GESTÃO & PRODUÇÃO

ISSN 0104-530X (Print)
ISSN 1806-9649 (Online)

THEMATIC SECTION: STATISTICAL PROCESS MONITORING

Monitoring the mean with least-squares support vector data description

O monitoramento da média com mínimos quadrados suporta a descrição de dados vetoriais

Edgard M. Maboudou-Tchao[1]

[1]Department of Statistics & Data Science, University of Central Florida, Orlando, FL, USA. E-mail: edgard.maboudou@ucf.edu.

How to cite: Maboudou-Tchao. E. M. (2021). Monitoring the mean with least-squares support vector data description. *Gestao & Producao*, 28(3), e19. http://doi.org/101590/1806-9649-2020v28e19.

Abstract: Multivariate control charts are essential tools in multivariate statistical process control (MSPC). “Shewhart-type” charts are control charts using rational subgroupings which are effective in the detection of large shifts. Recently, the one-class classification problem has attracted a lot of interest. Three methods are typically used to solve this type of classification problem. These methods include the *k*−center method, the nearest neighbor method, one-class support vector machine (OCSVM), and the support vector data description (SVDD). In industrial applications, like statistical process control (SPC), practitioners successfully used SVDD to detect anomalies or outliers in the process. In this paper, we reformulate the standard support vector data description and derive a least squares version of the method. This least-squares support vector data description (LS-SVDD) is used to design a control chart for monitoring the mean vector of processes. We compare the performance of the LS-SVDD chart with the SVDD and T^2 chart using out-of-control Average Run Length (ARL) as the performance metric. The experimental results indicate that the proposed control chart has very good performance.

Keywords: One-class classification; least squares support vector data description; least squares support vector machines; support vector data description; least squares one-class support vector machines.

Resumo: Gráficos de controle multivariados são ferramentas essenciais no controle estatístico multivariado de processos (MSPC). Os gráficos do “tipo Shewhart” são gráficos de controle usando subgrupos racionais que são eficazes na detecção de grandes mudanças. Recentemente, o problema de classificação de uma classe atraiu muito interesse. Normalmente, três métodos são usados para resolver esse tipo de problema de classificação. Esses métodos incluem o método k-center, o método do vizinho mais próximo, máquina de vetor de suporte de uma classe (OCSVM) e a descrição de dados de vetor de suporte (SVDD). Em aplicações industriais, como controle estatístico de processo (SPC), os profissionais usaram com sucesso o SVDD para detectar anomalias ou outliers no processo. Neste artigo, reformulamos a descrição de dados vetoriais de suporte padrão e derivamos uma versão de mínimos quadrados do método. Esta descrição de dados de vetor de suporte de mínimos quadrados (LS-SVDD) é usada para projetar um gráfico de controle para monitorar o vetor médio de processos. Comparamos o desempenho do gráfico LS-SVDD com o gráfico SVDD e T2 usando o comprimento médio de execução (ARL) fora de controle

Received Oct. 6, 2020 - Accepted Dec. 7, 2020
Financial support: None.

como a métrica de desempenho. Os resultados experimentais indicam que o gráfico de controle proposto tem um desempenho muito bom.

Palavras-chave: Classificação de uma classe; mínimos quadrados suportam descrição de dados vetoriais; mínimos quadrados suportam máquinas vetoriais; suportam descrição de dados vetoriais; mínimos quadrados suportam máquinas vetoriais.

1 Introduction

Due to recent advances in computing and availability of big data, the area of Statistical Process Monitoring (SPM) and Statistical Process Control (SPC) is facing a lot of challenges in dealing with these large data sets. The use of SPC methods in these new areas, the challenging difficulties of these new processes, and the increase of control improvement requirements has resulted in the improvement of classical SPC techniques. The use of classical SPC methods is based on very restrictive assumptions that are often not satisfied in practice. The well-known classical multivariate statistical process control (MSPC) that can monitor a multivariate process mean vector efficiently is Hotelling's T^2 chart. However, T^2 charts assume that the process observations follow a multivariate normal distribution. This assumption is unrealistic in modern industries and makes it impossible to apply the T^2 chart. The limitations of these classical SPC methods are overcome with the introduction of advanced computational intelligence tools, such as statistical learning and data science methods, into SPC. Work by Yeh et al. (2012) and Maboudou-Tchao & Diawara (2013), use a penalized likelihood ratio (PLR) chart based on the sparsity of the covariance matrix to monitor the covariance matrix for individual observations. Li et al. (2013), used the same idea of shrinking some of the components of the covariance matrix for monitoring the covariance matrix. Maboudou-Tchao & Agboto (2013), proposed the "Lasso chart" based on the graphical Lasso (glasso) of Friedman et al. (2008), to monitor singular covariance matrices.

These techniques allow management, monitoring of complex processes with high accuracy, and provide better process control results. Some of the most recent computational intelligence methods with promising results are Kernel methods. Kernel methods have the flexibility of handling various forms on different problems.

Recent control chart proposals are based on one-class classification algorithms. Sun & Tsung (2003), proposed kernel-distance-based charts (K charts) based on a Support Vector Data Description (SVDD) algorithm (Tax and Duin, 1999). Ning & Tsung (2013) discussed the optimal determination of the control boundary or control limit. In the same context, Weese et al. (2017), also discussed the selection of the bandwidth parameter for the k-chart.

Gani et al. (2011), explained how to apply the k-chart to industrial processes. Maboudou-Tchao et al. (2018), suggested a SVDD control chart using a Mahalanobis kernel. Kumar et al. (2006), and Camci et al. (2008), used another one-class SVM technique to construct robust K charts through normalized monitoring statistics. Liu & Wang (2014) proposed an adaptive kernel method for improving the sensitivity of the chart to small process shifts. In the context of one-class classification, Sukchotrat et al. (2009) proposed a K^2 chart based on the nearest neighbor's data description (kNNDD), while Kang & Kim (2011) suggested a K^2 chart based on the k-means. For one-class classification in higher order dimensions, Maboudou-Tchao (2018), suggested a control chart based on Support Matrix Data Description (SMDD) to monitor matrices. Maboudou-Tchao (2019), used tensor methods to monitor high-dimensional data. Maboudou-Tchao (2021), proposed support tensor data description (STDD). For more details on the use of statistical learning methods in process monitoring, the readers are referred to Weese et al. (2016).

Support vector machine (SVM) (Cortes & Vapnik, 1995), is a powerful method and has been proven to perform better than existing methods in many aspects. Least Squares support vector machine (LS-SVM) (Suykens & Vandewalle, 1999), is a variation of SVM and finds the separating hyperplane by solving a system of linear equations rather than a using quadratic programming solver used by SVM. This reduces computational costs significantly. Choi (2009), proposed least squares one-class SVM (LS-OCSVM). Maboudou-Tchao (2020), used LS-OCSVM to detect change-point in a process. Guo et al. (2017), proposed least squares support vector data description (LS-SVDD) and applied it to HRRP-based radar target recognition. In their paper, Guo et al. (2017), used quadratic programming to solve the dual problem. Because of the use of quadratic programming in solving LS-SVDD by Guo et al. (2017), their proposal has the same computational cost as SVDD (Tax & Duin, 1999) and does not achieve the main goals of least squares support vector methods.

The main contributions of this paper are as follows:

(i) Solve the LS-SVDD problem without using a quadratic programming solution as solving a constrained optimization criterion using quadratic programming (QP) leads to higher computational cost.

(ii) Derive the solution of the LS-SVDD problem in a closed form. This can efficiently reduce the computational cost and at the same time still makes LS-SVDD inherit the advantages of the conventional SVDD.

(iii) Propose a support vector control chart based on a closed form solution of the LS-SVDD like how Sun & Tsung (2003), proposed a support vector control chart based on SVDD formulation of Tax & Duin (1999). The LS-SVDD will ease the design of control charts using support vector methods and opens the doors to more advanced improvement of control charts using support vector methods.

