpit switch positions. Single-kernel One-Class SVM cannot capture the different notions of similarity in the Real and Binary valued data.Instead a composite Kernel K is used. This composite kernel can be any valid combination of individual kernels.

2.3.1 Weighted sum

This is the method used by Das et al. [18]. Here the composite kernel is a simple weighted sum of the individual kernels computed over all or a subset of the features i.e. $K(x_i, x_j) = \sum_{p=1}^{n} \eta_p k_p(x_i, x_j)$, with $\eta_p \geq 0$ and $\sum_{p=1}^{n} \eta_p = 1$. Here $k_p(x_i, x_j)$ represents the p^{th} kernel computed for data points x_i and x_j, and η_p are to weight individual kernels. The dual of this optimization problem is similar to that of OCSVM, with the Kernel replaced by the composite Kernel.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j K(x_i,x_j) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \tfrac{1}{\nu l} \quad i = 0,\ldots,l, \\ & \sum_i \alpha_i = 1, \\ & \rho \geq 0 \end{aligned} \tag{2.4}$$

where α_i are the Lagrange multipliers, ν is again a parameter that lets the user define the fraction of training samples rejected, l is the total number of training samples provided, ρ is the bias and K is the kernel matrix. Once the α_i are obtained, the following decision function is computed.

$$f(x) = sign(\sum_{i=1}^{l}\alpha_i K(x_i,x) - \rho) \tag{2.5}$$

2.3.2 Product

MKAD as given by Das et al. [18] can be extended to other valid kernel combinations. One such combination is to take the product of the individual kernels i.e. $K(x_i,x_j) = \prod_{p=1}^{n} k_p(x_i,x_j)$. Here $k_p(x_i,x_j)$ represents the p^{th} kernel computed for data points x_i and x_j. The dual of this optimization problem is the same as in 2.4 except that the composite Kernel is the product of the individual kernels.

Note that here the advantage of the multiple kernel learning approach is to incorporate knowledge of the differing notions of similarity in the decision process. Thus, we are able to achieve an improvement in detecting anomalies in a system that involves various data sources. A fixed combination rule (here a weighted summation or product) assigns the same weight to a kernel which remains fixed over the entire input space. However, this does not take into account the underlying localities in the data. Assigning different weights to a kernel in a data dependent way may lead to a further improvement in detecting the anomalies. We explore this possibility in the next section.

Chapter 3

Analysis and Objectives

There have however been few attempts to transfer these ideas in multiple kernel learning to the domain of One-Class Classification and Anomaly Detection. Das et al. [18] propose a simple weighted sum of two kernels, each of which describes respectively, the discrete and continuous streams in aviation data. This method takes advantage of multiple kernel learning to incorporate the different notions of similarity in the two streams for the task of detecting anomalies in a heterogeneous system. Though the method takes advantage of the ability of multi-kernel learning to combine knowledge from different data sources, it does not take into account the first advantage of multiple kernel learning, which is to allow more flexibility in selection of kernels. As such MKAD is a useful algorithm for anomaly detection only when the input data is heterogeneous and therefore lacks wider applicability. This report extends the MKAD algorithm by providing a localized formulation of multiple kernel based OCSVM for anomaly detection.
The main contributions of this project are:

- We propose an optimization problem for data driven anomaly detection in which a convex combination of kernels is used, with weights assigned locally. This optimization problem is analogous to the conventional One Class SVM and can be solved similarly.
- Our algorithm achieves significantly better Gmean scores and at the same time uses a lesser number of support vectors compared to conventional OCSVM and MKAD. For demonstrating the credibility of our algorithm, we perform extensive testing using five fold cross validation on benchmark datasets.

The rest of the report is organized as follows. In section 4, we propose OCSVM based Localized Multiple Kernel Anomaly Detection (LMKAD). Section 5 describes the experimental setup and evaluates the proposed (LMKAD) and existing One-Class Classifiers (OCSVM and MKAD) against 7 benchmark datasets. The report concludes in Section 6.

Chapter 4

Design Proposal

4.1 Localized Multiple Kernel Anomaly Detection

In this section we propose Localized Multiple Kernel Anomaly Detection. By assigning weights in a data dependent way we intend to give more weights to kernel functions which best match the underlying locality of the data in different regions of the input space. We modify the decision function in the previous sections to the following

$$f(x) = \sum_{m=1}^{p} \eta_m(x) \langle \omega_m, \phi_m(x) \rangle - \rho \tag{4.1}$$

where $\eta_m(x)$ is the weight corresponding to each kernel and is given by the gating model. The value of $\eta_m(x)$ is a function of the input and is defined by the parameters of the gating model. These parameters are in turn learned from the data during optimization as shown later in this section. We rewrite the original One-Class SVM optimization problem with our new decision function to get the following primal optimization problem

$$\begin{aligned} \min_{\omega_m, \eta_m, \xi, \rho} \quad & \frac{1}{2} \sum_{m=1}^{p} \omega_m^T \omega_m - \rho + \frac{1}{\nu l} \sum_{i=1}^{l} \xi_i \\ \text{s.t.} \quad & \sum_{m=1}^{p} \eta_m(x) \langle \omega_m, \phi_m(x) \rangle \geq \rho - \xi_i \quad \forall i, \\ & \xi_i \geq 0 \qquad \forall i \end{aligned} \tag{4.2}$$

where ν is the rate of rejection, l is the total number of training samples, and ξ_i are the slack variables as usual. We now need to solve this optimization problem for the above parameters. However, we do not solve this optimization problem directly, but use a two-step optimization scheme inspired by Rakotomamonjy et al. [19] to find the values of the parameters of both the gating model, $\eta_m(x)$ and the parameters of the decision function.

