

Kernel Ellipsoidal Trimming

A.N. Dolia^{*}, C.J. Harris^a, J.S. Shawe-Taylor^b,
D.M. Titterington^c

^a*School of Electronics and Computer Science, University of Southampton,
Southampton, SO17 3AS, UK*

^b*Centre for Computational Statistics and Machine Learning, University College
London, Gower Street, London, WC1E 6BT, UK*

^c*Department of Statistics, University of Glasgow, Glasgow, G12 8QQ, UK*

Abstract

Ellipsoid estimation is important in many practical areas such as control, system identification, visual/audio tracking, experimental design, data mining, robust statistics and statistical outlier or novelty detection. A new method, called Kernel Minimum Volume Covering Ellipsoid (KMVCE) estimation, that finds an ellipsoid in a kernel-defined feature space is presented. Although the method is very general and can be applied to many of the aforementioned problems, the main focus is on the problem of statistical novelty/outlier detection. A simple iterative algorithm based on Mahalanobis-type distances in the kernel-defined feature space is proposed for practical implementation. The probability that a non-outlier is misidentified by our algorithms is analysed using bounds based on Rademacher complexity. The KMVCE method performs very well on a set of real-life and simulated datasets, when compared with standard kernel-based novelty detection methods.

Key words: minimum volume covering ellipsoid, Rademacher complexity, kernel methods, outlier detection, novelty detection

^{*} Corresponding author: Southampton Statistical Sciences Research Institute, University of Southampton, Southampton, SO17 3AS, UK, Tel. (44)-23-8059-3216, Fax.: (44)-23-8059-5363.

Email addresses: od@soton.ac.uk (A.N. Dolia), cjh@ecs.soton.ac.uk (C.J. Harris), jst@cs.ucl.ac.uk (J.S. Shawe-Taylor), mike@stats.gla.ac.uk (D.M. Titterington).

1 Introduction

Outlier/novelty detection is a problem that arises in many applications such as credit card fraud, weather prediction, image enhancement, loan approval, object detection and condition monitoring. Surveys of outlier detection in statistics are given by Barnett and Lewis (1994) and Hawkins (1980). There is no precise definition of what outliers are but, intuitively, an outlier can be defined as an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism (Hawkins, 1980). The main focus of outlier detection in statistics is robust estimation of the parameters of the statistical model when the data contain outliers (Rousseeuw and Leroy, 1997; Hawkins, 1980; Barnett and Lewis, 1994; Croux et al., 2002; Woodward and Sain, 2003; Hardin and Rocke, 2004).

In the activity known as novelty detection, the primary problem is not to detect outliers among the observations, all regarded as “normal”, that are used to estimate the parameters of the statistical model, but rather to detect whether or not a single new measurement is “abnormal” or “anomalous” (Nairac et al., 1997). Here and in the rest of the paper the adjective “normal” is used as antonym of “outlier” or “anomalous”, and not as a synonym of “Gaussian”.

Novelty detection is concerned with identifying data that are atypical for the given distribution (Tax and Duin, 1999; Schölkopf et al., 1998). In a typical application in engine monitoring systems, estimation of the model is based on observations from normal operation since abnormal measurements are either very rare or simply not available for problems that have not occurred before. Once the parameters of the model have been estimated, the system monitors new data and reports either “normal pattern”, if the observation conforms to the given/estimated distribution, or “abnormal pattern”, if the observation appears to deviate significantly from the data used to estimate parameters of the model. A novelty detector must identify as “small” a normal region as possible while still reliably identifying data outside this region as abnormal. Whilst we cannot guarantee that data identified as normal are not misclassified, ensuring that the normal region is small guards against this possibility. In the context of misclassification of normal patterns as abnormal we present theoretical results bounding the probability that a normal observation is identified as abnormal.

In this paper, we develop the outlier detection approach based on calculation of the Minimum Volume Covering Ellipsoid (MVCE) (Titterton, 1978). The MVCE must contain the entire dataset. When a new observation arrives a “novelty score” based on a Mahalanobis-type distance is used to assess whether or not the new point is an outlier. If that exceeds a pre-defined threshold the new observation is considered to be an outlier.

The problem of calculating the MVCE becomes extremely difficult in a multi-dimensional kernel-defined feature space (defined in Section 2) when the dimension of that space might be unknown or the matrix that defines the MVCE is singular, i.e. positive semi-definite rather than positive definite. In order to deal with singularity, the proposed KMVCE method finds a minimum volume ellipsoid and calculates Mahalanobis-type distances in a subspace of the kernel-defined feature space. If we use a (possibly nonlinear) mapping to define the kernel-defined feature space then the observed data lie inside the MVCE in this space. The boundary of the MVCE in the kernel-defined feature space defines an estimate of the support of the distribution of observations in the original space.

The paper is organized as follows. Kernel Principal Component Analysis is introduced in Section 2. Our approach to calculating the MVCE centered on the origin is given in Section 3. The calculation of the KMVCE with optimally chosen center based on a simple iterative algorithm is discussed in Section 4. We compare the performance of the one-class Support Vector Machine (SVM), the Linear Programming Novelty Detection algorithm (LPND) and the KMVCE method in Section 5 followed by conclusions and acknowledgements in Section 6. The probability that a non-outlier is misidentified by the KMVCE centered on the origin is analysed using bounds based on Rademacher complexity in Appendix.

2 Kernel Principal Component Analysis (PCA)

In this section we introduce the kernel PCA and basic operations with kernels. For a theoretical account of kernel-defined feature spaces see Wahba (1990) and Vapnik (1998).

Consider a vector of random variables $\mathbf{x} \in X \subseteq \mathbb{R}^k$ with an unknown probability density function $p(\mathbf{x})$. We are provided with realizations that can help to reveal the structure of \mathbf{x} , that is with a dataset $D_n = \{\mathbf{x}_i \in X\}_{i=1}^n$ made up of a random sample from the distribution. The problem is to find a feature space or a possibly nonlinear mapping ϕ of the observations $\mathbf{x}_1, \dots, \mathbf{x}_n$ such that we can better explore the data structure and uncover relationships between the random variables, in particular with a view to detecting outliers.

Let the entries in the matrix $\mathbf{K} = \mathbf{X}\mathbf{X}'$ be the inner products $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where \mathbf{X} is defined by

$$\mathbf{X} = \begin{bmatrix} \phi_1(\mathbf{x}_1), \phi_2(\mathbf{x}_1), \dots, \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2), \phi_2(\mathbf{x}_2), \dots, \phi_m(\mathbf{x}_2) \\ \vdots \\ \phi_1(\mathbf{x}_n), \phi_2(\mathbf{x}_n), \dots, \phi_m(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} \phi(\mathbf{x}_1)' \\ \phi(\mathbf{x}_2)' \\ \vdots \\ \phi(\mathbf{x}_n)' \end{bmatrix}, \quad (1)$$

where each $\phi_r(\cdot)$ is a nonlinear mapping. If we define a function κ by

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = \phi(\mathbf{x})' \phi(\mathbf{z}),$$

then

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1), \kappa(\mathbf{x}_1, \mathbf{x}_2), \dots, \kappa(\mathbf{x}_1, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1), \kappa(\mathbf{x}_2, \mathbf{x}_2), \dots, \kappa(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots \\ \kappa(\mathbf{x}_n, \mathbf{x}_1), \kappa(\mathbf{x}_n, \mathbf{x}_2), \dots, \kappa(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}.$$

The function $\kappa(\mathbf{x}, \mathbf{z})$ is referred to as the kernel function and the matrix \mathbf{K} is termed the kernel or Gram matrix.