In this paper, we propose another approach to solve the dual problem of the LS-SVDD without using quadratic programming solvers as suggested by Guo et al. (2017). We then construct a control chart based on LS-SVDD. This chart is a Shewhart-type and distribution free chart in the sense that no distributional properties are required. Consequently, it can be used to monitor any process. Note that all existing control charts based on support vector methods solve long and computationally hard quadratic programming problems. However, the proposed control chart is the first support vector control chart which doesn't use a computationally expensive quadratic programming solver. Instead, it uses a closed form solution and therefore simplifies the computational burden.

2 Support vector data description, SVDD

Support Vector Data Description is a method used for one-class classification. Consider a training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \subseteq \mathrm{X}$ where N is the number of vectors and $\mathrm{X} \subseteq \mathrm{R}^p$ is an original input space for which we want to have a "description". A description is a model that provides a closed boundary around the data, and this closed boundary is a sphere defined by its center $\mathbf{a}$ and radius $r > 0$. SVDD tries to find a sphere with minimum volume containing all, or most, of the vector $\mathbf{x}_j$ for $j = 1, 2, ..., N$. The sphere is determined by solving the following optimization problem:

$$
\begin{aligned}
&\underset{r,\boldsymbol{a},\boldsymbol{\xi}}{\text{minimize}} \quad r^2 + C\sum_{i=1}^{N}\xi_i,\\
&\text{subject to} \quad \varphi(\boldsymbol{x}_i) - \boldsymbol{a}^2 = \left(\varphi(\boldsymbol{x}_i) - \boldsymbol{a}\right)'\left(\varphi(\boldsymbol{x}_i) - \boldsymbol{a}\right) \le r^2 + \xi_i, i = 1,2,\ldots,N\\
&\xi_i \ge 0, i = 1,2,\ldots,N
\end{aligned}
\qquad (1)
$$

where **a** is the center of the sphere, *r* is the radius of the sphere, ξ_i are the slack variables, the parameter $C > 0$ is introduced to control the influence of the slack variables, and φ is a mapping that takes an input $\mathbf{x} \in \mathcal{X}$ and maps it to a feature space $\mathcal{F}$, where the feature space $\mathcal{F}$ is required to be a Hilbert space. This is a convex optimization problem. Equation 1 is also called the primal optimization problem. More discussions on how to solve convex optimization problems are found in Boyd & Vandenberghe (2004). According to Boyd & Vandenberghe (2004), the general approach to solve the constrained optimization problem (2.1) is to use the Lagrange dual problem.

The Lagrangian is:

$$
\begin{aligned}
L\left(r,\mathbf{a},\beta_i,\gamma_i,\xi_i\right) &= r^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\beta_i\left(r^2 + \xi_i - \left(\varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_i) - 2\varphi(\mathbf{x}_i)'\mathbf{a} + \mathbf{a}'\mathbf{a}\right)\right) - \sum_{i=1}^{N}\gamma_i\xi_i,\\
&= r^2\left(1 - \sum_{i=1}^{N}\beta_i\right) + \sum_{i=1}^{N}\xi_i\left(C - \beta_i - \gamma_i\right) + \sum_{i=1}^{N}\beta_i\left(\varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_i) - 2\varphi(\mathbf{x}_i)'\mathbf{a} + \mathbf{a}'\mathbf{a}\right)
\end{aligned}
\qquad (2)
$$

where β_i and γ_i are the Lagrange multipliers. To obtain the Lagrange dual problem, we need to determine the optimal *r*, *ξ*, and **a** in terms of the dual variables β and *γ*. We achieve this by differentiating the constraints with respect to the primal variables and solving.

$$
\begin{aligned}
&\frac{\partial L}{\partial r} = 0 \Rightarrow \sum_{i=1}^{N}\beta_i = 1\\
&\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - \beta_i - \gamma_i = 0,\\
&\frac{\partial L}{\partial \mathbf{a}} = 0 \Rightarrow a = \sum_{i=1}^{N}\beta_i\varphi(\mathbf{x}_i).
\end{aligned}
$$

By substituting these equations into (Equation 2), the following dual problem is solved instead (Equation 3):

$$
\begin{aligned}
&\max_{\boldsymbol{\beta}} \sum_{i=1}^{N}\beta_i\varphi(\boldsymbol{x}_i)'\varphi(\boldsymbol{x}_i) - \sum_{i,j=1}^{N}\beta_i\beta_j\varphi(\boldsymbol{x}_i)'\varphi(\boldsymbol{x}_j)\\
&\text{subject to} \quad \sum_{i=1}^{N}\beta_i = 1\\
&0 \le \beta_i \le C, i = 1,2,\ldots,N
\end{aligned}
\qquad (3)
$$

There is no need to compute the features φ(**x**) when one knows how to compute the products directly using a kernel. Consequently, the inner product $\varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j)$ is replaced by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j)$ in the high dimensional feature space. The dual problem becomes now:

$$\begin{aligned}&\max_{\boldsymbol{\beta}} \sum_{i=1}^{N}\beta_i k\left(\boldsymbol{x}_i,\boldsymbol{x}_i\right)-\sum_{i,j=1}^{N}\beta_i\beta_j k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\\ &\text{subject to}\quad \sum_{i=1}^{N}\beta_i=1\\ &0\le\beta_i\le C, i=1,2,\ldots,N\end{aligned} \tag{4}$$

This is a quadratic programming problem and can be solved efficiently using quadratic programming solver packages. Once the dual is maximized, the resulting Lagrange multipliers *β* can be used to calculate the center of the hypersphere **a**. The observations $\mathbf{x}_i$ corresponding to $\beta_i > 0$ are the "support vectors" (SV). The center **a** is a linear combination of these support vectors. The support vectors that correspond to vectors located on the sphere boundary, with $0 < \beta_i < C$, are the Boundary support vectors. The other support vectors, with $\beta_i = C$, are the Non-boundary support vectors. The squared radius r^2 is the squared distance from the center of the hypersphere **a** to the boundary support vectors $\mathbf{x}_s$ given by Equation 5:

$$r^2=\frac{1}{N_s}\sum_{s=1}^{N_s}\left(k\left(\boldsymbol{x}_s,\boldsymbol{x}_s\right)-2\sum_{i=1}^{N_s}\beta_i k\left(\boldsymbol{x}_s,\boldsymbol{x}_j\right)+\sum_{j,l}^{N_s}\beta_j\beta_l k\left(\boldsymbol{x}_j,\boldsymbol{x}_l\right)\right) \tag{5}$$

where N_s is the number of boundary support vectors.

To decide if a test vector **u** is a target, check if the kernel distance between **u** and the center **a** is less than the squared radius. That is:

$$d_{\boldsymbol{u}}=K\left(\boldsymbol{u},\boldsymbol{u}\right)-2\sum_{i=1}^{N}\beta_i K\left(\boldsymbol{u},\boldsymbol{x}_j\right)+\sum_{j,l=1}^{N}\beta_j\beta_l K\left(\boldsymbol{x}_j,\boldsymbol{x}_l\right)\le r^2 \tag{6}$$

If the condition (Equation 6) is not satisfied, then **u** is an outlier.

3 Least Squares Support vector data description, LS-SVDD

To derive least squares version of the support vector data description, we reformulate the SVDD described in (2.1) by using a quadratic error function and equality constraints. Consider a training set data $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\} \subseteq X$, where N is the number of vectors and $X \subseteq R^p$ is an original input space, for which we want to have a "description". By description, we wish to define a model which gives a closed boundary around the data. This closed boundary is a sphere defined by its center **c** and radius $R > 0$. The mathematical formulation of this model for finding the sphere is given by the following optimization problem:

$$\begin{aligned}&\underset{R,\boldsymbol{c},\boldsymbol{\xi}}{\text{minimize}}\quad R^2+C\sum_{j=1}^{N}\xi_j^2,\\ &\text{subject to}\quad \varphi\left(\boldsymbol{x}_j\right)-\boldsymbol{c}^2=\left(\varphi\left(\boldsymbol{x}_j\right)-\boldsymbol{c}\right)'\left(\varphi\left(\boldsymbol{x}_j\right)-\boldsymbol{c}\right)=R^2+\xi_j, j=1,2,\ldots,N\\ &\xi_j\ge 0, j=1,2,\ldots,N\end{aligned} \tag{7}$$

where *R* is the radius of the sphere, the parameter *C* > 0 is introduced to control the influence of the slack variables, and *φ* is a function mapping data to a higher dimensional Hilbert space. Here, the constraints for the slack variables, $\xi_j \geq 0$ in (2.1), are no longer needed. Instead, one can think of the variable ξ_j as an error realized by a training vector $\mathbf{x}_j$ with respect to the hypersphere.