Before starting the optimization procedure we initialize the value of $\eta_m(x)$. Then, in the first step of the procedure we treat $\eta_m(x)$ as constant and solve the optimization problem (4.2) for ω, ξ and ρ. Note that if we treat $\eta_m(x)$ as constant, this step is essentially the same as solving canonical OCSVM under certain conditions as we will explain shortly. Solving the canonical OCSVM returns the optimal value of the Objective function and the Lagrange multipliers. In the second step we update the value of the parameters of $\eta_m(x)$ using gradient descent on the Objective function. The updated parameters define a new $\eta_m(x)$ which is used for the next iteration. The above two steps are repeated until convergence. From the Objective function in

(4.2) and the constraints, for fixed $\eta_m(x)$ the Lagrangian of the primal problem is written as:

$$L_D = \frac{1}{2}\sum_{m=1}^{p}\omega_m^T\omega_m + \sum_{i=1}^{l}\left(\frac{1}{\nu l} - \beta_i - \alpha_i\right)\xi_i - \rho \\ -\sum_{i=1}^{l}\alpha_i\left(\sum_{m=1}^{p}\eta_m(x_i)\langle\omega_m, \phi_m(x_i)\rangle - \rho\right)$$

and taking the derivatives of the Lagrangian L_D with respect to the variables in 4.2 gives :

$$\frac{\partial L_D}{\partial \omega_m} \Rightarrow \omega_m = \sum_{i=1}^{l}\alpha_i\eta_m(x_i)\phi_m(x_i) \quad \forall m \tag{4.3}$$

$$\frac{\partial L_D}{\partial \rho} \Rightarrow \sum_{i=1}^{l}\alpha_i = 1 \tag{4.4}$$

$$\frac{\partial L_D}{\partial \xi_i} \Rightarrow \frac{1}{\nu l} = \beta_i + \alpha_i \tag{4.5}$$

Substituting, (4.3), (4.4), and (4.5) we obtain the dual problem,

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\alpha^T Q\alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1 \quad i = 0, \ldots, l, \\ & e^T\alpha = 1 \end{aligned} \tag{4.6}$$

where $Q_{ij} = K_\eta(x_i, x_j)$

Where the kernel matrix is defined as

$$K_\eta(x_i, x_j) = \sum_{m=1}^{p}\eta_m(x_i)\langle\phi_m(x_i), \phi_m(x_j)\rangle\eta_m(x_j) \tag{4.7}$$

The dual formulation is exactly the same as the original One Class SVM formulation with Kernel function $K_\eta(x_i, x_j)$. Multiplying the Kernel matrix with a non-negative value will still give a positive definite matrix [20]. Note that the locally combined Kernel function will therefore satisfy the Mercer's condition if the gating function is non negative for both input instances. This can be easily ensured by picking a non-negative $\eta_m(x)$.In order to choose from among the different kernels a gating model is used. Here, the gating model is defined as

$$\eta_m(x) = \frac{exp(\langle v_m, x\rangle + v_{m0})}{\sum_{k=1}^{p} exp(\langle v_m, x\rangle + v_{m0})}$$

And is characterized by the parameters v_m and v_{m0} . The above function is called the Softmax function and ensures that $\eta_m(x)$ is non-negative. Note that if we use a constant gating function, the algorithm reduces to that of MKAD [18], and assigns fixed weights over the entire input space.

We can say that the objective value of the dual formulation (4.6) is equal to the objective value of the primal (4.2) , by strong duality provided that $\eta_m(x)$ is fixed. Therefore the objective function of the dual, $J(\eta)$ can be used as to calculate the gradient of the primal formulation with

respect to the parameters of the gating model (v_m and v_{m0}).

$$\frac{\partial J(\eta)}{\partial v_{m0}} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{p}\alpha_i\alpha_j\eta_k(x_i)K_k(x_i,x_j)\eta_k(x_j) \\ (\delta_m^k - \eta_m(x_i) + \delta_m^k - \eta_m(x_j))$$

$$\frac{\partial J(\eta)}{\partial v_m} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{p}\alpha_i\alpha_j\eta_k(x_i)K_k(x_i,x_j)\eta_k(x_j) \\ (x_i[\delta_m^k - \eta_m(x_i)] + x_j[\delta_m^k - \eta_m(x_j)])$$

$$f \text{ where, } \quad \delta_m^k = \begin{cases} 1, \text{ if } m = k \\ 0, \text{ otherwise} \end{cases}$$

Once we obtain the updated values of the parameters v_m and v_{m0} we calculate the new value of $\eta_m(x)$ using the gating function. Now substituting the new value of $\eta_m(x)$ in (4.7) gives us the new $K_\eta(x_i, x_j)$ which we use to solve the dual formulation (4.6). We again update the value of the gating model parameters and repeat until convergence.

4.2 Algorithm

The entire algorithm can be summarized as follows :

Algorithm 1 LMKAD algorithm

1: Initialize the values of v_m and v_{m0} to random values
2: **while** $v_m^{(t+1)} - v_m^{(t)} \geq \epsilon$ or $v_{m0}^{(t+1)} - v_{m0}^{(t)} \geq \epsilon$ **do**
3: Calculate η_m using v_m^t and v_{m0}^t
4: Calculate $K_\eta(x_i, x_j)$ using the gating model
5: Solve canonical one class SVM with $K_\eta(x_i, x_j)$
6: $v_m^{(t+1)} \Leftarrow v_m^{(t)} - \mu^{(t)}\frac{\partial J(\eta)}{\partial v_m} \quad \forall m$
7: $v_{m0}^{(t+1)} \Leftarrow v_{m0}^{(t)} - \mu^{(t)}\frac{\partial J(\eta)}{\partial v_{m0}} \quad \forall m$
8: **end**

Once the algorithm converges and the final $\eta_m(x)$ and the Lagrange multipliers are obtained, the decision function can be rewritten as

$$f(x) = \sum_{i=1}^{n}\sum_{m=1}^{p}\alpha_i\eta_m(x)K_m(x,x_i)\eta_m(x_i) - \rho$$

The sign of this decision function tells us whether the given input is target or outlier. Also we compute the average error using the sum over the (target value - predicted value) on the training data. Then we set the bias term to this mean value.