In order to analyze relationships between different random variables we explore the matrix $\frac{1}{n} \mathbf{X}' \mathbf{X}$, for example by performing PCA. However in order to uncover data structure and relationships between different observations $\phi(\mathbf{x}_i)$, one can analyze the kernel matrix \mathbf{K} . The kernel matrix shows the degree of proximity between observations $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ in the kernel-defined feature space.

In practice we are interested in cases where κ can be evaluated directly without explicit computation of the mapping ϕ . For example, if the mapping $\phi(\mathbf{x})$ is defined by

$$\phi : \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \longrightarrow \phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \in \mathbb{R}^3,$$

then the inner product between the vectors $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)'$ and $\phi(\mathbf{z}) = (z_1^2, z_2^2, \sqrt{2}z_1z_2)'$, i.e. the value of the function $\kappa(\mathbf{x}, \mathbf{z})$, is equal to

$$\begin{aligned} \kappa(\mathbf{x}, \mathbf{z}) &= \phi(\mathbf{x})' \phi(\mathbf{z}) = \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 = (\mathbf{x}'\mathbf{z})^2. \end{aligned}$$

In practice, it is common to use kernel functions $\kappa(\mathbf{x}, \mathbf{z})$ such as Gaussian radial basis functions or polynomials of degree ν , such as $\kappa(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z} + c)^\nu$, where c is a predefined constant; for example, if $c = 0$ and $\nu = 2$ we obtain $\kappa(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z})^2$, as above. The term ‘‘Gaussian kernel’’ is often used to define one of the following kernel functions:

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-0.5\|\mathbf{x} - \mathbf{z}\|^2/\rho^2\right), \quad (2)$$

and

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\|\mathbf{x} - \mathbf{z}\|^2/\rho^2\right). \quad (3)$$

Theorem 1 (Mercer, 1909). *If κ is a continuous kernel of a positive definite integral operator on $L_2(C)$ (where C some compact space) then it can be expanded as*

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{z}), \quad (4)$$

using eigenfunctions $\{\psi_i(\mathbf{x})\}_{i=1}^{\infty}$ and eigenvalues $\{\lambda_i > 0\}_{i=1}^{\infty}$ satisfying $\sum_{i=1}^{\infty} \lambda_i^2 < \infty$. The series converges absolutely and uniformly for each pair (\mathbf{x}, \mathbf{z}) on each compact subset of X .

In this case

$$\phi(\mathbf{x}) = \begin{bmatrix} \sqrt{\lambda_1} \psi_1(\mathbf{x}) \\ \sqrt{\lambda_2} \psi_2(\mathbf{x}) \\ \vdots \end{bmatrix}. \quad (5)$$

According to Mercer’s theorem, the size of the dimension m of the feature space $\phi(\cdot)$ can be less, equal or greater than the number of observations n . It depends on the speed of convergence of the values $\lambda_1, \lambda_2, \dots$ to zero. The Gaussian kernel is one of the examples when m could be infinite.

One way of applying the ellipsoid algorithm in high-dimensional feature spaces is first to project the data into a low-dimensional subspace and then to apply the algorithm to the projected data. Perhaps the most natural way to do this is to use kernel principal components analysis (a well-known method in machine learning, see Schölkopf et al. (1998); Schölkopf and Smola (2001)).

If we consider a singular value decomposition (SVD) of the matrix \mathbf{X} into $\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{U}'$, then

$$\mathbf{X}'\mathbf{X} = \mathbf{U}\mathbf{D}'\mathbf{D}\mathbf{U}' = \mathbf{U}\mathbf{\Lambda}\mathbf{U}'.$$

The columns \mathbf{u}_i of \mathbf{U} are the principal eigenvectors of the matrix $\mathbf{X}'\mathbf{X}$ with eigenvalues λ_i and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_m)$. Similarly the columns \mathbf{v}_i of \mathbf{V} are the eigenvectors of the Gram matrix $\mathbf{X}\mathbf{X}'$ with the same eigenvalues λ_i . Furthermore,

$$\mathbf{X}'\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}'\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{I}_m, \text{ so that } \mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{X}'\mathbf{v}_i.$$

This shows that the principal vectors in the feature space can be expressed as linear combinations of the sample projections with weights proportional to the corresponding eigenvector of the kernel matrix. This equation forms the basis of kernel PCA since we can evaluate the projection $\phi(\mathbf{x})$ of a new point onto a principal vector \mathbf{u}_i (Schölkopf et al., 1998)

$$\begin{aligned} \langle \mathbf{u}_i, \phi(\mathbf{x}) \rangle &= \frac{1}{\sqrt{\lambda_i}}\phi(\mathbf{x})'\mathbf{X}'\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{k}'\mathbf{v}_i, \quad i = 1, \dots, m, \\ k_j &= \langle \phi(\mathbf{x}), \phi(\mathbf{x}_j) \rangle = \kappa(\mathbf{x}, \mathbf{x}_j), \quad j = 1, \dots, n, \end{aligned}$$

where k_j is the j th element of \mathbf{k} .

In the next section we consider applying the ellipsoid algorithm directly in a subspace of kernel-defined feature space that forms the basis of the KMVCE algorithm.

3 Working in high-dimensional feature spaces

In this section we review basic facts related to calculation of the minimum volume covering ellipsoid. We describe our approach to calculating the MVCE centered on the origin in kernel defined feature spaces.

3.1 Conventional MVCE centered on the origin

Assume that we have a multivariate dataset containing n observations, $\{\phi(\mathbf{x}_i)\}_{i=1}^n$, that is not contaminated by outliers. In order to calculate the MVCE centered on the origin we need to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \log \det(\mathbf{M}) \\ \text{subject to} \quad & d(\phi(\mathbf{x}_i)) \leq k, \quad i = 1, 2, \dots, n, \end{aligned} \tag{6}$$

where $d(\phi(\mathbf{x}_i)) = \phi(\mathbf{x}_i)'\mathbf{M}^{-1}\phi(\mathbf{x}_i)$. This constrained minimization problem has a corresponding ‘dual’ maximization problem and the Strong Lagrangian

Principle implies that the problems share a common extreme value (Titterington, 1978). The dual optimization problem for the origin-centered MVCE problem can be written as follows (Titterington, 1978):

$$\begin{aligned} & \max_{\boldsymbol{\alpha}} \quad \log \det \mathbf{M}(\boldsymbol{\alpha}) & (7) \\ & \text{subject to} \\ & \quad \sum_{i=1}^n \alpha_i = 1, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, n, \end{aligned}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ are nonnegative numbers summing to 1 and $\mathbf{M}(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)'$.