Equation 7 is an example of the general equality-constrained optimization problems. The optimality conditions for an equality constrained problem are obtained by the Karush-Kuhn-Tucker (KKT) conditions. More discussions on how to solve general equality-constrained optimization problems are found in Boyd & Vandenberghe (2004).

Consequently, the typical approach for solving this constrained optimization problem (3.1) is to use the Lagrange dual problem. After introducing Lagrange multipliers α_j, $j = 1,...N$, we get the following Lagrangian function:

$$L\left(R,\boldsymbol{c},\boldsymbol{\alpha},\boldsymbol{\xi}\right) = R^2 + C\sum_{j=1}^{N}\xi_j^2 - \sum_{j=1}^{N}\alpha_j\left(R^2 + \xi_j - \varphi\left(\mathbf{x}_j\right) - \mathbf{c}^2\right) \tag{8}$$

where α_j, $j = 1,...N$ are the Lagrange multipliers, which can be either positive or negative due to the equality constraints.

The Karush-Kuhn-Tucker (KKT) equations of optimality for *L* are achieved by differentiating the Lagrangian with respect to the variables *R*, **c**, and *ξ*. Consequently, differentiating (Equation 8) with respect to these variables and setting the derivatives to zero give:

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_{j=1}^{N}\alpha_j = 1 \tag{9}$$

$$\frac{\partial L}{\partial \xi_j} = 0 \Rightarrow \alpha_j = C\xi_j \tag{10}$$

$$\frac{\partial L}{\partial \boldsymbol{c}} = 0 \Rightarrow \boldsymbol{c} = \sum_{j=1}^{N}\alpha_j\varphi\left(\mathbf{x}_j\right) \tag{11}$$

From Equations 9, 10 and 11, one key observation can be made. From (Equation 10), the support values α_j are proportional to the errors ξ_j at the data points in the LS-SVDD case, while in the SVDD case, many support values are typically equal to zero.

We now insert these values back into the original Lagrangian (Equation 8) to obtain the following dual problem:

$$L\left(\alpha\right) = \sum_{j=1}^{N}\alpha_j\varphi\left(\mathbf{x}_j\right)'\varphi\left(\mathbf{x}_j\right) - \sum_{i,j=1}^{N}\alpha_i\alpha_j\left(\varphi\left(\mathbf{x}_j\right)'\varphi\left(\mathbf{x}_j\right) + \frac{1}{2C}\delta_{ij}\right) \tag{12}$$

where δ_{ij} represents the Kronecker-delta. This objective function is subject to the constraint $\sum_{j=1}^{N}\alpha_j = 1$. Guo et al. (2017), suggest solving the following dual problem (Equation 12) by using a quadratic programming solver. Using quadratic programming complicates things and increases computational costs. Instead, the next section will propose a new way to get to the same solution of Guo et al. (2017), without using a quadratic programming solver.

3.1 Solution of the dual problem

The dual problem is:

$$\begin{aligned}&\max_{\boldsymbol{\alpha}} \sum_{j=1}^{N}\alpha_j\varphi(\boldsymbol{x}_j)'\,\varphi(\boldsymbol{x}_j)-\sum_{i,j=1}^{N}\alpha_i\alpha_j\left(\varphi(\boldsymbol{x}_i)'\,\varphi(\boldsymbol{x}_j)+\frac{1}{2C}\delta_{ij}\right)\\&\text{subject to} \sum_{j=1}^{N}\alpha_j=1\end{aligned} \tag{13}$$

This dual problem (Equation 13) involves only a single equality constraint, unlike SVDD where there are multiple inequality constraints. Therefore, this dual problem is no longer a quadratic programming problem, as in the SVDD case, but a quadratic problem and has an analytic solution. The interested reader can refer to Boyd & Vandenberghe (2004), for more details on how to solve this optimization problem.

To solve this dual problem, we will need to first obtain its Lagrangian. The Lagrangian of the dual problem is

$$L(\boldsymbol{\alpha})=\sum_{j=1}^{N}\alpha_j\varphi(\mathbf{x}_j)'\,\varphi(\mathbf{x}_j)-\sum_{i,j=1}^{N}\alpha_i\alpha_j\left(\varphi(\mathbf{x}_i)'\,\varphi(\mathbf{x}_j)+\frac{1}{2C}\delta_{ij}\right)+\gamma\left(\sum_{j=1}^{N}\alpha_j-1\right) \tag{14}$$

Using the kernel trick by letting k($\mathbf{x}_i$,$\mathbf{x}_j$) = φ($\mathbf{x}_i$)'φ($\mathbf{x}_j$), (Equation 14) becomes:

$$L(\boldsymbol{\alpha})=\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_j,\mathbf{x}_j)-\sum_{i,j=1}^{N}\alpha_i\alpha_j\left(k(\mathbf{x}_i,\mathbf{x}_j)+\frac{1}{2C}\delta_{ij}\right)+\gamma\left(\sum_{j=1}^{N}\alpha_j-1\right) \tag{15}$$

In matrix notation, let **K** denotes the Gram matrix with entries $K_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$, $\mathbf{I}_N$ denotes the identity matrix, $\boldsymbol{H}=\boldsymbol{K}+\frac{1}{2C}\boldsymbol{I}_N$, $e=(1,1,\ldots,1)'$ and $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_N)'$. Then we can write (Equation 15) as (Equation 16):

$$L(\boldsymbol{\alpha})=-\boldsymbol{\alpha}'\mathbf{H}\boldsymbol{\alpha}+\mathbf{k}'\boldsymbol{\alpha}+\gamma(\mathbf{e}'\boldsymbol{\alpha}-1) \tag{16}$$

where **k** denotes a vector with entries $k_j = k(\mathbf{x}_j,\mathbf{x}_j)$, $j = 1,\ldots,N$.

This is an unconstrained optimization over the variable *α* and therefore can be solved using any algorithms for unconstrained convex optimization.

Taking the derivative of *L*(*α*) with respect to *α* and setting it to **0** gives

$$\frac{\partial L(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}=-2\mathbf{H}\boldsymbol{\alpha}+\mathbf{k}+\gamma\mathbf{e}=0$$

Since **H** is always invertible, for an appropriate choice of *C* > 0, we get:

$$\boldsymbol{\alpha} = \frac{1}{2}\mathbf{H}^{-1}\left(\mathbf{k} + \gamma\mathbf{e}\right) \tag{17}$$

Next, we need to find *γ*. Consequently, we take the derivative of *L*(*α*) with respect to *γ* and set it to 0, after replacing *α* with (Equation 17). This yields:

$$\frac{\partial \mathrm{L}\left(\boldsymbol{\alpha}\right)}{\partial \gamma} = \ \mathbf{e}'\alpha - 1 = 0 \ \Rightarrow \frac{1}{2}\ \mathbf{e}'\mathbf{H}^{-1}\left(\mathbf{k} + \gamma\mathbf{e}\right) = 1$$

We can then compute *γ* as:

$$\gamma = \frac{2 - \ \mathbf{e}'\mathbf{H}^{-1}\ \mathbf{k}}{\mathbf{e}'\mathbf{H}^{-1}\ \mathbf{e}} \tag{18}$$

Replacing *γ* in (Equation 17) with (Equation 18) gives the analytic solution of the dual problem for LS-SVDD as,

$$\boldsymbol{\alpha} = \frac{1}{2}\mathbf{H}^{-1}\left(\mathbf{k} + \frac{2 - \ \mathbf{e}'\mathbf{H}^{-1}\ \mathbf{k}}{\mathbf{e}'\mathbf{H}^{-1}\ \mathbf{e}}\mathbf{e}\right) \tag{19}$$