Chapter 5

Experiments

5.1 Datasets

In this section, the performance of the algorithm is tested on 20 datasets, which are selected from two disciplines viz. medical and signal processing. These datasets were originally proposed for binary or multi-class classification problems. We adapt these datasets for the OCC task by considering one class as the target class and the remaining classes as representative of the outlier class.

5.2 Experimental Setup:

All the experiments[1] have been conducted on MATLAB 2016a in Windows 7 (64 bit) environment with 64 GB RAM, 3.00 GHz Intel Xeon processor. For implementing existing SVM based One-Class classifiers, LIBSVM package [5] is used. For implementing other existing One Class classifiers, DDTOOLS package by Tax [21] is used. For every dataset 5 fold cross-validation indices are generated and this procedure is repeated 5 times each time constituting a run. These indices are kept same throughout the experiment for all the classifiers. In 5-fold CV, 4 folds are used for training and 1 fold is used for testing. However, out of the 4-folds used for training, only samples from one of the classes (i.e. target class) are used for training the model. Samples from the other classes are used as validation samples to find optimal parameters. We calculate and report the the average Gmean for 5-fold cross validation over 5 runs. Gmean is defined as follows equation:

$$Gmean = \sqrt{precision * recall}$$

In our experiments, three commonly used Kernels are employed which are defined as follows:

1. Linear Kernel (l):
   $K_L(x_i, x_j) = x_i^T x_j$
2. Polynomial Kernel (p):
   $K_P(x_i, x_j) = (x_i^T x_j + 1)^q$
3. Gaussian Kernel (g):
   $K_G(x_i, x_j) = e^{-\frac{(x_i - x_j)^T (x_i - x_j)}{\sigma^2}}$

The order of the Polynomial Kernel is chosen through the parameter q to be 2 or 3. The value of σ used in the Gaussian Kernel is set to the average Euclidean distance between the points in the training data. The Linear Kernel has no special parameters. The kernel parameter set

[1] All presented results in this report are reproducible. Codes with datasets can be produced on request.

that has the highest Gmean on the validation folds is considered to be the best configuration and these parameters are used as input along with the training folds for training the model. The trained model is then evaluated over the test set. Since we use 5-fold CV and repeat the experiment five times, for each dataset, we have twenty five test set results from which we find and report the average Gmean value and the average percentage of support vectors used. The same setup is followed for evaluating MKAD, One Class SVM and other one class classifiers . While evaluating MKAD we run the experiment twice, first where the composite kernel is taken as the mean of the individual kernels. Second, we run it with the composite kernel taken as the product of the individual kernels. We have also performed z-score normalization on each dataset before training and testing.

5.3 Results and Discussion:

In order to illustrate the significance of the multiple kernel approach, we have performed extensive experiments on 7 triple combinations of kernels for the existing (MKAD) as well as proposed (LMKAD) method. Table 5.1and 5.2 shows the results of OCSVM for Linear and Polynomial Kernels along with the results for MKAD and LMKAD for two combinations of Linear and Polynomial Kernels viz., (l-l-l) and (p-p-p). Table 5.3 presents results based on the two more possible combinations of Linear and Polynomial Kernels viz., (p-l-l) and (p-p-l). Similarly, Table 5.4 presents results of OCSVM for Gaussian Kernel along with results for MKAD and LMKAD for the triplet combination of Gaussian kernel with Polynomial and Linear respectively i.e. (g-p-p) and (g-l-l). Lastly, Table 5.5 presents the combination of all three Kernels i.e. (g-p-l).

TABLE 5.1: Linear Kernels

DATASET	OCSVM Linear		MKAD Sum l-l-l		MKAD Prod l-l-l		LMKAD l-l-l	
	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV
Iris setosa	59.21	21.30	58.77	27.50	58.25	62.50	57.74	25.00
Iris versicolor	48.80	25.80	57.74	25.00	57.74	57.50	57.74	27.50
Ionosphere good	62.77	40.24	80.07	35.56	80.07	33.33	80.32	5.56
Ionosphere bad	45.60	57.42	59.91	55.45	59.69	95.05	59.91	47.52
Diabetes present	55.79	8.79	80.69	7.50	80.69	30.00	80.69	40.75
Diabetes absent	47.94	12.39	59.07	10.28	59.07	40.65	59.04	41.59
Liver 1	55.00	15.83	64.83	11.21	64.83	41.38	64.83	52.59
Liver 2	61.32	13.43	76.14	10.00	76.14	36.88	76.14	11.88
Breast Malignant	63.69	48.18	79.21	32.87	79.21	60.14	79.25	21.33
Breast Benign	43.83	53.51	61.04	34.12	61.04	69.41	61.04	28.24
German credit (good risk)	68.51	16.64	83.67	14.82	83.67	66.43	83.67	33.57
German credit (bad risk)	45.54	26.50	54.77	19.58	54.77	86.25	54.77	45.42
Australia credit (good risk)	43.48	17.17	66.70	14.29	66.70	62.86	66.70	54.29
Autralia credit (bad risk)	52.70	15.10	74.50	10.78	74.50	53.92	74.44	43.14
Japan credit (good risk)	46.63	18.74	67.20	13.56	67.20	60.59	67.20	31.78
Japan credit (bad risk)	54.30	15.67	74.05	11.89	74.05	51.40	74.05	30.07
Heart diseased	56.23	26.13	73.40	19.53	73.40	84.38	73.40	42.97
Heart healthy	55.34	28.17	67.91	24.55	67.91	85.45	67.91	50.00
Parkinson patient	56.51	34.38	70.71	29.81	70.71	35.34	70.71	20.19
Parkinson Healthy	58.50	31.95	70.71	26.92	70.71	40.14	70.69	5.77