Let $d(\phi(\mathbf{x}_i), \boldsymbol{\alpha}) = \phi(\mathbf{x}_i)' \mathbf{M}(\boldsymbol{\alpha})^{-1} \phi(\mathbf{x}_i)$. Then the maximum of $\log \det \mathbf{M}(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ can be characterized in terms of $d(\phi(\mathbf{x}_i), \boldsymbol{\alpha})$ by the following theorem. Note that $\phi(\mathbf{x}_i)$ is a vector in m -dimensional space.

Theorem 2 (Kiefer and Wolfowitz, 1960). *If $\sum_{i=1}^n \alpha_i = 1$ and if $\boldsymbol{\alpha} \geq 0$ then the following three statements are equivalent:*

- (i) $\boldsymbol{\alpha}^* = \operatorname{argmax}_{\boldsymbol{\alpha}} \log \det \mathbf{M}(\boldsymbol{\alpha})$;
- (ii) $\boldsymbol{\alpha}^* = \operatorname{argmin}_{\boldsymbol{\alpha}} \max_{j \in \{1, \dots, n\}} d(\phi(\mathbf{x}_j), \boldsymbol{\alpha})$;
- (iii) $\max_{j \in \{1, \dots, n\}} d(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^*) = m$.

In order to calculate $\boldsymbol{\alpha}^*$ we can employ the iterative algorithm (Titterington, 1978)

$$\alpha_j^{r+1} = \alpha_j^r \frac{d(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^r)}{m}, \quad r = 0, 1, 2, \dots, \quad (8)$$

where probability measure $\boldsymbol{\alpha}^r$ is the value of $\boldsymbol{\alpha}$ after the r th iteration; we can initialize by taking $\alpha_j^0 = 1/n$, for $j = 1, \dots, n$. The limit of the sequence $\{\boldsymbol{\alpha}^0, \boldsymbol{\alpha}^1, \dots\}$ is the required $\boldsymbol{\alpha}^*$ (Titterington, 1978). Therefore, the MVCE centered on the origin can be written as

$$\varepsilon(\mathbf{0}, \mathbf{M}(\boldsymbol{\alpha}^*), m) = \{\phi(\mathbf{x}) \in \mathbb{R}^m : d(\phi(\mathbf{x}), \boldsymbol{\alpha}^*) \leq m\}. \quad (9)$$

3.2 The kernel MVCE centered on the origin

For simplicity, we use the notation \mathbf{M} instead of $\mathbf{M}(\boldsymbol{\alpha})$. As shown above, a key quantity is

$$d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \phi(\mathbf{x})' \mathbf{M}^{-1} \phi(\mathbf{x}),$$

where we can write

$$\mathbf{M} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)' = (\mathbf{A}\mathbf{X})'(\mathbf{A}\mathbf{X}) = \mathbf{X}'\mathbf{A}^2\mathbf{X},$$

in which \mathbf{A} is a diagonal matrix with $\mathbf{A}_{ii} = \sqrt{\alpha_i}$. Therefore, the problem is to calculate $d(\phi(\mathbf{x}), \boldsymbol{\alpha})$ using the given kernel function $\kappa(\mathbf{x}, \mathbf{x}_j)$ without explicit calculation of the $\phi(\mathbf{x}_i)$. We address this issue in the following theorem.

Theorem 3 *Given that $\mathbf{M} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)'$, with $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i = 1$, then $d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \phi(\mathbf{x})' \mathbf{M}^{-1} \phi(\mathbf{x})$ can be written as a function of the kernel $\kappa(\mathbf{x}, \mathbf{x}_j)$ in the form $d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^m \lambda_i^{-2} \left(\sum_{j=1}^n \sqrt{\alpha_j} \mathbf{v}_{ji} \kappa(\mathbf{x}_j, \mathbf{x}) \right)^2$, where \mathbf{v}_{ji} is the j -th element of the eigenvector \mathbf{v}_i corresponding to the eigenvalue λ_i of the matrix $\mathbf{A}\mathbf{X}\mathbf{X}'\mathbf{A} = \mathbf{A}\mathbf{K}\mathbf{A}$, with $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$.*

Proof: Let $\mathbf{S}_\alpha = \mathbf{A}\mathbf{X}$. Then a singular-value decomposition of the matrix \mathbf{S}_α can be written as $\mathbf{S}_\alpha = \mathbf{A}\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}_\alpha^{1/2}\mathbf{U}'$, where the matrix $\boldsymbol{\Lambda}_\alpha$ is the $n \times m$ matrix (the same dimensions as \mathbf{S}_α) with nonnegative diagonal elements λ_i in decreasing order of magnitude; \mathbf{V} and \mathbf{U} are $n \times n$ and $m \times m$ unitary matrices with eigenvectors $\{\mathbf{v}_i\}_{i=1}^n$ and $\{\mathbf{u}_i\}_{i=1}^m$, respectively; both matrices \mathbf{V} and \mathbf{U} have orthogonal columns so that $\mathbf{V}\mathbf{V}' = \mathbf{I}_n$ and $\mathbf{U}\mathbf{U}' = \mathbf{I}_m$, where the matrices \mathbf{I}_n and \mathbf{I}_m are identity matrices of size n and m , respectively. Note that $\boldsymbol{\Lambda}_\alpha$

can be represented as a block matrix $\boldsymbol{\Lambda}_\alpha = \begin{pmatrix} \boldsymbol{\Lambda} \\ \mathbf{0} \end{pmatrix}$, where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_m)$,

and the matrix $\mathbf{0}$ is a zero matrix of size $(n - m) \times m$. Let $\boldsymbol{\Lambda}_\alpha^{-1}$ be equal to $(\boldsymbol{\Lambda}^{-1}, \mathbf{0})'$. Then $\mathbf{U} = (\boldsymbol{\Lambda}_\alpha^{-1/2} \mathbf{V}' \mathbf{S}_\alpha)' = \mathbf{X}' \mathbf{A} \mathbf{V} \boldsymbol{\Lambda}_\alpha^{-1/2}$. Obviously, if \mathbf{S}_α , \mathbf{U} and $\boldsymbol{\Lambda}$ are known then we can obtain the matrix \mathbf{M} as

$$\mathbf{M} = \mathbf{S}_\alpha' \mathbf{S}_\alpha = (\mathbf{V} \boldsymbol{\Lambda}_\alpha^{1/2} \mathbf{U}')' (\mathbf{V} \boldsymbol{\Lambda}_\alpha^{1/2} \mathbf{U}') = \mathbf{X}' \mathbf{A} \mathbf{X} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}',$$

and $\mathbf{M}^{-1} = (\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}')^{-1} = \mathbf{U} \boldsymbol{\Lambda}^{-1} \mathbf{U}'$, where we use the fact that $\boldsymbol{\Lambda} = (\boldsymbol{\Lambda}_\alpha^{1/2})' \boldsymbol{\Lambda}_\alpha^{1/2}$.