From (Equation 10), it follows immediately that the errors realized by the training vectors with respect to the hypersphere, *ξ*, can be computed by (Equation 20):

$$\xi = \frac{\boldsymbol{\alpha}}{\mathrm{C}} = \frac{1}{2\mathrm{C}}\mathbf{H}^{-1}\left(\mathbf{k} + \frac{2 - \ \mathbf{e}'\mathbf{H}^{-1}\ \mathbf{k}}{\mathbf{e}'\mathbf{H}^{-1}\ \mathbf{e}}\mathbf{e}\right) \tag{20}$$

Once the analytic solution for *α* is obtained, the radius *R* is found as follows (Equation 21):

$$\mathrm{R}^2 = \frac{1}{\mathrm{N}}\sum_{s=1}^{\mathrm{N}}\left(\mathrm{K}\left(\mathbf{x}_s, \mathbf{x}_s\right) - 2\sum_{j=1}^{\mathrm{N}}\alpha_j \mathrm{K}\left(\mathbf{x}_s,\ \mathbf{x}_j\right) + \sum_{j,l}^{\mathrm{N}}\alpha_j\alpha_l \mathrm{K}\left(\mathbf{x}_j, \mathbf{x}_l\right)\right) \tag{21}$$

For classification, a test vector **z** is in the target class if the kernel distance between **z** to the sphere center, **c**, is smaller or equal to the radius, i.e.

$$\mathrm{d}_z = \mathrm{K}\left(z, z\right) - 2\sum_{j=1}^{\mathrm{N}}\alpha_j \mathrm{K}\left(z, \boldsymbol{x}_j\right) + \sum_{j,l=1}^{\mathrm{N}}\alpha_j\alpha_l \mathrm{K}\left(\boldsymbol{x}_j, \boldsymbol{x}_l\right) \le \mathrm{R}^2 \tag{22}$$

If the condition (Equation 22) does not hold, then **z** is declared to be an outlier.

4 LS-SVDD control charts

4.1 Issue with SVDD

One can think of SVDD as a special case of SVM and therefore will inherit all the issues with SVM. The major drawback of the SVM (or the SVDD) algorithm is its higher computational burden for the required constrained optimization programming. This disadvantage has been overcome by least squares support vector machines (LS-SVM), which solves linear equations instead of a quadratic programming problem, (Wang & Hu, 2005). This becomes a bigger issue when dealing with large-scale problems, (Liu et al., 2016; Qiu et al., 2016). With its advantage of producing sparse solutions, it is a huge challenge to train these high dimensional data using quadratic programming (Rodriguez-Lujan et al., 2010). Moreover, SVM (or SVDD) performance suffers as we increase the number of dimensions. This is due to the constrained optimization problem that backs SVMs (or SVDDs). SVM (or SVDD) will likely struggle with a dataset where the number of features is much larger than the number of observations (Guyon et al., 2002). This, again, can be understood by looking at the constrained optimization problem. These disadvantages can be easily overcome by using least squares version of SVM (or SVDD) instead. In LS-SVM or LS-SVDD, one works with equality instead of inequality constraints and a sum squared error (SSE) cost function. This reformulation greatly simplifies the problem in such a way that the solution is characterized by a linear system, more precisely a Karush-Kuhn-Tucker (KKT) system. Therefore, it simplifies greatly the training and design step.

4.2 LS-SVDD chart

Consider a phase I sample **X** = {**x**$_1$,**x**$_2$,⋯, **x**$_N$} of *p* component vectors assumed to be in control is available. The LS-SVDD problem solves problem (Equation 7) using (Equation 19). For each new vector **z**$_i$, compute the chart statistic d_i, the kernel distance from **z**$_i$ to the center **c** as

$$\mathrm{d_i} = \mathrm{K}\left(z_i, z_i\right) - 2\sum_{\mathrm{j=1}}^{\mathrm{N}} \alpha_{\mathrm{j}} \mathrm{K}\left(z_i, \boldsymbol{x}_{\mathrm{j}}\right) + \sum_{\mathrm{j,l=1}}^{\mathrm{N}} \alpha_{\mathrm{j}} \alpha_{\mathrm{l}} \mathrm{K}\left(\boldsymbol{x}_{\mathrm{j}}, \boldsymbol{x}_{\mathrm{l}}\right) \tag{23}$$

To design the chart, the radius threshold R^2 is needed. However, in the context of process control, the upper limit threshold *h* based on the desired in-control average run length (IC ARL) is more suitable. The control chart involves plotting d_i against *i* and detecting a shift when $d_i > h$. The control limit *h* is selected to satisfy a specified IC ARL. Getting the appropriate value of *h* to achieve a specific IC ARL is obtained by using a bootstrap simulation. This chart will be called the LS-SVDD chart.

4.3 Control limit *h*

Note that one can use the radius R^2 for monitoring purposes. However, if the practitioner is more interested about controlling for the type I error, the upper limit threshold *h* based on the desired IC ARL is more suitable. The control limit *h* defines the run lengths. Values of *h* are obtained by using a bootstrap method. The bootstrap procedure to calculate the control limits is summarized as follows:

1. Compute the *d* statistics with *N* observations from the training sample using (Equation 23).
2. Generate *B* independent bootstrap samples from phase I. Use *B* = 5000.
3. Let $d_1^{(i)}, d_2^{(i)}, \ldots, d_N^{(i)}$ be a set of *N d* statistics from the *i*th bootstrap sample, *i* = 1,...,*B*.
4. In each of the *B* bootstrap samples, let $\alpha = \frac{1}{\text{ICARL}}$ be the false alarm corresponding to the IC ARL desired, determine the 100 ×(1 − *α*)th percentile value.
5. Obtain the control limit by taking an average of *B* 100 ×(1−*α*)th percentile values, *h*.
6. Use the established control limit, *h*, to monitor a new observation.

4.4 Algorithm Summary

In short, the LS-SVDD control chart using least squares support vector methods consists of the following steps:

1. Assume that an in-control phase I sample is available. Determine the tuning parameters *C* and *σ*. We suggest an approach similar to the one described in Choi (2009).
2. Use the bootstrap to obtain the appropriate control limit desired, *h*.
3. Apply LS-SVDD to the phase I sample to obtain the solutions to the parameter α_i, *i* = 1, 2,...,*N* by using (Equation 19). Note that no quadratic programming solvers are needed for this step as with the standard SVDD control chart.
4. For each new observation $\mathbf{z}_i$, compute the chart statistic,

$$d_i = K(\mathbf{z_i},\mathbf{z_i}) - 2\sum_{j=1}^{N}\alpha_j K(\mathbf{z_i},\mathbf{x_j}) + \sum_{j,l=1}^{N}\alpha_j\alpha_l K(\mathbf{x_j},\mathbf{x_l})$$

5. Detect an out-of-control behavior when $d_i > h$.

These steps can be summarized easily in the flowchart shown below.
Control Chart Design 1: LS-SVDD
Inputs: training data vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$

1. Use the phase I sample to determine the tuning parameters *C* and *σ*.
2. Use the phase I sample to get:

$$\boldsymbol{\alpha} = \frac{1}{2}\mathbf{H}^{-1}\left(\mathbf{k} + \frac{2- \ \mathbf{e}'\mathbf{H}^{-1}\ \mathbf{k}}{\mathbf{e}'\mathbf{H}^{-1}\ \mathbf{e}}\mathbf{e}\right).$$

3. Obtain the appropriate control limit *h*.
4. Compute for a new observation $\mathbf{z}_i$

$$d_i = K(\mathbf{z_i},\mathbf{z_i}) - 2\sum_{j=1}^{N}\alpha_j K(\mathbf{z_i},\mathbf{x_j}) + \sum_{j,l=1}^{N}\alpha_j\alpha_l K(\mathbf{x_j},\mathbf{x_l})$$

5. Signal a change if $d_i > h$.

4.5 LS-SVDD boundary

We investigate the LS-SVDD control chart's boundary in two-dimensional cases. The data set used here was discussed in Montgomery (2001). The data set is from a chemical process with four variables. The data set has two classes labeled as "Original" and "New" and was initially analyzed with principal component analysis. There are 20 items in "Original" and 10 in "New". Our discussion will focus on the two principal components Z_1 and Z_2 available from the dataset. Figure 1 shows several control boundary and trajectory plots for the principal components Z_1 and Z_2. The black 'dot' points represent the 20 training points labeled "original" while the blue 'triangle' points represent the 10 test points labeled "new". Similarly, to the SVDD chart case, the control boundary for LS-SVDD chart is also not always an ellipse. The shapes adapt to real data and become irregular as in the SVDD chart case. These irregular shapes can detect shifts that occur in the data that elliptical shapes may not. Figure 1 shows how the LS-SVDD boundary shapes change from an elliptical shape to adapt to the shape of the training data by changing the values of the tuning parameters C and σ.