These results clearly indicate that multiple kernel approaches i.e. MKAD and LMKAD, outperform conventional One Class SVM with respect to Gmean scores. For the Linear Kernel combinations in Table 5.1, the results of LMKAD(l-l-l) are similar or better compared to MKAD(l-l-l). However, at the same time, LMKAD uses fewer support vectors than MKAD. A similar observation can be made for the Polynomial kernel combination in Table 5.2 with the exception of iris and ionosphere datasets. For these two datasets, LMKAD (p-p-p) significantly outperforms MKAD (p-p-p) by margins of 35% and 8%.

TABLE 5.2: Polynomial Kernels

DATASET	OCSVM Polynomial		MKAD Sum p-p-p		MKAD Prod p-p-p		LMKAD p-p-p	
	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV
Iris setosa	65.90	43.90	57.74	30.00	58.10	80.00	69.76	32.50
Iris versicolor	50.03	15.40	59.86	27.50	58.67	70.00	69.29	40.00
Ionosphere good	74.82	9.87	80.11	19.44	80.07	44.44	87.81	19.44
Ionosphere bad	49.46	25.95	59.91	73.27	59.51	97.03	59.91	76.24
Diabetes present	76.96	7.37	80.69	12.00	80.69	58.50	80.68	19.00
Diabetes absent	54.33	11.20	59.07	21.96	59.07	80.84	59.52	35.05
Liver 1	60.44	10.28	64.83	20.69	64.83	67.24	64.92	14.66
Liver 2	73.97	7.93	76.14	15.63	76.14	56.25	76.07	16.88
Breast Malignant	73.75	9.55	79.21	24.13	79.21	74.48	83.51	20.98
Breast Benign	54.63	12.81	61.04	28.24	61.04	76.47	61.56	28.24
German credit (good risk)	76.65	15.05	83.67	31.25	83.67	93.04	83.67	32.68
German credit (bad risk)	43.23	32.03	54.77	52.92	54.77	98.75	54.69	52.92
Australia credit (good risk)	58.63	21.70	66.70	34.29	66.70	86.53	66.67	33.06
Autralia credit (bad risk)	66.34	16.05	74.50	29.41	74.50	86.93	74.52	28.76
Japan credit (good risk)	60.74	15.65	67.20	30.51	67.20	87.29	67.21	30.93
Japan credit (bad risk)	65.44	18.52	74.05	30.77	74.05	82.87	74.07	50.70
Heart diseased	50.50	58.88	73.40	48.44	73.65	96.88	73.40	50.00
Heart healthy	51.72	35.37	67.91	58.18	67.91	98.18	67.91	58.18
Parkinson patient	66.41	8.17	91.53	13.22	70.71	60.82	99.41	21.88
Parkinson Healthy	66.30	8.15	70.71	15.87	70.71	60.82	70.94	31.01

TABLE 5.3: Kernel combination of Polynomial and Linear kernels

DATASET	MKAD-Prod p-l-l		MKAD Sum p-l-l		LMKAD p-l-l		MKAD-Prod p-p-l		MKAD Sum p-p-l		LMKAD p-p-l	
	Gmean	SV	Gmean	SV	Gmean	SV	SV		Gmean	SV	Gmean	SV
Iris setosa	60.76	50.00	57.77	67.50	57.74	37.50	58.06	32.50	57.77	72.50	57.77	35.00
Iris versicolor	59.54	45.00	60.25	60.00	58.04	27.50	61.50	27.50	58.91	62.50	60.02	42.50
Ionosphere good	80.02	32.78	80.07	35.56	82.25	14.44	80.07	20.56	80.11	45.00	87.25	18.89
Ionosphere bad	59.91	75.25	59.91	93.07	59.91	64.36	59.91	74.26	59.20	96.04	59.91	71.29
Diabetes present	80.69	14.50	80.69	40.25	80.69	12.75	80.69	14.00	80.69	46.50	80.60	12.00
Diabetes absent	59.21	22.90	59.07	60.28	59.07	20.56	59.07	21.50	59.07	71.50	59.07	21.96
Liver 1	64.85	25.86	64.83	57.76	64.83	24.14	64.83	21.55	64.83	56.90	64.83	22.41
Liver 2	76.14	19.38	76.14	43.13	76.08	17.50	76.14	16.88	76.14	44.38	76.14	16.25
Breast Malignant	79.21	23.78	79.21	51.05	80.25	16.08	79.21	23.78	79.21	77.62	81.44	17.48
Breast Benign	61.04	32.35	61.04	56.47	62.14	28.24	61.04	29.41	61.04	85.29	61.57	27.06
German credit (good risk)	83.67	33.75	83.67	80.71	83.67	30.89	83.67	34.29	83.67	89.29	83.67	31.25
German credit (bad risk)	54.77	58.75	54.77	93.33	54.77	53.33	54.77	57.50	54.77	96.67	54.77	55.42
Australia credit (good risk)	66.70	35.92	66.70	76.33	66.70	33.88	66.70	33.47	66.70	81.22	66.70	32.24
Autralia credit (bad risk)	74.61	33.66	74.50	75.49	74.50	29.08	74.50	30.39	74.50	81.37	74.49	29.74
Japan credit (good risk)	67.20	32.63	67.20	76.27	67.20	32.20	67.20	31.78	67.20	80.93	67.14	33.47
Japan credit (bad risk)	74.05	29.02	74.05	72.38	74.05	26.92	74.05	30.07	74.05	79.37	73.92	29.72
Heart diseased	73.50	46.88	73.40	88.28	73.40	48.44	73.40	47.66	73.48	92.97	73.40	48.44
Heart healthy	67.91	61.82	67.91	95.45	67.84	57.27	67.94	60.00	67.84	98.18	67.91	59.09
Parkinson patient	71.25	15.38	70.71	41.59	94.42	14.90	86.72	15.63	70.71	59.62	98.33	14.66
Parkinson Healthy	70.70	16.59	70.71	43.99	70.99	21.39	70.71	17.55	70.71	63.46	71.68	13.94