Note that $\mathbf{S}_\alpha \mathbf{S}_\alpha' = (\mathbf{V} \boldsymbol{\Lambda}_\alpha^{1/2} \mathbf{U}') (\mathbf{V} \boldsymbol{\Lambda}_\alpha^{1/2} \mathbf{U}')'$, which implies that $\mathbf{A}\mathbf{X}\mathbf{X}'\mathbf{A} =$

$\mathbf{V} \boldsymbol{\Lambda}_k \mathbf{V}'$, where $\boldsymbol{\Lambda}_k$ is the $n \times n$ matrix such that $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_\alpha^{1/2} (\boldsymbol{\Lambda}_\alpha^{1/2})' = \begin{pmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$.

It follows that we can obtain the i th eigenvector \mathbf{u}_i using the matrix \mathbf{K} and the matrix \mathbf{A} :

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{X}' \mathbf{A} \mathbf{v}_i, \quad (10)$$

where \mathbf{v}_i is the eigenvector corresponding to the eigenvalue λ_i of the corresponding kernel matrix $\mathbf{A}\mathbf{X}\mathbf{X}'\mathbf{A} = \mathbf{A}\mathbf{K}\mathbf{A}$. Note that the vectors \mathbf{u}_i and \mathbf{v}_i are now dependent on α , but we have suppressed this dependence to enhance readability. Now consider

$$d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \phi(\mathbf{x})' \mathbf{U} \boldsymbol{\Lambda}^{-1} \mathbf{U}' \phi(\mathbf{x}) = \sum_{i=1}^m \lambda_i^{-1} (\mathbf{u}_i' \phi(\mathbf{x}))^2. \quad (11)$$

After substitution of (10) into (11) we obtain

$$d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^m \lambda_i^{-1} \left(\left(\frac{1}{\sqrt{\lambda_i}} \mathbf{X}' \mathbf{A} \mathbf{v}_i \right)' \phi(\mathbf{x}) \right)^2.$$

Since $(\mathbf{X}' \mathbf{A} \mathbf{v}_i)' = (\mathbf{v}_i' \mathbf{A}' \mathbf{X})$ we obtain

$$d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^m \lambda_i^{-1} \left(\frac{1}{\sqrt{\lambda_i}} \mathbf{v}_i' \mathbf{A}' \mathbf{X} \phi(\mathbf{x}) \right)^2 = \sum_{i=1}^m \lambda_i^{-2} \left(\mathbf{v}_i' \mathbf{A} \begin{bmatrix} \phi(\mathbf{x}_1)' \phi(\mathbf{x}) \\ \vdots \\ \phi(\mathbf{x}_n)' \phi(\mathbf{x}) \end{bmatrix} \right)^2,$$

which shows that $d(\phi(\mathbf{x}), \boldsymbol{\alpha})$ can be calculated using the kernel function $\kappa(\mathbf{x}_j, \mathbf{x})$ instead of the obtaining the inner product $\phi(\mathbf{x}_i)' \phi(\mathbf{x})$ explicitly:

$$d(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^m \lambda_i^{-2} (\mathbf{v}_i' \mathbf{A} \boldsymbol{\kappa}_{\mathbf{x}})^2, \quad (12)$$

where $\boldsymbol{\kappa}_{\mathbf{x}} = (\kappa(\mathbf{x}_1, \mathbf{x}), \dots, \kappa(\mathbf{x}_n, \mathbf{x}))'$. \square

4 Optimal selection of the MVCE centre

In this section we consider three problems. The first one is the extension of the kernel MVCE to the case in which the center of the kernel MVCE has to be chosen optimally. The second task is to estimate the dimensionality of the unknown mapping ϕ for the given kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_y)$. The third problem is to deal with possible singularity of the matrix \mathbf{M} .

4.1 Conventional MVCE with the center chosen optimally

In many practical cases, the structure of the random variable \mathbf{x} can be better explained by the MVCE, with the center chosen optimally, that includes all observations from the data set D_n . In order to find this ellipsoid we need to obtain an $(m \times m)$ positive definite matrix $\mathbf{M}_{\mathbf{c}} \in \mathbb{R}^{m \times m}$ and the center of the ellipsoid $\mathbf{c}_{\phi} \in \mathbb{R}^m$, so as to minimize $\det \mathbf{M}_{\mathbf{c}}$ subject to (Titterington, 1978)

$$(\phi(\mathbf{x}_i) - \mathbf{c}_\phi)' \mathbf{M}_c^{-1} (\phi(\mathbf{x}_i) - \mathbf{c}_\phi) \leq m, \quad i = 1, \dots, n. \quad (13)$$

The dual optimization problem of the MVCE problem has its roots in D-optimal experimental design (Titterington, 1978) and is that of maximizing $\log \det \mathbf{M}_c(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$, where $\mathbf{M}_c(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i (\phi(\mathbf{x}_i) - \mathbf{c}_\phi(\boldsymbol{\alpha})) (\phi(\mathbf{x}_i) - \mathbf{c}_\phi(\boldsymbol{\alpha}))'$ and $\mathbf{c}_\phi(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i)$; as before, $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ are nonnegative numbers summing to 1. Note that the matrix $\mathbf{M}_c(\boldsymbol{\alpha})$ can be also written as $\mathbf{M}_c(\boldsymbol{\alpha}) = \mathbf{M}(\boldsymbol{\alpha}) - \mathbf{c}_\phi(\boldsymbol{\alpha}) \mathbf{c}_\phi(\boldsymbol{\alpha})'$, where $\mathbf{M}(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)'$ (see Titterington (1978) for details). We write

$$d_c(\phi(\mathbf{x}), \boldsymbol{\alpha}) = (\phi(\mathbf{x}_i) - \mathbf{c}_\phi(\boldsymbol{\alpha}))' \mathbf{M}_c^{-1}(\boldsymbol{\alpha}) (\phi(\mathbf{x}_i) - \mathbf{c}_\phi(\boldsymbol{\alpha})). \quad (14)$$

Therefore, given the optimal values $\boldsymbol{\alpha}^*$, the MVCE, $\varepsilon_{\mathbf{c}_\phi}(\mathbf{c}_\phi(\boldsymbol{\alpha}^*), \mathbf{M}_c(\boldsymbol{\alpha}^*), m)$, is defined by (Titterington, 1978)

$$\varepsilon_{\mathbf{c}_\phi}(c(\boldsymbol{\alpha}^*), \mathbf{M}_c(\boldsymbol{\alpha}^*), m) = \{\phi(\mathbf{x}) \in \mathbb{R}^m : d_c(\phi(\mathbf{x}), \boldsymbol{\alpha}^*) \leq m\}. \quad (15)$$

The MVCE for the dataset $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ must go through at least $m + 1$ and at most $\frac{1}{2}m(m + 3) + 1$ support points, that is, points \mathbf{x}_i such that the corresponding α_i is greater than zero (Titterington, 1978). There could be more than $\frac{1}{2}m(m + 3) + 1$ points on the surface of the ellipsoid, but $\frac{1}{2}m(m + 3) + 1$ is the largest number that are necessary. If $\alpha_i^* > 0$ then the corresponding $\phi(\mathbf{x}_i)$ from the dataset is on the boundary of the ellipsoid $\varepsilon_{\mathbf{c}_\phi}(c(\boldsymbol{\alpha}^*), \mathbf{M}_c(\boldsymbol{\alpha}^*), m)$. If $\alpha_j^* = 0$ it is still possible that the point $\phi(\mathbf{x}_j)$ lies on the boundary of the ellipsoid, but this can be checked by seeing whether or not $d_c(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^*) = m$.

