# Driving Business Analytics through Machine Learning and Big Data

ISBN: 978-81-924713-8-9

**Rekha A G**
**Mohammed Shahid Abdulla**
*Indian Institute of Management*
(rekha05fpm@iimk.ac.in)
(shahid@iimk.ac.in)

**Asharaf S.**
*Indian Institute of Technology Management*
(asharaf.s@iiitmk.ac.in)

*The expanding corporate data volumes pose challenges as well as opportunities for business analytics applications. The present day business environments demand real time processing of data to support real time decision making for keeping the business competitive. Machine learning techniques have been successfully applied in various domains such as finance, retail, marketing etc. to gain insights from data and to improve decision making. These applications need to have the capacity to process large volumes of data and hence presents various implementation challenges. This work focuses on developing machine learning techniques which can process big data for the purpose of business analytics. Classification is one of the most studied data mining techniques and One Class Classification (OCC) has got various applications in business analytics. Support Vector Data Description (SVDD) is a popular kernel based OCC method used for outlier detection. But because of the computational complexity of SVDD they are not preferred in applications that require great classification speed. This work proposes three approaches based on SVDD to make efficient outlier detection possible using Big Data. Experiments conducted to assess feasibility of the approach shows that it has an order-of-magnitude advantage in terms of running time. This paper also discuss a use case from business and demonstrates the application of the proposed approach using a real world dataset obtained from UCI machine learning repository.*

**Keywords***: SVDD, Big Data, Hadoop,*

## 1. Introduction

Nowadays organizations gather terabytes of data from their customers and software applications. There is a need to analyze these massive datasets for various uses, e.g. providing services that are increasingly personalized. 'Big Data' as input enhances the inferential power of established algorithms, but it challenges even state-of-the-art computation and analysis methods. Classification tasks, especially one-class classification tasks, where single class information is available in high quality and resolution, and a few outliers exist, have numerous applications in business. Our research will focus on developing machine learning techniques on big data for the purpose of making one-class classification practical on large business databases.
Support Vector Data Description (SVDD) is a popular kernel based method used for one class classification [6] [12] [13][14]. The major drawback of this method is its computational complexity during the training phase and hence making it not feasible for very large datasets. The current best complexity to solve the SVDD training problem is O (N), an improvement from the original O (N3) as demonstrated in the core vector application of [8].

This work proposes and demonstrates a novel method LT-SVDD that could be used to analyze large data sets. The proposed algorithm reduces the complexity by avoiding the calculation of the Lagrange multipliers by locating an approximate pre-image of the SVDD sphere's center in the input space during the training phase itself. It retains the benefit of the kernel trick: i.e. a minimum enclosing space is more descriptive of the data when calculated in a higher-dimensional feature space. As a result we could reduce the complexity to O (d) where d is the dimension of the data set. The crux of the training algorithm is a gradient descent of the primal objective function using Simultaneous Perturbation Stochastic Approximation (SPSA). We also propose a variant of LT-SVDD named as ELT-SVDD to eliminate an inherent drawback of LT-SVDD. ie. the decision hyper sphere is restricted to a sphere in the input space. For dealing with real time applications on Big data we propose the hadoopization of ELT-SVDD which is named as HELT-SVDD.

### 1. Related Work

Support Vector Data Description (SVDD) is a popular one class classifier involving creation of a hyper-sphere in the pattern space around the target class data with the minimum radius to encompass almost all the target instances and exclude the non-target ones [8].
Given a set of data points, $x_i$; i=1: N, the objective is to

$$\text{minimize: } F(R,a) = R^2 + C\sum_i \xi_i \quad (1)$$

with the constraints $\|x_i - a\|^2 \leq R^2 + \xi_i,\ \xi_i \geq 0\ \forall i$
where $R$ is the radius and $a$ is the center of the sphere and the slack variables $\xi_i$ allow the possibility of outliers in the training set. For $N$ target patterns of dimension $d$ given for training, the current best complexity to solve SVDD training problem is

$O(N)$)-an improvement from the original $O(N^3)$as demonstrated in [2]. The objective function of SVDD, eqn(1) can be translated into the following form by employing Lagrangian multipliers $\alpha_i \geq 0$ and $\gamma_i \geq 0$ as:

$O(R, a, \alpha_i, \gamma_i, \xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i\{R^2 + \xi_i - (\|x_i\|^2 - 2a.x_i + \|a\|^2)\} - \sum_i \gamma_i \xi_i$ (2)