TABLE 5.4: Kernel combinations of Gaussian Kernel with Polynomial and Linear Kernels

DATASET	OCSVM Gaussian		MKAD Sum g-p-p		MKAD Prod g-p-p		LMKAD g-p-p		MKAD Sum g-l-l		MKAD Prod g-l-l		LMKAD g-l-l	
	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV	Gmean	SV
Iris setosa	92.10	13.80	60.38	32.50	57.74	67.50	98.96	30.00	66.19	17.50	57.74	40.00	100.00	22.50
Iris versicolor	89.77	14.10	60.06	25.00	57.74	55.00	77.44	30.00	63.83	17.50	57.74	40.00	73.77	20.00
Ionosphere good	91.82	6.67	85.24	16.11	80.07	45.00	93.69	16.67	80.98	8.89	80.07	24.44	89.80	10.56
Ionosphere bad	50.87	17.54	59.91	43.56	59.91	93.07	59.91	45.54	59.91	28.71	59.91	83.17	59.91	27.72
Diabetes present	79.63	5.46	80.69	11.00	80.69	39.75	80.64	17.25	80.69	6.00	80.69	13.25	80.65	6.75
Diabetes absent	57.06	5.95	59.07	17.29	59.07	63.55	58.88	7.94	59.07	7.48	59.07	26.17	59.02	7.94
Liver 1	63.16	6.93	64.83	17.24	64.83	51.72	65.01	17.24	64.85	8.62	64.83	27.59	65.28	9.48
Liver 2	73.75	5.80	76.14	15.63	76.14	38.75	76.02	23.13	76.14	8.13	76.14	20.00	76.14	8.75
Breast Malignant	93.02	5.78	86.58	16.08	79.21	58.04	91.96	14.69	85.40	7.69	80.01	24.13	90.94	7.69
Breast Benign	72.21	6.96	60.86	14.71	61.04	63.53	62.15	9.41	60.99	8.82	61.04	31.76	61.16	10.00
German credit (good risk)	81.62	5.78	83.67	28.04	83.67	80.18	83.63	18.93	83.67	7.32	83.67	34.64	83.67	8.21
German credit (bad risk)	53.28	7.13	54.67	38.75	54.77	92.92	54.64	59.58	54.77	12.92	54.77	55.83	54.67	13.75
Australia credit (good risk)	64.59	6.22	66.70	21.63	66.70	73.47	66.31	38.78	66.70	8.57	66.70	39.59	66.70	9.80
Autralia credit (bad risk)	76.63	5.81	75.38	20.26	74.50	74.51	76.27	41.18	75.35	7.84	74.72	34.31	75.76	9.48
Japan credit (good risk)	69.42	6.51	67.20	23.73	67.20	72.46	66.68	37.71	67.20	8.47	67.20	36.02	67.20	9.75
Japan credit (bad risk)	75.73	5.71	74.82	23.78	74.05	71.33	75.98	38.11	74.79	8.04	74.18	33.57	75.25	7.34
Heart diseased	76.22	7.84	73.40	41.41	73.42	90.63	73.34	59.38	73.40	13.28	73.40	55.47	73.40	17.97
Heart healthy	68.60	8.32	67.91	42.73	67.91	96.36	67.61	37.27	67.91	13.64	67.91	63.64	67.76	16.36
Parkinson patient	91.26	5.60	93.19	13.94	70.71	41.35	99.24	11.78	87.90	7.21	92.77	16.11	92.81	6.73
Parkinson Healthy	70.41	5.40	70.80	12.50	70.71	44.95	72.71	15.14	70.90	6.49	70.71	17.07	70.77	7.45

TABLE 5.5: Kernel combination containing all kernels

DATASET	MKAD-Prod g-p-l		MKAD Sum g-p-l		LMKAD g-p-l	
	Gmean	SV	Gmean	SV	Gmean	SV
Iris setosa	62.93	27.50	57.74	57.50	99.79	27.50
Iris versicolor	59.41	22.50	57.74	55.00	73.67	30.00
Ionosphere good	82.87	14.44	80.14	36.11	92.72	12.78
Ionosphere bad	59.91	39.60	59.91	94.06	60.05	38.61
Diabetes present	80.69	10.50	80.69	22.50	80.68	16.50
Diabetes absent	59.07	14.49	59.07	39.72	58.86	21.03
Liver 1	64.83	13.79	64.83	42.24	65.34	16.38
Liver 2	76.14	11.25	76.14	33.75	76.03	13.13
Breast Malignant	86.17	13.64	79.21	59.44	91.25	16.78
Breast Benign	60.95	14.71	61.04	72.94	61.71	14.71
German credit (good risk)	83.67	24.46	83.67	64.82	83.67	23.57
German credit (bad risk)	54.78	37.08	54.77	86.67	54.65	36.25
Australia credit (good risk)	66.70	18.37	66.70	62.45	66.54	20.41
Autralia credit (bad risk)	75.36	22.22	74.50	60.78	76.02	30.72
Japan credit (good risk)	67.20	20.76	67.20	61.86	66.96	26.27
Japan credit (bad risk)	74.79	20.63	74.05	54.55	75.57	33.57
Heart diseased	73.40	34.38	73.40	81.25	73.42	43.75
Heart healthy	67.91	40.00	67.91	87.27	67.84	32.73
Parkinson patient	91.27	11.30	70.71	39.42	96.06	12.26
Parkinson Healthy	70.83	12.02	70.71	39.90	71.38	8.89