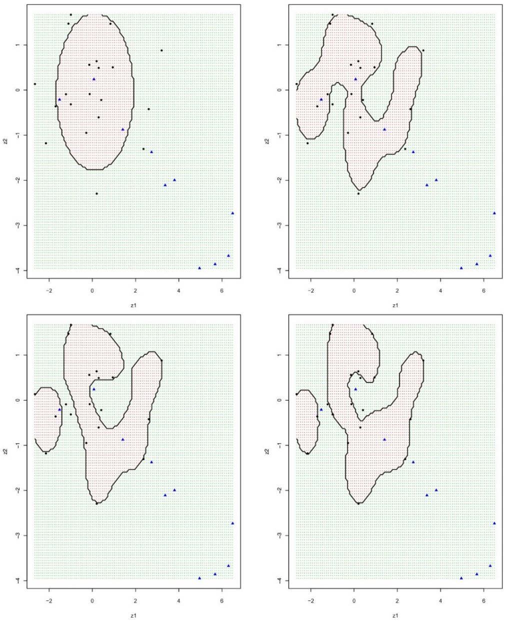

Figure 1. Boundary Shapes for LS-SVDD, (top, left) C = 0.5 and σ = 10.5, (top right) C = 10 and σ = 1.5, (bottom, left) C = 50 and σ = 1.5, (bottom, right) C = 500 and σ = 1.5.

5 Performance study settings

A typical measure to assess the performance of a control chart is the out-of-control average run length (OOC ARL), i.e., average run length (ARL) following an out-of-control shift. We compare the performance of the LS-SVDD chart to its counterpart, the SVDD chart, Sun & Tsung (2003), and the Hotelling T^2 chart.

The Hotelling T^2 chart is well known control chart used for monitoring process mean vectors. To design the Hotelling's T^2 control chart, for each individual observation j, we calculate (Equation 24)

$$\mathrm{T}_{\mathrm{j}}^{2} = \left(\boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}}\right)' \boldsymbol{S}^{-1} \left(\boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}}\right), \tag{24}$$

Where $\bar{\boldsymbol{x}} = \mathrm{N}^{-1}\sum_{\mathrm{j}=1}^{\mathrm{N}} \boldsymbol{x}_{\mathrm{j}}$ is the phase I sample mean and $\boldsymbol{S} = \left(\mathrm{N}-1\right)^{-1} \sum_{\mathrm{j}=1}^{\mathrm{N}} (\boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}})(\boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}})'$ is the phase I sample covariance matrix.

We choose to compare the LS-SVDD to the SVDD because the SVDD is the other control chart based on support vector methods. Also, since the design of the LS-SVDD chart is simple compared to the design of the SVDD chart, if the two charts have similar or close performances, one can just use the LS-SVDD instead. In that case, the LS-SVDD chart should be used to design online control chart using support vector methods rather than SVDD chart. The incremental and decremental updating steps, needed for an online scheme, for SVDD are very tough problems. On the other hand, the incremental and decremental updating steps are very easy for LS-SVDD. The Hotelling T^2 chart is a parametric control chart that is optimum for multivariate normal data whereas the LS-SVDD and SVDD are both nonparametric methods that can be used for any distribution.

The phase I sample is obtained from a multivariate normal distribution, multivariate t distribution, and multivariate lognormal. For these distributions, we set the mean vector to **0** and covariance matrix $\mathbf{\Sigma}_0 = \mathbf{I}_p$, the identity matrix.

The in-control average run length is set to 200. The performance comparison requires adjusting the approaches to have comparable in-control behavior, and then selecting informative out-of-control settings to evaluate.

We used an approach similar to the one described in Choi (2009) to find the parameters *C* and *σ* in the RBF kernel function. As for dimension, we investigate p = 5 and p = 10. A total of N = 100 in-control phase I observations were generated from the three distributions of interest.

5.1 Out-of-control settings

In the first scenario, we used the multivariate normal distribution and changed the mean from its in-control value of **0** to a vector having *δ* in its first component while the other elements remain unchanged. In the second scenario, we investigated the multivariate t-distribution with 3 degrees of freedom. For the third case, we used the multivariate lognormal.

5.1.1 Performance for multivariate normal

With the 100 phase I observations generated from the multivariate normal, the control limit for an in-control ARL of 200 was computed. Note that when the process shifts from

N(μ, $\mathbf{\Sigma}_0$) to N(μ$_1$, $\mathbf{\Sigma}_0$), the chart behavior is determined by the noncentrality parameter $\delta^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu})' \Sigma_0^{-1} (\boldsymbol{\mu}_1 - \mu)$. Consequently, it follows that the performance after any step change in the mean can be modeled by changing a single component of the mean vector as follows: Set the in-control mean vector to **0** and the covariance matrix to $\mathbf{\Sigma}_0$. Thereafter, by adding a shift *δ* to the first component of each vector we can explore all possible step changes in the mean vector.

We investigate the performance by varying *δ* from 0 to 1, simulating a minimum of 20,000 independent series at each *δ* value. The results are shown in Table 1. The table displays the comparison between the LS-SVDD, SVDD, and Hotelling's T^2 charts for p = 5 and 10.

Table 1. Performance comparison of the charts for Multivariate Normal – ARL (standard errors are shown inside the parentheses), (left) p = 5, (right) p = 10.

Charts, *p* = 5				**Charts, *p* = 10**			
δ	**LS-SVDD**	**SVDD**	***T2***	***δ***	**LS-SVDD**	**SVDD**	***T2***
0	200.3 (1.651)	200.1 (1.642)	202.8 (1.61)	0	201.5 (1.65)	199.6 (1.62)	201.8 (1.651)
0.1	194 (1.556)	192.1 (1.584)	201.8 (1.671)	0.1	194.8 (1.608)	195 (1.591)	199.7 (1.62)
0.2	180.2 (1.475)	180.5 (1.465)	200.5 (1.646)	0.2	190.3 (1.533)	191.7 (1.594)	195.2 (1.567)
0.3	166.5 (1.358)	166.2 (1.337)	198.9 (1.634)	0.3	176.9 (1.439)	180.2 (1.45)	187.8 (1.531)
0.4	149.9 (1.216)	151.6 (1.232)	190.4 (1.54)	0.4	162 (1.309)	168.9 (1.362)	184.3 (1.507)
0.5	131.7 (1.069)	133.2 (1.07)	173.1 (1.418)	0.5	153 (1.231)	156.3 (1.262)	174.8 (1.436)
0.6	114.9 (0.9353)	117.6 (0.9608)	156.5 (1.27)	0.6	140.3 (1.142)	143.6 (1.173)	161.4 (1.305)
0.7	99.21 (0.8095)	104.8 (0.8424)	140.7 (1.128)	0.7	124.7 (1.015)	130.6 (1.063)	149.1 (1.207)
0.8	84.3 (0.6843)	89.38 (0.7232)	123.5 (1.01)	0.8	111.5 (0.8993)	116.5 (0.9506)	139.3 (1.14)
0.9	71.68 (0.5658)	75.57 (0.616)	106.2 (0.8754)	0.9	97.06 (0.7921)	101.5 (0.8192)	125.42 (1.0 16)
1	60.53 (0.4889)	63.36 (0.5102)	93.19 (0.7584)	1	88.12 (0.7205)	88.7 (0.7163)	112.1 (0.9194)

In terms of performance, LS-SVDD and SVDD charts have similar performance with a slight advantage for LS-SVDD and both charts perform better than Hotelling T^2 chart with individual observations.