O should be minimized w.r.t. R, a,$\xi_i$ and maximized w.r.t $\alpha_i$and $\gamma_i$. The parameter C controls the trade-off between the volume and the errors and $\xi$ is the slack variable. Setting partial derivatives (w.r.t a and R) to zero gives the center of the sphere as a linear combination of the objects as given by: $a = \sum_i \alpha_i x_i$; the distance from the center of the sphere, a, to (any of the support vectors on) the boundary, $R^2$ is given by: $R^2 = (x_k.x_k) - 2\sum_i \alpha_i (x_i.x_k) + \sum_{i,j} \alpha_i \alpha_j (x_i.x_j)$

Also, $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0$.

The function becomes

$L = \sum_i \alpha_i (x_i.x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i.x_j)$

In all formulae, objects $x_i$ appear only in the form of inner products with other objects $(x_i.x_j)$ and these inner products can be replaced by Gaussian kernel function to obtain more flexible methods. By placing kernel functions $K(x,y)$ instead of the product $(x,y)$ in the above equations we have: $L = 1 - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$

Only objects $x_i$ with $\alpha_i > 0$ are needed in the description and these objects are therefore called the support vectors of the description (SV's). Hence, an object in feature space,z, is accepted by the description (stated to be within the boundary of the sphere) when: $\|(z-a)\|^2 = (z - \sum_i \alpha_i x_i)(z - \sum_i \alpha_i x_i) \leq R^2$. Similarly, by applying the kernel function, the decision function now becomes:

$1 - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_j \alpha_j K(x_i, x_j) \leq R^2$

Hence the complexity of SVDD approach is linear in the number of support vectors.

The testing time complexity of SVDD is reduced in [10] by calculating the pre-image of a point termed 'agent of the centre' of the SVDD sphere. The authors in [10] begin by solving the classic SVDD optimization problem of

Minimize $O_p(R, a_F, \xi_i) = R^2 + C\sum_{i=1}^N \xi_i$ (3)

Subject to $\|\phi(x_i) - a_F\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \forall i \in \{1..N\}$

Note that $a_F$ and the kernel-trick based transformation of input pattern $x_i$ , $\phi(x_i)$, are potentially vectors in infinite dimensional space. Since it is convenient using existing computational techniques, it is the dual of this problem that is solved:

Maximize $O_d(\alpha) = 1 - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j)$ (4)

Subject to $\sum_i^N \alpha_i = 1, 0 \leq \alpha_i \leq C \ \forall I \in \{1..N\} C \in \left[\frac{1}{N}, 1\right]$

The expression for the centre of the minimum enclosing ball is obtained as $a_F = \sum_{i=1}^{N_s} \alpha_i \phi(x_i)$.The key modification that [10] proposes relies on calculating a pre-image, in the input space, of this centre $a_F$. In particular, they show equivalence of the centre with another point named the agent of the centre $\psi_a$ on the input space's manifold in feature. They essentially solve the optimization problem $\min_{\tilde{x}} \|a_F - \phi(\tilde{x})\|^2$where $\tilde{x} \in R^m$ is in the space of input patterns and obtain a closed form for the solution $\tilde{x}$ as

$$\tilde{x} = \frac{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j) x_i}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j)}$$

By employing the Gaussian kernel, the pre-image of the agent of the sphere centre, $\tilde{x}$ can be calculated as :

$\tilde{x} = \frac{1}{\alpha^T K \alpha} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} x_i$

Once $\tilde{x}$ is known, the classification of a pattern x is now O(1) since the decision function can be calculated as explained in [10] as:

$D_f(x) = \|\phi(x) - a_F\|^2 - R^2 = \left\|\phi(x) - \frac{\phi(\tilde{x})}{\gamma}\right\|^2 - R^2 = c' - \frac{2}{\gamma} K(x, \tilde{x})$, where $c' = 1 - R^2 + 1/\gamma^2$ is a constant and $\gamma = \frac{\psi_a}{a_F}$.

This expression for $\tilde{x}$ is, however, calculable only after the training problem is solved in the regular fashion, i.e. by obtaining $\alpha_i$ in [10].

Studies suggest that primal optimization will be superior for large scale optimization[5], since when the number of training points is large, the number of support vectors will likely also be large thus resulting in updates of nearly N Lagrange multiplier parameters during optimization and a complicated decision function.