For the rest of the combinations presented in Tables 5.3 to 5.5, LMKAD outperformed OCSVM and performed similar to or significantly better (for iris, ionosphere and wdbc) when compared to MKAD. Impact of localization can be observed in LMKAD (g-l-l), where even though the Gaussian kernel is locally combined with the simplest kernel i.e. Linear kernel, it either outperformed or produced a similar performance when compared to most Kernel combinations in MKAD and also used fewer support vectors.

TABLE 5.6: Increasing order of FRank and their MGmean value of all the one-class classifiers

One-class Classifier	FRank	MGmean
LMKAD(g-p-l)	3.86	83.39
LMKAD(g-p-p)	4.71	83.50
MKAD(g-p-p)	5.64	81.79
LMKAD(g-l-l)	6.64	82.92
LMKAD(p-p-p)	7.00	81.84
MKAD(g-p-l)	7.07	80.40
MKAD(g-l-l)	7.21	78.84
MKAD(l-l-l)	7.79	74.86
LMKAD(p-p-l)	8.43	79.78
MKAD(p-p-p)	8.71	75.50
MKAD(p-p-l)	10.36	75.01
LMKAD(p-l-l)	10.57	75.77
MKAD(p-l-l)	10.86	74.86
LMKAD(l-l-l)	11.29	75.05
OCSVM(g)	12.14	72.41
OCSVM (l)	14.93	71.61
OCSVM(p)	15.79	65.20

As we have a total of 17 variants of One-Class classifiers viz. 7 combinations of MKAD and LMKAD each and One-Class SVM for 3 different kernels, it is a cumbersome task to analyze them just by looking at the tables. Therefore, we have performed the Friedman test [22] to verify the statistical significance of these classifiers. Friedman Rank (FRank) and mean of Gmean (MGmean) over all 20 datasets for all the classifiers are presented in Table 5.6. This table presents all the One-Class classifiers in the increasing order of their FRank. This table shows that OCSVM stands last in the table. LMKAD(g-p-l) and LMKAD(g-p-p) stand first in the table in terms of FRank and MGmean respectively. It is to be noted that the difference in the FRank of LMKAD(g-p-l) and LMKAD(g-p-p) is significant, however, LMKAD(g-p-p) has a better MGmean. This shows that LMKAD(g-p-l) performs more stably compared to LMKAD(g-p-p). As expected from above discussion LMKAD(g-l-l) yields better MGmean compared to the rest of the classifiers, except LMKAD(g-p-l) and LMKAD(g-p-p). It can be observed from the table that all combinations of LMKAD perform better compared to their corresponding MKAD in term of FRank as well as MGmean, except LMKAD(l-l-l). In the all Linear Kernel combination case , LMKAD(l-l-l) performs better in terms of MGmean but not in terms of FRank. The p-value of the Friedman test is 0.000028, which is very less. The low p-value indicates that we can reject the null hypothesis and state that presented results of the various classifiers are statistically significant.

TABLE 5.7: Other One Class Classifiers

DATASET	Autoencdd			Gaussdd			kndd		
	Accuracy	Gmean	Std dev(GM)	Accuracy	Gmean	Std dev(GM)	Accuracy	Gmean	Std dev(GM)
Iris setosa	97.07	95.39	0.96	96.00	93.66	0.77	97.73	96.44	1.23
Iris versicolor	94.27	91.52	1.75	95.33	92.90	1.38	93.73	90.98	3.15
Ionosphere good	87.40	90.64	0.75	91.45	93.29	0.27	91.62	93.54	0.42
Ionosphere bad	33.73	56.89	0.89	26.78	41.17	0.83	33.85	56.93	0.54
Diabetes present	66.56	79.71	0.22	66.85	79.81	0.18	67.89	80.42	0.33
Diabetes absent	35.39	56.94	0.36	35.44	56.89	0.23	35.31	57.15	0.32
Liver 1	45.28	63.23	0.68	45.28	63.28	0.28	44.58	63.77	0.74
Liver 2	56.58	73.57	0.38	56.52	73.76	0.29	56.35	73.98	0.43
Breast Malignant	90.26	92.50	0.60	88.44	91.07	0.25	88.08	91.02	0.27
Breast Benign	42.28	60.41	0.56	42.88	60.19	0.36	40.98	60.31	0.18
German credit (good risk)	68.58	81.67	0.21	68.94	81.74	0.11	69.14	82.04	0.31
German credit (bad risk)	34.10	53.35	0.38	37.44	53.27	0.54	34.48	53.96	0.31
Australia credit (good risk)	48.73	65.93	0.26	50.66	66.42	0.42	49.39	66.34	0.18
Autralia credit (bad risk)	67.28	77.74	0.55	66.87	77.44	0.29	67.48	77.83	0.26
Japan credit (good risk)	58.29	69.96	0.96	61.85	71.44	0.67	51.31	67.44	0.35
Japan credit (bad risk)	67.28	77.61	0.38	66.85	77.23	0.20	67.03	77.61	0.19
Heart diseased	69.08	77.92	1.04	69.09	76.95	0.60	67.01	77.02	0.51
Heart healthy	50.04	66.27	0.84	51.91	63.58	1.64	48.61	66.61	1.11
Parkinson patient	90.33	90.83	0.79	97.02	96.96	0.26	97.52	97.48	0.25
Parkinson Healthy	55.23	70.73	0.38	61.73	73.21	0.20	72.44	78.68	0.11