5.1.2 Performance for multivariate t

A random vector **X** from a multivariate Student's t distribution with mean vector μ, scale matrix **Σ**, and degrees of freedom v has expected value $E(\mathbf{X})$ = μ and covariance matrix $\mathrm{Cov}(X) = \frac{v}{v-2}\Sigma$. Consequently, we set the in-control mean vector to μ = **0** and the scale matrix $\Sigma = \frac{v-2}{v}\Sigma_0$. This yields the covariance matrix of the simulated random vector to be $\mathbf{\Sigma}_0$. We evaluate the performance of the chart by adding a shift δ to the first component of each vector.

Using 100 phase I observations simulated from the multivariate Student's t with 3 degrees of freedom, we investigate the performance of the charts with δ varying from 0 to 10, simulating a minimum of 20,000 independent series at each δ value. The resulting ARL values are shown in Table 2. For this scenario, LS-SVDD performed better than SVDD but both were outperformed by T^2.

5.1.3 Performance for multivariate lognormal

Next, we simulate 100 phase I observations from the multivariate lognormal with mean vector **0** and covariance matrix $\mathbf{\Sigma}_0 = \mathbf{I}_p$. The in-control ARL was set at 200. A shift δ was introduced to the first component of the mean vector. We explore the performance of the chart by varying δ from 0 to 1, simulating a minimum of 20,000 independent series at each *δ* value. The resulting out-of-control ARL values are shown in Table 3. The table displays the comparison between the LS-SVDD, SVDD, and Hotelling's T^2 charts for $p = 5$ and 10.

Table 2. Performance comparison of the charts for Multivariate *t* with 3 degrees of freedom – ARL (standard errors are shown inside the parentheses), (left) $p = 5$, (right) $p = 10$.

Charts, *p* = 5				**Charts, *p* = 10**			
δ	**LS-SVDD**	**SVDD**	***T2***	***δ***	**LS-SVDD**	**SVDD**	***T2***
0	199.1 (1.612)	200.1(1.675)	198.5 (1.611)	0	199.3 (1.626)	201.9 (1.658)	199 (1.607)
1	200.1 (1.623)	201.1 (1.697)	195.5 (1.589)	1	199.1 (1.622)	201.1 (1.635)	201.9 (1.65)
2	197.8 (1.614)	200.8 (1.67)	193.3 (1.556)	2	196.8 (1.596)	200.6 (1.62)	195.4 (1.579)
3	189.1 (1.544)	200.6 (1.653)	184.9 (1.48)	3	194 (1.575)	194.9 (1.575)	189.9 (1.558)
4	183.5 (1.512)	188.3 (1.531)	176.1 (1.443)	4	186 (1.496)	191.2 (1.573)	180 (1.468)
5	169.5 (1.377)	181.1 (1.464)	162.7 (1.306)	5	179.7 (1.445)	182.1 (1.484)	175.2 (1.415)
6	159.2 (1.291)	169.3 (1.361)	150.7 (1.226)	6	170.2 (1.384)	178.2 (1.434)	162.3 (1.313)
7	140.2 (1.139)	151.8 (1.238)	133.7 (1.101)	7	160 (1.318)	168.6 (1.373)	148.6 (1.209)
8	122.2 (0.9952)	134.3 (1.087)	115.2 (0.933)	8	151.9 (1.221)	157.1 (1.277)	134.9 (1.089)
9	100.9 (0.8227)	111.8 (0.9191)	98.63 (0.8058)	9	136.8 (1.102)	146.9 (1.181)	118.1 (0.9529)
10	80.80 (0.6504)	92.66 (0.7525)	80.08 (0.6485)	10	121.7 (0.9961)	131.8 (1.053)	102.4 (0.831)

Table 3. Performance comparison of the charts for Multivariate Lognormal – ARL (standard errors are shown inside the parentheses), (left) $p = 5$, (right) $p = 10$.

Charts, *p* = 5				**Charts, *p* = 10**			
δ	**LS-SVDD**	**SVDD**	***T2***	***δ***	**LS-SVDD**	**SVDD**	***T2***
0	199.1 (1.613)	201(1.646)	202.8 (1.641)	0	199.9 (1.609)	201.4 (1.654)	199 (1.607)
0.1	178.4 (1.469)	175.5 (1.414)	176.2 (1.439)	0.1	187.7 (1.535)	187.9 (1.523)	188.1 (1.547)
0.2	156.6 (1.281)	153.5 (1.248)	151.9 (1.236)	0.2	170.8 (1.393)	169.3 (1.39)	171.3 (1.396)
0.3	136.7 (1.098)	129 (1.042)	133.2 (1.088)	0.3	155.9 (1.271)	154.3 (1.255)	154.6 (1.255)
0.4	115.8 (0.9386)	110.1 (0.8888)	111.5 (0.914)	0.4	140.1 (1.142)	135.5 (1.109)	136 (1.113)
0.5	97.72 (0.7888)	90.84 (0.745)	94.56 (762)	0.5	125.7 (1.023)	119.9 (0.9781)	120.7 (0.9829)
0.6	83.87 (0.6676)	74.84 (0.6061)	78.61 (0.6332)	0.6	112.2 (0.9167)	106 (0.8701)	106.7 (0.8655)
0.7	68.02 (0.5489)	62.83 (0.5064)	66.59 (0.5359)	0.7	95.55 (0.7829)	90.06 (0.7174)	93.75 (0.756)
0.8	56.21 (0.4531)	50.71 (0.4079)	55.43 (0.4417)	0.8	83.29 (0.6778)	77.24 (0.6326)	80.97 (0.6652)
0.9	47.06 (0.38)	41.86 (0.3417)	45.33 (0.3698)	0.9	72.73 (0.5759)	65.71 (0.5379)	69.99 (0.5665)
1	39.51 (0.3247)	34.33 (0.2742)	37.97 (0.305)	1	61.22 (0.4953)	55.46 (0.4569)	59.74 (0.485)

From Table 3, SVDD chart has the best performance. LS-SVDD and T^2 charts seem to have similar performance with a slight advantage to T^2 chart but both charts were outperformed by SVDD chart.

5.1.4 Impact of the sample size

In this section, a study is conducted to evaluate the impact of the sample size n on the performance of the control chart. We simulated 100 observations from the multivariate normal with mean vector **0** and covariance matrix $\mathbf{\Sigma}_0 = \mathbf{I}_p$. Out of these 100 observations, we randomly select N observations to form our phase I sample, which is used for the design of the LS-SVDD chart.

To evaluate the performance, we introduce a change of δ in the first component while the other components remain unchanged. We investigate the performance of the chart by varying δ from 0 to 1, simulating a minimum of 20,000 independent series at each δ value. We considered the cases p = 5 and 10, and N = 30, 50, 75, 90, and 100. Again, we set the in-control average run length to 200. Tables 4 and 5 show the impact of the sample size on the performance of the chart.

From the results, it is very hard to find a clear pattern. However, as N increases, the performances seem to stabilize. This can be seen for N = 75, 90, and 100 where the ARLs are close to each other. In the SVDD chart case, the number of observations required to establish an in-control reference sample is unclear, and little advice is given as to how to obtain this reference sample. The same can be said also for the LS-SVDD chart. Based on our simulations, increasing the number of observations in the phase I sample seems to stabilize and improve the performance of the charts considered.

Table 4. Impact of the Sample size – ARL (standard errors are shown inside the parentheses), p = 5.