The problem of finding the MVCE when the center of the ellipsoid $\mathbf{c}_\phi(\boldsymbol{\alpha})$ must be chosen optimally is identical (in the sense of values $\boldsymbol{\alpha}$) to determination of a MVCE, *centered at the origin*, in an $(m + 1)$ -dimensional space (Titterington,

1978). If $\tilde{\phi}(\mathbf{x}) = \begin{bmatrix} \phi(\mathbf{x}) \\ 1 \end{bmatrix}$ and $\tilde{\mathbf{M}}(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i \tilde{\phi}(\mathbf{x}_i) \tilde{\phi}(\mathbf{x}_i)' = \begin{bmatrix} \mathbf{M}(\boldsymbol{\alpha}) & \mathbf{c}_\phi(\boldsymbol{\alpha}) \\ \mathbf{c}_\phi(\boldsymbol{\alpha})' & 1 \end{bmatrix}$

then an ellipsoid $\tilde{\varepsilon}(\mathbf{0}, \tilde{\mathbf{M}}(\boldsymbol{\alpha}), \tilde{m})$, centered at the origin, in $(m + 1)$ -dimensional space can be defined as

$$\tilde{\varepsilon}(\mathbf{0}, \tilde{\mathbf{M}}(\boldsymbol{\alpha}), \tilde{m}) = \{\tilde{\phi}(\mathbf{x}) \in \mathbb{R}^{\tilde{m}} : \tilde{d}(\tilde{\phi}(\mathbf{x}), \boldsymbol{\alpha}) \leq \tilde{m}\}, \quad (16)$$

where $\tilde{d}(\tilde{\phi}(\mathbf{x}), \boldsymbol{\alpha}) = \tilde{\phi}(\mathbf{x}_i)' \tilde{\mathbf{M}}(\boldsymbol{\alpha})^{-1} \tilde{\phi}(\mathbf{x}_i)$ and $\tilde{m} = m + 1$. In order to find the optimal values $\boldsymbol{\alpha}$ one can use the appropriate version of (8):

$$\alpha_j^{r+1} = \alpha_j^r \frac{\tilde{d}(\tilde{\phi}(\mathbf{x}_j), \boldsymbol{\alpha}^r)}{\tilde{m}}, \quad j = 1, 2, \dots, n, r = 0, 1, \dots, \quad (17)$$

initialized at $\alpha_j^0 = 1/n$, for $j = 1, \dots, n$. The limit of the sequence $\{\boldsymbol{\alpha}^0, \boldsymbol{\alpha}^1, \dots\}$ is the $\boldsymbol{\alpha}^*$. The following results hold (Titterton, 1978):

- (i) $\boldsymbol{\alpha}^r$ contains nonnegative numbers summing to 1 for all r ;
- (ii) the sequence $\{\det \tilde{\mathbf{M}}(\boldsymbol{\alpha}^r) : r = 0, 1, \dots\}$ is monotonic increasing unless $\boldsymbol{\alpha}^r$ is equal to the optimal value, $\boldsymbol{\alpha}^*$;
- (iii) the same is true for $\{\det \mathbf{M}_c(\boldsymbol{\alpha}^r) : r = 0, 1, \dots\}$ because $\det \tilde{\mathbf{M}}(\boldsymbol{\alpha}) = \det \mathbf{M}_c(\boldsymbol{\alpha})$ for any $\boldsymbol{\alpha}$;
- (iv) in the limit, $\boldsymbol{\alpha}^r$ converges to the optimal value $\boldsymbol{\alpha}^*$ that defines the MVCE.

In practice, one can use the fact that $\max_j \tilde{d}(\tilde{\phi}(\mathbf{x}_j), \boldsymbol{\alpha}^*) = \tilde{m}$ and stop the iteration if

$$|\max_j \tilde{d}(\tilde{\phi}(\mathbf{x}_j), \boldsymbol{\alpha}^r) - \tilde{m}| \leq \epsilon, \quad (18)$$

where ϵ is a small predefined constant.

4.2 Kernel MVCE

Let

$$\mathbf{X}_c = \begin{bmatrix} (\phi(\mathbf{x}_1) - \mathbf{c}_\phi(\boldsymbol{\alpha}))' \\ (\phi(\mathbf{x}_2) - \mathbf{c}_\phi(\boldsymbol{\alpha}))' \\ \vdots \\ (\phi(\mathbf{x}_n) - \mathbf{c}_\phi(\boldsymbol{\alpha}))' \end{bmatrix}, \quad \tilde{\mathbf{X}} = \begin{bmatrix} \phi(\mathbf{x}_1)', 1 \\ \phi(\mathbf{x}_2)', 1 \\ \vdots, 1 \\ \phi(\mathbf{x}_n)', 1 \end{bmatrix}, \quad (19)$$

where $\mathbf{c}_\phi(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i)$, and let $\mathbf{S}_\alpha^c = \mathbf{A}\mathbf{X}_c$. Then a singular-value decomposition of the matrix \mathbf{S}_α^c can be written as $\mathbf{S}_\alpha^c = \mathbf{A}\mathbf{X}_c = \mathbf{V}_c \boldsymbol{\Lambda}_c^{1/2} \mathbf{U}_c$, where the diagonal matrix $\boldsymbol{\Lambda}_c$ is the $n \times m$ matrix with nonnegative diagonal elements λ_i^c ; \mathbf{V}_c and \mathbf{U}_c are $n \times n$ and $m \times m$ unitary matrices with eigenvectors $\{\mathbf{v}_i^c\}_{i=1}^n$ and $\{\mathbf{u}_i^c\}_{i=1}^m$, respectively; $\mathbf{V}_c \mathbf{V}_c' = \mathbf{I}_n$ and $\mathbf{U}_c \mathbf{U}_c' = \mathbf{I}_m$.