TABLE 5.8: Other One Class Classifiers

DATASET	pcadd			somdd			svdd		
	Accuracy	Gmean	Std dev(GM)	Accuracy	Gmean	Std dev(GM)	Accuracy	Gmean	Std dev(GM)
Iris setosa	79.73	78.76	2.57	97.20	95.60	0.93	95.07	92.10	1.04
Iris versicolor	91.47	87.62	2.08	89.47	85.60	2.06	93.47	89.77	1.33
Ionosphere good	93.27	94.67	0.83	83.82	88.49	0.65	88.83	91.80	0.28
Ionosphere bad	31.39	40.80	0.49	33.39	56.41	0.92	29.91	50.87	0.75
Diabetes present	67.53	79.96	0.36	66.69	79.90	0.40	65.89	80.32	0.14
Diabetes absent	36.33	56.62	0.60	35.78	57.18	0.34	35.81	58.07	0.24
Liver 1	44.58	62.80	0.82	44.81	63.56	0.34	44.81	64.13	0.56
Liver 2	56.17	73.30	0.54	56.29	73.73	0.47	56.70	74.38	0.22
Breast Malignant	82.68	87.50	0.42	89.84	92.16	0.64	89.91	92.51	0.27
Breast Benign	40.81	59.32	0.37	42.88	60.88	0.78	60.63	68.28	0.81
German credit (good risk)	68.38	80.78	0.35	68.66	81.73	0.14	68.70	81.84	1.40
German credit (bad risk)	38.16	50.84	0.87	33.90	53.34	0.24	33.98	54.11	0.26
Australia credit (good risk)	53.66	66.81	1.07	48.61	66.10	0.62	44.90	65.85	0.23
Autralia credit (bad risk)	67.65	77.04	1.01	68.03	78.17	0.38	63.42	76.99	0.23
Japan credit (good risk)	56.61	68.64	0.19	57.91	69.88	0.70	46.14	66.83	0.16
Japan credit (bad risk)	67.49	76.93	0.39	67.49	77.61	0.09	62.76	76.32	0.17
Heart diseased	63.25	72.62	0.33	70.44	78.77	0.35	64.79	76.06	0.38
Heart healthy	44.99	60.09	2.14	52.53	68.04	0.53	56.03	68.47	1.24
Parkinson patient	96.94	96.89	0.18	89.44	90.06	0.28	91.33	91.83	1.30
Parkinson Healthy	84.21	86.01	0.45	53.27	70.04	0.10	52.04	70.44	0.13

TABLE 5.9: Best kernel combinations of LMKAD, MKAD(sum) and MKAD(Product)

DATASET	LMKAD			MKAD-Weighted Sum			MKAD-Product		
	Accuracy	Gmean	SV	Accuracy	Gmean	SV	Accuracy	Gmean	SV
Iris setosa	100.00	100.00	22.50	46.67	66.19	17.50	35.33	58.25	62.50
Iris versicolor	77.20	77.44	30.00	50.13	63.83	17.50	40.27	60.25	60.00
Ionosphere good	91.11	93.69	16.67	75.79	85.24	16.11	64.28	80.14	36.11
Ionosphere bad	36.35	60.05	38.61	35.90	59.91	55.45	35.90	59.91	93.07
Diabetes present	65.10	80.69	40.75	65.10	80.69	7.50	65.10	80.69	30.00
Diabetes absent	37.16	59.52	35.05	35.47	59.21	22.90	34.90	59.07	40.65
Liver 1	43.71	65.34	16.38	42.09	64.85	8.62	42.03	64.83	41.38
Liver 2	57.97	76.14	11.88	57.97	76.14	10.00	57.97	76.14	36.88
Breast Malignant	88.79	91.96	14.69	79.13	86.58	16.08	64.71	80.01	24.13
Breast Benign	42.57	62.15	9.41	37.26	61.04	34.12	37.26	61.04	69.41
German credit (good risk)	70.00	83.67	33.57	70.00	83.67	14.82	70.00	83.67	66.43
German credit (bad risk)	30.00	54.77	45.42	30.02	54.78	37.08	30.00	54.77	86.25
Australia credit (good risk)	44.49	66.70	54.29	44.49	66.70	14.29	44.49	66.70	62.86
Autralia credit (bad risk)	60.52	76.27	41.18	57.80	75.38	20.26	56.09	74.72	34.31
Japan credit (good risk)	45.19	67.21	30.93	45.16	67.20	13.56	45.16	67.20	60.59
Japan credit (bad risk)	60.37	75.98	38.11	57.02	74.82	23.78	55.18	74.18	33.57
Heart diseased	53.94	73.42	43.75	54.62	73.50	46.88	54.55	73.65	96.88
Heart healthy	46.12	67.91	50.00	46.12	67.91	24.55	46.12	67.91	85.45
Parkinson patient	99.40	99.41	21.88	92.44	93.19	13.94	91.88	92.77	16.11
Parkinson Healthy	56.27	72.71	15.14	50.67	70.90	6.49	50.00	70.71	40.14

TABLE 5.10: Summary of Classifier Performance

One-Class Classifiers	FRank	MGmean
LMKAD	3.65	75.25
Knndd	4.10	75.48
MKAD-Sum	4.90	71.59
Svdd	5.05	74.55
MKAD-Prod	5.05	70.33
Somdd	5.10	74.36
Gaussdd	5.30	74.21
Autoencdd	5.50	74.64
Pcadd	6.35	72.90