## 2. Proposed Methods

### 1.1 Lightly Trained SVDD (LT_SVDD)

The current work proposes a primal variant of the SVDD problem which calculates the pre-image $\tilde{x}$ in input space itself. In particular, what is obtained is an approximate solution to the SVDD problem since the exact pre-image of the feature space ball's centre does not exist in input space. Besides, in the training phase it uses the *primal* form of the kernelized SVDD problem as suggested in [6].

Fig 1 shows the geometric properties of SVDD. Here $B_F$ is the feature space unit sphere and $B_S$ is the SVDD sphere centered at $a_F$. $\psi_a$ is the agent of $a_F$. Also, $\phi(x_i)$ is any point on the SVDD sphere and 'A' is a support vector.

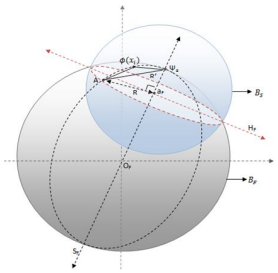

**Figure 1** *Geometric Properties of SVDD*

Using the geometric properties of SVDD we are making two claims.

***Claim1:*** *If any feature space pattern $\phi(x_i)$ is within the ($a_F$, R) SVDD hypersphere, then it is also within the ($\psi_a$, R') sphere.*

***Claim 2 :*** ***2.a)*** $\boldsymbol{a_F = \sqrt{1-R^2}\,\phi(\hat{x})}$ ***2. b)*** $\boldsymbol{(R')^2 = 2(1-\sqrt{1-R^2})}$

Using claims 2.a and 2.b we can simplify, the SVDD primal problem. Note the N constraints $\|\phi(x_i) - a_F\|^2 \leq R^2 + \xi_i$ and take $\beta = \sqrt{1-R^2}$

$$< \phi(x_i) - a_F, \phi(x_i) - a_F > \leq R^2 + \xi_i$$

$$1 + < a_F, a_F > -2 < \phi(x_i), a_F > \leq R^2 + \xi_i$$

$$1 + < \beta\psi_a, \beta\psi_a > -2 < \phi(x_i), \beta\psi_a > \leq R^2 + \xi_i \quad \text{(Using Claim 2a)}$$

$$1 + \beta^2 - 2\beta < \phi(x_i), \psi_a > \leq R^2 + \xi_i$$

The problem is still not soluble due to $a_F$ belonging to possibly infinite-dimensional feature space. But a further transformation is possible as below:

$$\text{Minimize} \quad O_p(R^2, \hat{x}, \xi_i) = R^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{S.T} \quad 1 + \beta^2 - 2\beta K(x_i, \hat{x}) \leq R^2 + \xi_i$$

$$\beta = \sqrt{1-R^2}$$

$$\xi_i \geq 0, \forall i \in \{1..N\}$$

$$R^2 > 0$$

In the training phase we have used Simultaneous Perturbation Stochastic Approximation (SPSA) adapted to sub-gradients as in [4] to solve the optimization problem.

The algorithm for LT-SVDD is as follows:

**Algorithm 1 : LT-SVDD**

**Training**

1. Initialize kernel parameters C and σ using same methods as C-SVDD
2. Solve the pre-image problem for agent of the sphere center by performing optimization in the modified primal to calculate the value of R and $\hat{x}$.

**Testing**

A data point $x_i$ is treated as typical if $1 + \beta^2 - 2\beta K(x_i, \hat{x}) \leq R^2$

### 1.1.1 Experiments

Initially simulations were performed using synthetic data sets. Training and testing files were created using 'pointscreator_gaussian.c' (sample distribution shown in Figure 2). C-SVDD and F-SVDD are implemented using MATLAB for comparison of accuracies as well as execution times. Preliminary experiments on these datasets consistently yielded promising accuracies. We have observed that as we increased the distance between the centre of distributions of train and test data sets, there is a steady increase in classification accuracy. Figure 3 shows the observed accuracies for different levels of overlapping between train and test data sets.