Then we can follow the proof of Theorem 3 but substituting $(\phi(\mathbf{x}) - \mathbf{c}_\phi(\boldsymbol{\alpha}))$ and \mathbf{X}_c for $\phi(\mathbf{x})$ and \mathbf{X} , respectively. Therefore,

$$d_c(\phi(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^m \lambda_i^{-2} \left((\mathbf{v}_i^c)' \mathbf{A} \begin{bmatrix} \kappa_c(\mathbf{x}_1, \mathbf{x}) \\ \vdots \\ \kappa_c(\mathbf{x}_n, \mathbf{x}) \end{bmatrix} \right)^2, \quad (20)$$

where the centered kernel function $\kappa_c(\mathbf{x}_s, \mathbf{x}_j)$ can be expressed in terms of the kernel function $\kappa(\cdot, \cdot)$ as follows:

$$\begin{aligned} \kappa_c(\mathbf{x}_s, \mathbf{x}_j) &= (\phi(\mathbf{x}_s) - c_\phi(\boldsymbol{\alpha}))' (\phi(\mathbf{x}_j) - c_\phi(\boldsymbol{\alpha})) \\ &= \phi(\mathbf{x}_s)' \phi(\mathbf{x}_j) - \phi(\mathbf{x}_s)' c_\phi(\boldsymbol{\alpha}) - c_\phi(\boldsymbol{\alpha})' \phi(\mathbf{x}_j) + c_\phi(\boldsymbol{\alpha})' c_\phi(\boldsymbol{\alpha}) \\ &= \kappa(\mathbf{x}_s, \mathbf{x}_j) - \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_s, \mathbf{x}_i) - \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i,y=1}^n \alpha_i \alpha_y \kappa(\mathbf{x}_i, \mathbf{x}_y). \end{aligned}$$

A similar approach can be taken to find the MVCE centered on the origin in $(m+1)$ -dimensional space. As discussed before, the solution of this problem corresponds to the MVCE with the center chosen optimally. Let $\tilde{\phi}(\mathbf{x}_i)' = (\phi(\mathbf{x}_i)', 1) \in \tilde{\mathbf{X}}$, so that $\tilde{\phi}(\mathbf{x}_i)' \tilde{\phi}(\mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) + 1 = \kappa(\mathbf{x}_i, \mathbf{x}_j) + 1$. Then we can follow the proof of Theorem 3, but substituting $\tilde{\phi}(\mathbf{x})$ and $\tilde{\mathbf{X}}$ for $\phi(\mathbf{x})$ and \mathbf{X} , respectively.

Let $\mathbf{S}_\alpha^{\tilde{\phi}} = \mathbf{A} \tilde{\mathbf{X}}$. Then a singular-value decomposition of the matrix $\mathbf{S}_\alpha^{\tilde{\phi}}$ can be written as $\mathbf{S}_\alpha^{\tilde{\phi}} = \mathbf{A} \tilde{\mathbf{X}} = \tilde{\mathbf{V}} \tilde{\boldsymbol{\Lambda}}^{1/2} \tilde{\mathbf{U}}$, where $\tilde{\boldsymbol{\Lambda}}$ is the $n \times (m+1)$ matrix with nonnegative elements $\tilde{\lambda}_i$; $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{U}}$ are $n \times n$ and $(m+1) \times (m+1)$ unitary matrices with eigenvectors $\{\tilde{\mathbf{v}}_i\}_{i=1}^n$ and $\{\tilde{\mathbf{u}}_i\}_{i=1}^{m+1}$, respectively; $\tilde{\mathbf{V}} \tilde{\mathbf{V}}' = \mathbf{I}_n$ and $\tilde{\mathbf{U}} \tilde{\mathbf{U}}' = \mathbf{I}_{m+1}$.

Therefore,

$$\tilde{d}(\tilde{\phi}(\mathbf{x}), \boldsymbol{\alpha}) = \sum_{i=1}^{m+1} \tilde{\lambda}_i^{-1} \left(\tilde{\mathbf{u}}_i' \tilde{\phi}(\mathbf{x}) \right)^2 = \sum_{i=1}^{m+1} \tilde{\lambda}_i^{-2} \left(\tilde{\mathbf{v}}_i' \mathbf{A} \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}) + 1 \\ \vdots \\ \kappa(\mathbf{x}_n, \mathbf{x}) + 1 \end{bmatrix} \right)^2.$$

Note that $\tilde{d}(\tilde{\phi}(\mathbf{x}), \boldsymbol{\alpha}) = d_c(\phi(\mathbf{x}), \boldsymbol{\alpha}) + 1$ (Titterton, 1978).

In practice, the value of m can be unknown for the given kernel function $\kappa(\cdot, \cdot)$ and we propose a simple heuristic to estimate the dimension of the vector $\phi(\mathbf{x})$. The value of m is equal to the number of nonzero eigenvalues λ_i^c obtained after eigenvector decomposition of the matrix \mathbf{K}_c when $\alpha_1 = \dots = \alpha_n = 1/n$. Note that this value corresponds to the number of nonzero eigenvalues obtained as a result of the eigenvector decomposition of the sample

covariance matrix, $\mathbf{M}_c(\boldsymbol{\alpha}) = \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)' - \frac{1}{n^2} \bar{\phi}(\mathbf{x})\bar{\phi}(\mathbf{x})'$. In practice, in order to acknowledge the influence of round-off errors during computations we compare the values of λ_i^c not with zero but with a predefined threshold t . In this case we try also to avoid problems with possible singularity of the matrix $\mathbf{M}_c(\boldsymbol{\alpha})$ during computation of $d_c(\phi(\mathbf{x}), \boldsymbol{\alpha})$.

Recall that the MVCE for the dataset $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ must go through at least $m+1$ and at most $\frac{1}{2}m(m+3)+1$ support points, for which $\alpha_i > 0$ (Titterington, 1978). Therefore, in the case of the MVCE method the choice of m can be also dictated by the following two extreme cases $m+1 \leq n$ and $\frac{1}{2}m(m+3)+1 \leq n$.

We summarize our ideas in the Algorithm below. The algorithm is initialized at $\alpha_j^0 = 1/n$, for $j = 1, \dots, n$.

After obtaining α^* we can use the following rules to check if a new point \mathbf{x}_{new} belongs to the estimated support of the distribution:

$$d_c(\phi(\mathbf{x}_{new}), \boldsymbol{\alpha}^*) < \eta \quad \text{indicates a normal pattern;} \quad (21)$$

$$d_c(\phi(\mathbf{x}_{new}), \boldsymbol{\alpha}^*) = \eta \quad \text{indicates a point on the surface of the MVCE;} \quad (22)$$

$$d_c(\phi(\mathbf{x}_{new}), \boldsymbol{\alpha}^*) > \eta \quad \text{indicates an outlier,} \quad (23)$$

where $\eta = \max_j d_c(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^*)$, for $j = 1, \dots, n$. According to the *Kiefer-Wolfowitz* theorem, for α^* , $\eta = m$. If we treat η as a free parameter then we can re-scale the MVCE, $\varepsilon_{\mathbf{c}_\phi}(\mathbf{c}_\phi(\boldsymbol{\alpha}^*), \mathbf{M}_c(\boldsymbol{\alpha}^*), \eta)$, with $\eta = m + \gamma$. See Appendix for theoretical analysis giving a Rademacher complexity bound (A.1) on the probability that a non-outlier is misidentified by the KMVCE algorithm. In the next section we will use re-scaling of the MVCE in ellipsoidal trimming in order to deal with outliers.