We further compare the results of LMKAD and MKAD with other existing One Class Classifiers in the tables 5.7, 5.8, 5.9. Here, for LMKAD and MKAD, the best kernel combinations for each dataset is shown. Again as the number of classifiers are very large, we have calculated the FRank and MGmean for each classifier so as to be able to compare them. Table 5.10 shows the corresponding Frank and MGmean for each classifier. We can see that LMKAD with a score of 3.65 has the lowest FRank and is therefore at the top performing better than all existing one class classifiers. Also LMKAD performs better than both MKAD(Sum) and MKAD(Product). In fact some other classifiers perform better than MKAD. This shows that though MKAD is useful in cases where the data is Heterogeneous, it lacks wider applicability and does not perform well on homogeneous data. Proposed classifier LMKAD however is able to exploit the localities in the input data space and performs better than the other classifiers.

Chapter 6

Conclusion and Future Work

The objectives of this project were to

- Introduce a new and novel method for anomaly detection
- Prove its correctness
- Evaluate the performance of this new method against existing state-of-the-art methods.
- Showcase the statistical significance of our results.

In this project, we have proposed a new method of Anomaly Detection by extending MKAD and creating a localized formulation for Multi kernel learning for anomaly detection based on a local assignment of weights. Previous multiple kernel based anomaly detection method i.e. MKAD has its strengths and shortcomings, which are explained in the report and addressed by the proposed method.
We also proposed an optimization problem based on the localized decision function and described a two-step optimization procedure to solve it. The derived formulation is also shown to be analogous to conventional One-Class SVM and is solved in a similar fashion using a LIBSVM solver. The algorithm is empirically tested against 20 benchmark datasets, and its performance proves the credibility of our approach.
The results are discussed under two broad groupings. First, the results among different Kernel combinations are studied in LMKAD and MKAD. Second,the best performing kernels, are chosen and compared against other One Class Classifiers. LMKAD outperforms both conventional One-Class SVM and MKAD for most of the datasets. For some other datasets, LMKAD performs similar to MKAD but uses fewer support vectors. The Friedman test is performed which statistically verifies that the algorithm performs significantly better than its counterparts.

Future work in this direction would be to extend the concept of Localized Multiple kernel Learning to SVDD and other Kernel based methods especially to Kernel Ridge regression.

Bibliography

[1] M. A. Pimentel et al. "A review of novelty detection". In: *Signal Processing* 99 (2014), pp. 215–249.

[2] D. M. J. Tax and R. P. W. Duin. "Support vector domain description". In: *Pattern recognition letters* 20.11 (1999), pp. 1191–1199.

[3] B. Schölkopf et al. "Support Vector Method for Novelty Detection." In: *NIPS*. Vol. 12. 1999, pp. 582–588.

[4] D. M. J. Tax. "One-class classification; Concept-learning in the absence of counter-examples". In: *ASCI dissertation series* 65 (2001).

[5] C.-C. Chang and C.-J. Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, pp. 1–27.

[6] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[7] Francis R. Bach, Gert RG. Lanckriet, and Michael I. Jordan. "Multiple kernel learning, conic duality, and the SMO algorithm". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 6.

[8] M. Gönen and E. Alpaydin. "Localized multiple kernel learning". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 352–359.

[9] M. Goene and E. Alpaydin. "Localized multiple kernel machines for image recognition". In: *Neural Information Processing Systems—Workshop on Understanding Multiple Kernel Learning Method*. 2009.

[10] Larry M Manevitz and M. Yousef. "One-class SVMs for document classification". In: *Journal of Machine Learning Research* 2.Dec (2001), pp. 139–154.

[11] S. Das and Nikunj C. Oza. "Sparse solutions for single class SVMs: A bi-criterion approach". In: *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM. 2011, pp. 816–827.

[12] N. Görnitz, M. Braun, and M. Kloft. "Hidden markov anomaly detection". In: *International Conference on Machine Learning*. 2015, pp. 1833–1842.

[13] Larry M. Manevitz and Malik Yousef. "One-class Svms for Document Classification". In: *J. Mach. Learn. Res.* 2 (Mar. 2002), pp. 139–154. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944790.944808.

[14] Francesco De Comité et al. "Positive and Unlabeled Examples Help Learning". In: *Algorithmic Learning Theory: 10th International Conference, ALT'99 Tokyo, Japan, December 6–8, 1999 Proceedings*. Ed. by Osamu Watanabe and Takashi Yokomori. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 219–230. ISBN: 978-3-540-46769-4. DOI: 10.1007/3-540-46769-6_18. URL: https://doi.org/10.1007/3-540-46769-6_18.

[15] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1-55860-238-0.

[16] François Denis, Rémi Gilleron, and Fabien Letouzey. "Learning from positive and unlabeled examples". In: *Theoretical Computer Science* 348.1 (2005). Algorithmic Learning Theory (ALT 2000), pp. 70 –83. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2005.09.007. URL: http://www.sciencedirect.com/science/article/pii/S0304397505005256.

[17] X. Wang and M. Han. "Online sequential extreme learning machine with kernels for non-stationary time series prediction". In: *Neurocomputing* 145 (2014), pp. 90–97.

[18] S. Das et al. "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2010, pp. 47–56.

[19] A. Rakotomamonjy et al. "More efficiency in multiple kernel learning". In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 775–782.

[20] S. Amari and S. Wu. "Improving support vector machine classifiers by modifying kernel functions". In: *Neural Networks* 12.6 (1999), pp. 783–789.

[21] DMJ Tax. "Ddtools, the data description toolbox for matlab". In: *Delft University of Technology ed* (2005).

[22] J. Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *Journal of Machine learning research* 7.Jan (2006), pp. 1–30.