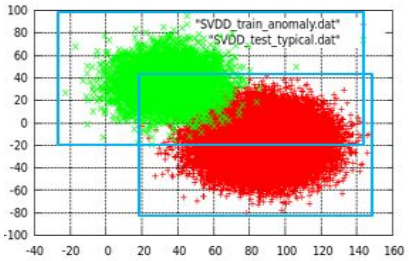


**Figure 2** *Distribution of Synthetic Datasets*

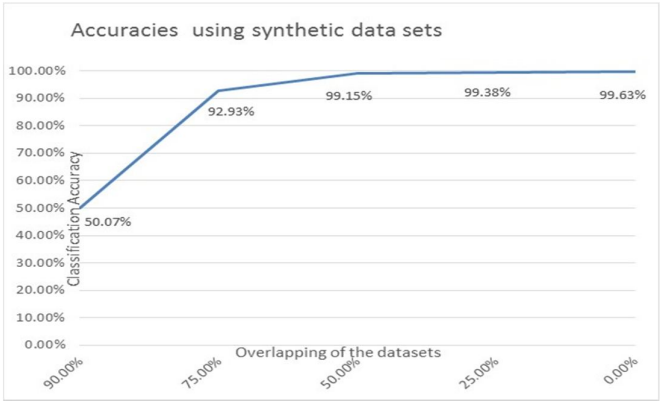


**Figure 2** *Results from Synthetic Data Sets*

Then we have conducted experiments on 3 datasets namely IRIS, WINE and CANCER obtained from the UCI repository. Our implementations of C-SVDD and primal based SVDD gave comparable accuracies as shown in Table 1.

**Table 1** *Comparison of Accuracies*

| Data set | d | Target Class | N-train | N-test | C-SVDD | F-SVDD | LT-SVDD |
|---|---|---|---|---|---|---|---|
| IRIS | 4 | 0 | 25 | 125 | 100 | 100 | 100 |
| | | 1 | 25 | 125 | 86.4 | 86.4 | 84 |
| | | 2 | 25 | 125 | 76.6 | 79.2 | 78.4 |
| WINE | 13 | 0 | 29 | 149 | 94.6 | 96.6 | 96.64 |
| | | 1 | 35 | 143 | 79 | 73.4 | 78.32 |
| | | 2 | 24 | 154 | 90.3 | 92.2 | 96.1 |
| CANCER | 9 | 0 | 222 | 461 | 95.4 | 81.3 | 93.71 |
| | | 1 | 119 | 564 | 93.1 | 95.2 | 95.74 |

**Direct Marketing Data set**

We have then conducted experiments using a real world business dataset related with direct marketing campaigns of a Portuguese banking institution. This dataset has been obtained from UCI and has been used in [9]. This data set is related to a tele-marketing campaign. The problem is to detect the potential customers for a product from the entire customer database. The dataset is analysed in [11]. Table 2 shows the accuracy values of C-SVDD and LT-SVDD from our implementations and a comparison with the accuracies from LAD tree algorithm, Radial Basis Function network and SVM obtained from [9]. Table 3 shows the comparison of execution times which indicates that LT-SVDD has a huge advantage over C-SVDD.

**Table 2** *Comparison of Accuracies*

| Model | Accuracy |
|---|---|
| **LAD tree algorithm** | **76.08%** |
| **Radial Basis Function network** | **74.34%** |
| **SVM** | **86.95%** |
| **C-SVDD** | **88.3%** |
| **LT-SVDD** | **90.00%** |

**Table 3 Comparison of** Execution Times

| Model | Training Time taken (in sec) |
|---|---|
| **C-SVDD** | 4.96 |
| **LT- SVDD** | 0.41 |

### 1.1.2 Limitations of LT_SVDD

Given the Gaussian kernel, the decision hypersphere of LT-SVDD becomes a sphere in the input space according to the computation of preimage in FSVDD as explained in [7]. Hence LT-SVDD may not be suitable for many real cases.

### 1.2 Efficient LT_SVDD (ELT_SVDD)

Peng et al. in [7] has explained an approach, named as Efficient-SVDD which performs a clustering on the training dataset before applying F_SVDD. They calculate the pre-images of the centroid points and then uses the expansion of the images of these pre-images to approximate the center of SVDD sphere. Here we apply a similar method to eliminate an inherent drawback of LT_SVDD, i.e. a restriction that the SVDD non-anomalous zone has to be a sphere in the input space. The proposed algorithm is as below:

**Algorithm 2 : (ELT-SVDD)**

**Training**

1. Initialize kernel parameters C and σ
2. Determine the number of clusters
3. Find the pre-images $\psi_{a_k} = \phi(\hat{x}_k)$ with k from 1..c
4. Calculate the approximation of the center

**Testing**

1. For the testing data point x, predict the output using the LT-SVDD decision function

### 1.2.1 Limitations of ELT_SVDD

ELT_SVDD will take slightly longer training time than LT-SVDD since clustering is a pre-processing step. Hence it will not be suitable for applications involving near real time responses and for Big data. In real life applications were datasets cross the petabyte threshold, the computational requirements are massive. Hence, efficiency is unlikely if these tasks are vertically integrated.