		Sample size, p = 5			
δ	***N* = 30**	***N* = 50**	***N* = 75**	***N* = 90**	***N* = 100**
0	200.5 (1.629)	201.5 (1.656)	201.3 (1.63)	199.3 (1.619)	200.3 (1.651)
0.1	196.7 (1.594)	188.8 (1.535)	191.6 (1.549)	189.5 (1.549)	194 (1.556)
0.2	187.4 (1.524)	172.7 (1.416)	179.2 (1.453)	179.4 (1.458)	180.2 (1.475)
0.3	175.6 (1.434)	157.6 (1.271)	162.7 (1.34)	165.9 (1.353)	166.5 (1.358)
0.4	158.1 (1.297)	140.6 (1.14)	147.5 (1.192)	145.1 (1.189)	149.9 (1.216)
0.5	143.1 (1.16)	120.7 (0.9761)	127.9 (1.037)	131 (1.058)	131.7 (1.069)
0.6	126.1 (1.032)	103.3 (0.8459)	113 (0.9194)	114.4 (0.9322)	114.9 (0.9353)
0.7	109.8 (0.9006)	89.42 (0.7207)	95.17 (0.7678)	98.41 (0.8046)	99.21 (0.8095)
0.8	94.34 (0.7618)	74.46 (0.6041)	82.86 (0.6567)	83.93 (0.6867)	84.3 (0.6843)
0.9	80.76 (0.6551)	62.63 (0.5052)	68.3 (0.5513)	70.43 (0.5595)	71.68 (0.5658)
1	67.77 (0.5489)	52.99 (0.4264)	58.97 (0.478)	60.34 (0.4843)	60.53 (0.4889)

Table 5. Impact of the Sample size – ARL (standard errors are shown inside the parentheses), p = 10.

		Charts, p = 10			
δ	***N* = 30**	***N* = 50**	***N* = 75**	***N* = 90**	***N* = 100**
0	198.6 (1.663)	202.7 (1.654)	202.5 (1.643)	199 (1.637)	199.6 (1.62)
0.1	211 (1.709)	204.3 (1.654)	199.6 (1.632)	201.6 (1.642)	195 (1.591)
0.2	216.5 (1.776)	204 (1.678)	192.2 (1.566)	192.5 (1.58)	191.7 (1.594)

Table 5. Continued…

Charts, *p* = 10					
δ	***N* = 30**	***N* = 50**	***N* = 75**	***N* = 90**	***N* = 100**
0.3	221.4 (1.796)	202.2 (1.65)	183.3 (1.499)	185.3 (1.507)	180.2 (1.45)
0.4	218 (1.784)	191.8 (1.558)	169.5 (1.389)	174.9 (1.412)	168.9 (1.362)
0.5	215.7 (1.731)	182.7 (1.491)	157.8 (1.295)	160.9 (1.322)	156.3 (1.262)
0.6	211.3 (1.735)	171.4 (1.381)	142.8 (1.153)	144.4 (1.169)	143.6 (1.173)
0.7	199.2 (1.653)	155.9 (1.205)	128.1 (1.045)	130.1 (1.065)	130.6 (1.063)
0.8	184.2 (1.512)	139.8 (1.148)	116.1 (0.9488)	118.2 (0.9557)	116.5 (0.9506)
0.9	170.1 (1.38)	128.6 (1.034)	100.4 (0.8142)	105.3 (0.8544)	101.5 (0.8192)
1	158.1 (1.29)	112.7 (0.9111)	86.81 (0.7095)	92.3 (0.7377)	88.7 (0.7163)

5.1.5 Summary

In this section, we summarize the results obtained from the simulations.

- For multivariate normal distributions, LS-SVDD slightly performs better than SVDD and both charts have better performances than Hotelling's T^2.
- For multivariate *t* distributions, Hotelling's T^2 performs better than LS-SVDD and SVDD. LS-SVDD has a better performance than SVDD.
- For multivariate lognormal distributions, SVDD has better performance than LSSVDD and T^2.
- In terms of the impact of the phase I sample size, it is not straightforward to find a pattern. However, increasing the phase I sample size seems to improve the performance of LS-SVDD.

6 Example

We evaluate the performance of LS-SVDD with an application to the 'breast cancer dataset'. The dataset was created by Dr. William H. Wolberg, physician at the University of Wisconsin. The breast cancer dataset is publicly available and can be found in UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/breast+cancer.

The goal is to detect changes in the mean vector of the process. There are two target classes, labeled as 2 or 4 in the dataset. The dataset consists of 9 features measured on 683 patients: 444 diagnosed with benign cancer (target class 2) and 239 diagnosed with malignant cancer (target class 4).

We use the first 80 observations of the target class 2 data (benign cancer group) as our phase I sample. This yields $p = 9$ variables with $N = 80$. The distribution of this data is unknown and definitively not multivariate normal. Consequently, control charts based on multivariate normality cannot be used here. Hence, LS-SVDD and SVDD charts, both distribution free, are used instead. An in-control ARL of 200 is used to obtain the control limit.

For the LS-SVDD chart, the tuning parameters were selected according to Choi (2009). We then compute the α's using (Equation 19). For the SVDD chart, we compute the β's using Equation 4. Since there is no analytical solution for (Equation 4) like in the LS-SVDD case, a quadratic programming solver will be required in this case to optimize the dual problem.

Our phase II sample is formed by using the last 5 observations of the target class 2 data (benign cancer group) and the first 8 observations of the target class 4 data (malignant cancer group). This yields a total of 13 observations used in the phase II sample. We applied the LS-SVDD and SVDD charts to the 13 observations in the monitoring set and compare their performance in detecting changes in the process.

The results for LS-SVDD chart are shown on the left panel of Figure 2 after computing the statistics d_i, Equation 23 for the 13 observations in the phase II sample. Calculating the statistics d_u, (Equation 6), for the next 13 observations in the monitoring set yields the results for SVDD chart, shown on the right panel of Figure 2.

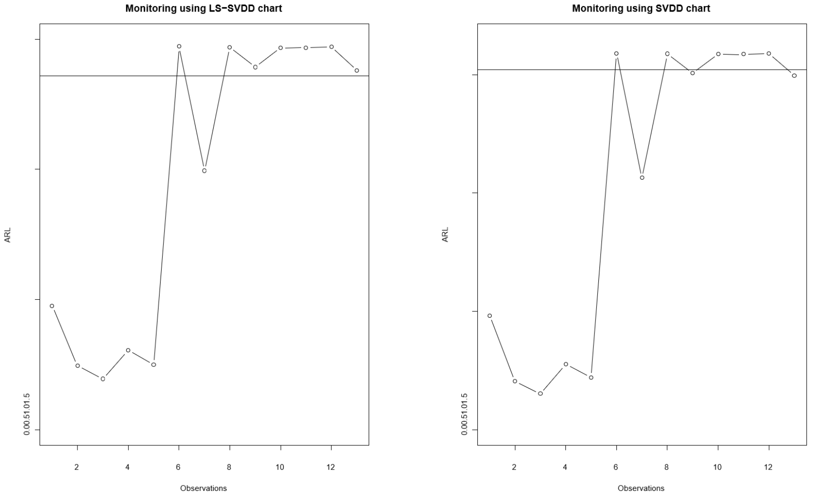

Figure 2. LS-SVDD and SVDD charts for the Breast cancer dataset example. (left) LS-SVDD chart, (right) SVDD chart.

Both charts show a similar trend as the process evolves. The first 5 observations, which belong to the target class 2 data (benign cancer group), are correctly classified to be in control by the two charts. Out of the 8 observations of the target class 4 data (malignant cancer group), 7 of them are correctly classified as being out of control by the LS-SVDD chart. Only observation 7 is incorrectly classified to be in-control. On the other hand, out of the 8 observations from target class 4 data, the SVDD chart correctly classifies 5 of them as out-of-control and incorrectly mis-classifies observations 7, 9, and 13 as in-control.

This demonstrates the effectiveness of the proposed control chart in a real application when dealing with a process, which may or may not follow the multivariate normal distribution.

7 Conclusion

This paper proposed an alternative one-class classification method to monitor the mean vector of a process. The least squares support vector data description estimates a boundary around a set of vectors such that it encloses a volume in the feature space.

The data description (i.e. finding the α's) with this proposal is obtained using an analytical solution in a closed form. Consequently, this proposal does not have the same issue as SVDD, i.e., using quadratic programming solvers, and therefore, is simpler to use than SVDD. This satisfies the objective (i) stated in the introduction of not using quadratic programming to solve the problem. Moreover, it satisfies objective (ii) stated in the introduction of obtaining the solutions in a closed form. However, LS-SVDD does lose the sparseness property of the SVDD. This loss can be overcome by pruning training samples, as in de Kruif & de Vries (2003) and Kuh & de Wilde (2007). This proposal does not rely on distributional assumptions such as multivariate normal assumptions. The use of kernels allows the boundary control shape to adapt easily to real data and therefore improving the monitoring process.