5 Experiments.

In this section the performance of the KMVCE method with the optimally chosen center is analyzed on simulated and real-life datasets. We compare our results with two standard novelty detection methods.

5.1 Simulated dataset

We use an artificial dataset that is similar to one previously reported by Campbell and Bennett (2000) for novelty detection based on the linear programming approach. In our experiment we generate a sample from the bivariate

Algorithm 1 Kernel MVCE Algorithm

Initialization: Define the kernel matrix \mathbf{K} , the number of iterations r_{max} , the threshold t and α . For example, in the condition monitoring experiment (see Section 5.2) \mathbf{K} was a Gaussian kernel with $\rho = 296$, $r_{max} = 150$, $t = 0.0001$, and $\alpha_j^0 = 1/n$, for $j = 1, \dots, n$.

The main loop of the algorithm:

for $r = 0, \dots, r_{max} - 1$ **do**

1. Find the ‘centered’ kernel matrix \mathbf{K}_c

$$\mathbf{K}_c = \mathbf{K} - \mathbf{1}_n(\alpha^r)' \mathbf{K} - \mathbf{K}' \alpha^r \mathbf{1}_n' + b \mathbf{1}_n \mathbf{1}_n',$$

where $b = \sum_i^n \sum_j^n \mathbf{B}_{ij}$, $\mathbf{B} = \hat{\mathbf{A}} \mathbf{K} \hat{\mathbf{A}}$ and $\hat{\mathbf{A}} = \text{diag}(\alpha_1^r, \dots, \alpha_n^r)$.

2. Obtain eigenvectors $\mathbf{v}_i \in \mathbf{V}$ and eigenvalues $\lambda_i \in \lambda$ of the matrix $\mathbf{A} \mathbf{K}_c \mathbf{A}$ where $\mathbf{A} = \text{diag}(\sqrt{\alpha_1^r}, \dots, \sqrt{\alpha_n^r})$:

$$\mathbf{V} \mathbf{\Lambda} \mathbf{V}' = \mathbf{A} \mathbf{K}_c \mathbf{A}, \text{ where } \mathbf{V} \text{ and } \mathbf{\Lambda} = \text{diag}(\lambda) \text{ are } n \times n \text{ matrices.}$$

3. Sort λ_i in decreasing order and correspondingly permute the columns of the matrix \mathbf{V} .

During the first iteration estimate dimensionality of $\phi(\cdot)$, m :

if m is known and $n \geq \frac{1}{2}m(m+3) + 1$ **then**

$$\hat{m} = m$$

else

Set \hat{m} equal to the number of eigenvalues λ_i , for $i = 1, \dots, n$, that are greater or equal to t .

if $n \leq \frac{1}{2}\hat{m}(\hat{m}+3) + 1$ **then**

Find \hat{m} as a solution of the equation $0.5\hat{m}^2 + 1.5\hat{m} + (1-n) = 0$:

$$\hat{m} = \lfloor -1.5 + \sqrt{2.25 + 2(n-1)} \rfloor; \text{ where } n \text{ is known and } \lfloor R \rfloor \text{ denotes the largest integer not exceeding } R;$$

end if

end if

for $j = 1, \dots, n$ **do**

4. Calculate the Mahalanobis norm $d_c(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^r)$ for the training point $\phi(\mathbf{x}_j)$:

$$d_c(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^r) = \sum_{i=1}^{\hat{m}} \lambda_i^{-2} \left(\mathbf{v}_i' \mathbf{A} \begin{bmatrix} \kappa_c(\mathbf{x}_1, \mathbf{x}_j) \\ \vdots \\ \kappa_c(\mathbf{x}_n, \mathbf{x}_j) \end{bmatrix} \right)^2.$$

5. Obtain new values for α_j using the algorithm (Titterton, 1978):

$$\alpha_j^{r+1} = \alpha_j^r d_c(\phi(\mathbf{x}_j), \boldsymbol{\alpha}^r) / \hat{m}.$$

end for

end for

return $\alpha^* = \alpha^{r_{max}-1}$

Gaussian distribution $N(\mu, \Sigma)$ with mean $\mu = \begin{pmatrix} 10 \\ 5 \end{pmatrix}$ and covariance matrix

$\Sigma = \begin{pmatrix} 0.0163 & -0.0062 \\ -0.0062 & 0.0163 \end{pmatrix}$. There are 4 outliers (see Fig. 1). We use the simple

inner product kernel $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}'\mathbf{y}$ with $t = 0.001$ and of course $k = m = 2$. We first find the smallest ellipse that encloses all points. Then we remove from the dataset points on the boundary of that ellipse. We then find the smallest ellipse that contains the remaining points. This way of removing outliers is called ellipsoidal trimming (Titterton, 1978). Clearly our method successfully removed the four outliers. We used this approach to remove outliers from the dataset for the condition monitoring example that we describe below.

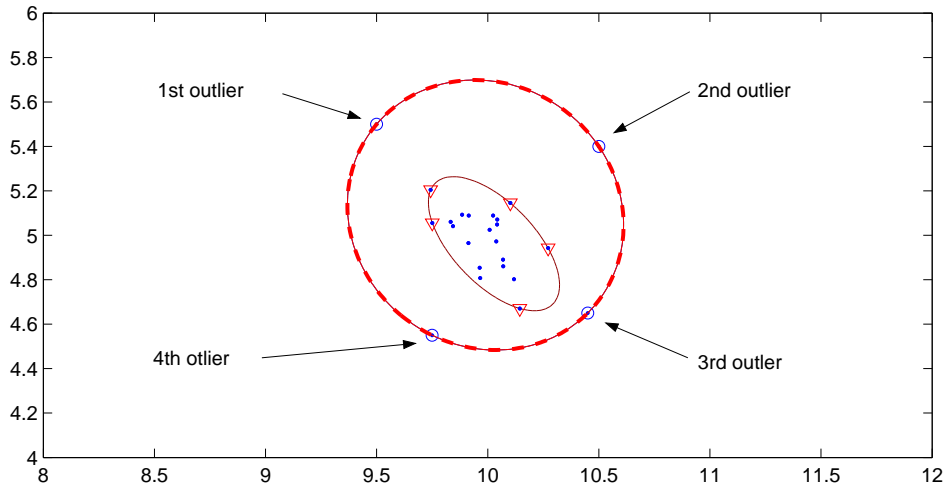


Fig. 1. Illustration of the the ellipsoidal trimming experiment. Both the larger (before the ellipsoidal trimming) and smaller (after the ellipsoidal trimming) ellipses are calculated by Algorithm 1. The kernel $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}'\mathbf{y}$, $(\phi(\mathbf{x}) = \mathbf{x})$ and $t = 0.001$ are used. Support points on the boundaries of these two ellipses are denoted by *circles* and *triangles*, respectively.