### 1.3 Hadoopized ELT_SVDD(HELT_SVDD)

Hadoop is a framework built for implementing reliable and scalable computational networks to support data intensive distributed applications with a master-slave architecture. Hadoopization of ELT_SVDD will be possible on a Big Data grid since individual nodes of this grid can be used to calculate each cluster's agent-of-center. Hence these computations can be parallelized. The proposed algorithm is as follows:

**Algorithm 3 : (HELT-SVDD)**

**Training**

1. Initialize kernel parameters C and σ
2. Determine the number of clusters c, parcel each cluster to one slave node
3. Find the pre-images $\psi_{a_k} = \phi(\hat{x}_k)$ ,k from 1..c using slave nodes 4. Calculate the approximation of the center at the master node

**Testing**

1. For the testing data point x, predict the output using the LT-SVDD decision function

## 3. Conclusions

In this work we proposed a method which solves the SVDD problem using the primal form and reduces the complexity from $O(N)$to $O(d)$ by locating an approximate pre-image of the SVDD sphere's center during the training phase itself. The use of SPSA allows us to calculate the gradient for primal gradient-descent even if there is no closed form for the first derivative. Experiments on both artificial and real-world business datasets have demonstrated that the proposed method is promising. The future work includes implementation of ELT_SVDD and HELT_SVDD and experimentations using large datasets. The outcome of hadoopization would be that analyzing of data streaming in real time would be possible. But an anticipated limitation would be that Big data environments consisting of dynamically growing datasets with a large varieties of class types can lead to inaccurate classification results.

## 4. References

1. Calvin S. Chu and Ivor W. Tsang and James T. Kwok, "Scaling up Support Vector Data Description by using Core Sets," IJCNN, 2004.
2. Calvin S. Chu and Ivor W. Tsang and James T. Kwok, "Scaling up Support Vector Data Description by using Core Sets," IJCNN, 2004.
3. M. C. Fu, and S. I. Marcus Y. He, "Fast support vector Fast support vector," IEEE Transactions on Automatic Control, vol. 48, no. 8, 2003.

4. M. S. Abdulla and S. Bhatnagar, "SPSA Algorithms with measurement reuse," in Proceedings of the 2006 Winter Simulation Conference, 2006, pp. 320-328.
5. O. Chapelle, "Training a support vector machine in the primal," Neural computation, vol. 19, no. 5, pp. 1155-78, May 2007.
6. Pauwels Eric J., Onkar Ambekar.(2011). One Class Classification for Anomaly Detection: Support Vector Data
7. Peng, X., & Xu, D. (2012). Efficient support vector data descriptions for novelty detection. Neural Computing and Applications, 21(8), 2023-2032.
8. Robert P.W. Duin David M.J. Tax, "Support Vector Data Description," Machine Learning, vol. 54, pp. 45-66, 2004.
9. Wisaeng,K. (2013). "A Comparison of Different Classification Techniques for Bank Direct Marketing," International Journal of Soft Computing and Engineering (IJSCE), vol. 3, no. 4, pp. 116-119.
10. Y.-C. Liu, and Y.-J. Chen Y.-H. Liu, "Fast support vector data descriptions for novelty detection," IEEE Transactions on Neural Networks, vol. 21, no. 8, 2010.
11. R. Laureano and P. Cortez. S. Moro, "Using Data Mining For Bank Direct Marketing:An Application Of The Crisp-Dm Methodology," *Proceedings of the European Simulation and Modelling Conference - ESM'2011*, pp. 117-121, 2011.
12. Lingjun, L., Zhousuo, Z., & Zhengjia, H. (2003). Research of mechanical system fault diagnosis based on support vector data description. JOURNAL-XIAN JIAOTONG UNIVERSITY, 37(9), 910-913.
13. Muñoz-Marí, J., Bruzzone, L., & Camps-Valls, G. (2007). A support vector domain description approach to supervised classification of remote sensing images. Geoscience and Remote Sensing, IEEE Transactions on, 45(8), 2683-2692.
14. Pan, Y., Chen, J., & Guo, L. (2009). Robust bearing performance degradation assessment method based on improved wavelet packet–support vector data description. Mechanical Systems and Signal Processing, 23(3), 669-681.