The performance of this method was discussed and compared to some existing charts like the SVDD chart proposed by Sun & Tsung (2003) and T^2 charts. Simulations show that the LS-SVDD chart produces very good results as compared to the SVDD and T^2 charts.

This satisfies the objective (iii) stated in the introduction of proposing a support vector control chart using a closed form solution for its design step. Moreover, this proposed chart is performing as well as SVDD chart and therefore can offer tremendous opportunities in design advanced monitoring charts with support vector methods.

We illustrate the application of LS-SVDD and SVDD on the ‘breast cancer data’. The results indicate that LS-SVDD performs as well as SVDD. The results obtained indicate that the overall performance of this chart makes it an appealing choice for monitoring the mean vector.

Acknowledgements

We would like to thank the editor and the anonymous referees for their constructive comments and suggestions that have considerably improved this paper.

References

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization. Stanford University*. New York: Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511804441.

Camci, F., Chinnam, R. B., & Ellis, R. D. (2008). Robust Kernel distance multivariate control chart using support vector principles. *International Journal of Production Research*, 46(18), 5075-5095. http://dx.doi.org/10.1080/00207540500543265.

Choi, Y. S. (2009). Least squares one-class support vector machine. *Pattern Recognition Letters*, 30(13), 1236-1240. http://dx.doi.org/10.1016/j.patrec.2009.05.007.

Cortes, C., & Vapnik, V. (1995). Support-vector network. *Machine Learning*, 20(3), 273-297. http://dx.doi.org/10.1007/BF00994018.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics (Oxford, England)*, 9(3), 432-441. http://dx.doi.org/10.1093/biostatistics/kxm045. PMid:18079126.

Gani, W., Taleb, H., & Liman, M. (2011). An Assessment of the Kernel-distance-based Multivariate Control Chart through an Industrial Application. *Quality and Reliability Engineering International*, 27(4), 391-401. http://dx.doi.org/10.1002/qre.1117.

Guo, Y., Xiao, H., & Fu, Q. (2017). Least square support vector data description for HRRP-based radar target recognition. *Journal of Applied Intelligence*, 46(2), 365-372. http://dx.doi.org/10.1007/s10489-016-0836-5.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1/3), 389-422. http://dx.doi.org/10.1023/A:1012487302797.

Kang, J. H., & Kim, S. B. (2011). Clustering-Algorithm-based Control Charts for Inhomogeneously Distributed TFT-LCD Processes. *International Journal of Production Research*, 51(18), 5644-5657. http://dx.doi.org/10.1080/00207543.2013.793427.

de Kruif, B. J., & de Vries, T. J. A. (2003). Pruning error minimization in least squares support vector machines. *IEEE Transactions on Neural Networks*, 14(3), 696-702. http://dx.doi.org/10.1109/TNN.2003.810597. PMid:18238050.

Kumar, S., Choudhary, A. K., Kumar, M., Shankar, R., & Tiwari, M. K. (2006). Kernel distance-based robust support vector methods and its application in developing a robust K-chart. *International Journal of Production Research*, 44(1), 77-96. http://dx.doi.org/10.1080/00207540500216037.

Kuh, A., & De Wilde, P. (2007). Comments on Pruning Error Minimization in Least Squares Support Vector Machines. *IEEE Transactions on Neural Networks*, 18(2), 606-609. http://dx.doi.org/10.1109/TNN.2007.891590. PMid:17385646.

Li, B., Wang, K., & Yeh, A. B. (2013). Monitoring covariance matrix via penalized likelihood estimation. *IIE Transactions*, 45(2), 132-146. http://dx.doi.org/10.1080/0740817X.2012.663952.

Liu, C., & Wang, T. (2014). An AK-chart for the Non-Normal Data. *International Journal of Computer, Information, Systems and Control Engineering*, 8, 992-997.

Liu P, Choo KKR, Wang L, Huang F (2016). SVM or deep learning? A comparative study on remote sensing image classification, *Soft Comput.* 43(2), 113-126.

Maboudou-Tchao, E. M., & Diawara, N. (2013). A lasso chart for monitoring the covariance matrix. *Quality Technology & Quantitative Management*, 10(1), 95-114. http://dx.doi.org/10.1080/16843703.2013.11673310.

Maboudou-Tchao, E. M., & Agboto, V. (2013). Monitoring the covariance matrix with fewer observations than variables. *Computational Statistics & Data Analysis*, 64, 99-112. http://dx.doi.org/10.1016/j.csda.2013.02.028.

Maboudou-Tchao, E. M., Silva, I., & Diawara, N. (2018). Monitoring the mean vector with Mahalanobis kernels. *Quality Technology & Quantitative Management*, 15(4), 459-474. http://dx.doi.org/10.1080/16843703.2016.1226707.

Maboudou-Tchao, E. M. (2018). Kernel methods for changes detection in covariance matrices. *Communications in Statistics. Simulation and Computation*, 47(6), 1704-1721. http://dx.doi.org/10.1080/03610918.2017.1322701.

Maboudou-Tchao, E. M. (2019). High-dimensional data monitoring using support machines. *Communications in Statistics. Simulation and Computation*, 1-16. http://dx.doi.org/10.1080/03610918.2019.1588312.

Maboudou-Tchao, E. M. (2020). Change detection using least squares one-class classification control chart. *Quality Technology & Quantitative Management*, 17(5), 609-626. http://dx.doi.org/10.1080/16843703.2019.1711302.

Maboudou-Tchao, E. M. (2021). Support tensor data description. *Journal of Quality Technology*, 53(2), 109-134. http://dx.doi.org/10.1080/00224065.2019.1642815.

Montgomery, D. C. (2001). *Introduction to Statistical Quality Control* (4th ed.). New York: Wiley.

Ning, X., & Tsung, F. (2013). Improved design of Kernel-Distance-Based charts using Support Vector Methods. *IIE Transactions*, 45(4), 464-476. http://dx.doi.org/10.1080/0740817X.2012.712237.

Rodriguez-Lujan, I., Huerta, R., Elkan, C., & Cruz, C. S. (2010). Quadratic programming feature selection. *Journal of Machine Learning Research*, 11, 1491-1516.

Sukchotrat, T., Kim, S. B., & Tsung, F. (2009). One-Class classification-based control charts for multivariate process monitoring. *IIE Transactions*, 42(2), 107-120. http://dx.doi.org/10.1080/07408170903019150.

Sun, R., & Tsung, F. A. (2003). Kernel-distance-based multivariate control charts using support vector methods. *International Journal of Production Research*, 41(13), 2975-2989. http://dx.doi.org/10.1080/1352816031000075224.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293-300. http://dx.doi.org/10.1023/A:1018628609742.

Tax, D., & Duin, R. (1999). Support vector domain description. *Pattern Recognition Letters*, 20(11-13), 1191-1199. http://dx.doi.org/10.1016/S0167-8655(99)00087-2.

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 1, 1-16. http://dx.doi.org/10.1186/s13634016-0355-x.

Yeh, A. B., Li, B., & Wang, K. (2012). Monitoring multivariate process variability with individual observations via penalized likelihood estimation. *International Journal of Production Research*, 50(22), 6624-6638. http://dx.doi.org/10.1080/00207543.2012.676684.

Wang, H., & Hu, D. (2005). Comparison of SVM and LS-SVM for regression. In *International conference on neural networks and brain* (Vol. 1, pp. 279-283). Beijing, China: IEEE.

Weese, M., Martinez, W., & Jones-Farmer, L. A. (2017). On the selection of the bandwidth parameter for the k-Chart. *Quality and Reliability Engineering International*, 33(7), 1527-1547. http://dx.doi.org/10.1002/qre.2123.

Weese, M., Martinez, W., Megahed, F. M., & Jones-Farmer, L. A. (2016). Statistical Learning Methods Applied to Process Monitoring: An Overview and Perspective. *Journal of Quality Technology*, 48(1), 4-24. http://dx.doi.org/10.1080/00224065.2016.11918148.