5.2 Condition monitoring

We analyse the comparative performance of the one-class SVM (Schölkopf and Smola, 2001), the proposed KMVCE method (see Algorithm) and the linear programming novelty detection algorithm (Campbell and Bennett, 2000) on a real-life dataset from the Structural Integrity and Damage Assessment Network (Campbell and Bennett, 2000). In this dataset vibration measurements s_k , obtained from a pump, correspond to “Healthy” (without a fault) measurements and 4 types of malfunction of machinery (see Fig. 2): 1) Fault 1 (the bearing had an outer race completely broken); 2) Fault 2 (broken cage with one loose element); 3) Fault 3 (broken cage with four loose elements); 4) Fault 4 (a badly worn ball-bearing with no apparent damage). A time series of length 700 is shown in Fig. 2 for each of these five situations.

In order to obtain a dataset $\{\mathbf{x}_j\}_{j=1}^n$ we estimate the power spectrum $|FFT(\cdot)|$ of s_k weighted by the Hanning window h_i . The vector \mathbf{x}_j , for $j = 1, \dots, n$, consists of half (the first 32 components) of the power spectrum $|FFT(\cdot)|$ obtained using the 64-tuple Fast Fourier transform $FFT(\cdot)$. There is no *a priori* information about the distribution of the random variable $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^k$, $k = 32$. The vectors \mathbf{x}_j obtained for the different conditions of the pump are shown in Fig. 3. It can be seen that when the pump is broken completely the spectrum of the signal s is significantly different from the pump without fault (see \mathbf{x}_j corresponding to “Healthy” and “Fault 1” in Fig. 3). However, if the pump has a different type of fault the difference between “Healthy” and “Faulty” pumps could be less obvious.

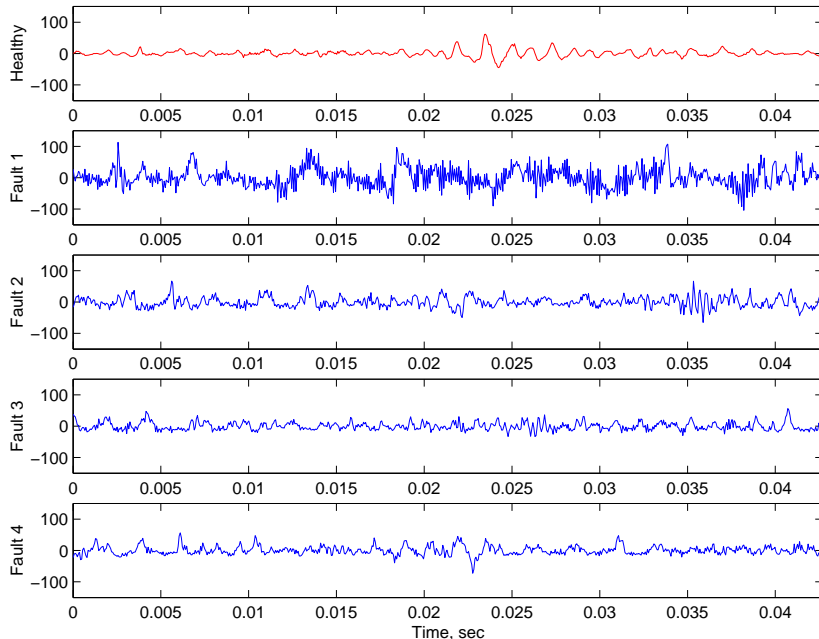


Fig. 2. Condition monitoring experiment. In this example measurements, on one channel, $\{s_k\}$ are obtained from a healthy machine and when four different faults are present (Campbell and Bennett, 2000).

In order to compare the KMVCE method with the LPND method and the one-class SVM method, we performed experiments in the same way as described by Campbell and Bennett (2000), using the Gaussian kernel and the same training set, validation set (see Stone (1977) and Kearns (1997) for properties of cross-validation) and test set. The training dataset $\{\mathbf{x}_j\}_{i=1}^n$ that is used to calculate the KMVCE consists of $n = 913$ vectors from the “Healthy” class. In order to validate the model we use another 913 vectors \mathbf{x}_j from the “Healthy” class. To analyse the performance of the KMVCE we use a dataset of 417 vectors from the “Healthy” class and 913 vectors \mathbf{x}_j for every class corresponding to the broken pump (Faults 1,2,3 and 4). We assume that the vectors \mathbf{x}_j that correspond to the different fault types are not available when we fit the KMVCE to the 913 vectors \mathbf{x}_j from the “Healthy” class. As a result

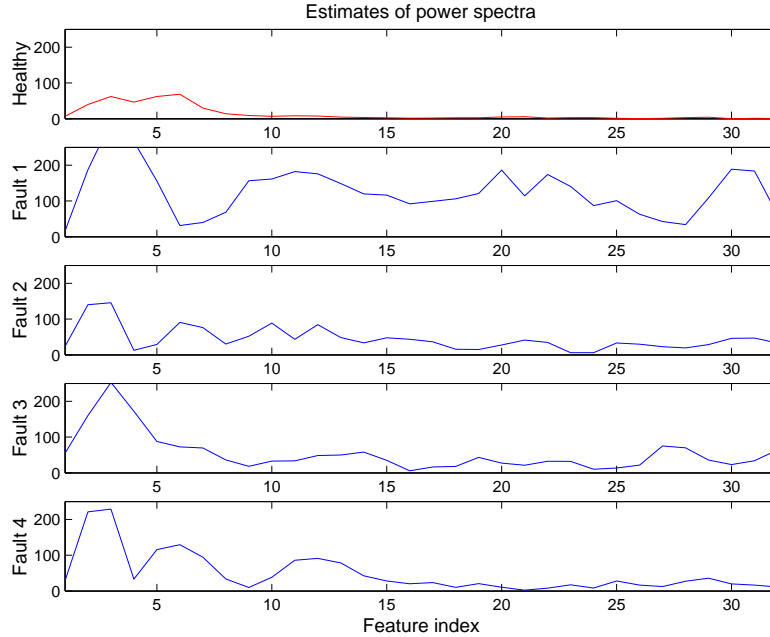


Fig. 3. Illustration of the features used in the condition monitoring experiment (Campbell and Bennett, 2000). The horizontal axis corresponds to the index of the component of the vector \mathbf{x} , and, therefore, the frequency of the signal s . The vertical axis is the estimate of the power spectrum using the Hanning window.

we cannot apply classical neural networks or support vector machine classifiers (Venables and Ripley, 2002; Shawe-Taylor and Cristianini, 2004). A decision about whether or not \mathbf{x} is an outlier (the pump is broken) has to be based on only the observations from the “Healthy” class.

In the experiments we use the Gaussian kernel (see (2)) with standard deviation equal to $\rho = 320$ for LPND (suggested by Campbell and Bennett (2000)) and the one-class SVM and the KMVCE method with $\rho = 226$. After training, the KMVCE algorithm showed very poor performance for Fault 1. The method labels almost all points from this class as “Healthy”. It suggests that the training sample contains outliers; for example, there are large vibrations between 0.2s and 0.3s, see as shown in Fig. 2.

In order to remove these outliers we carried out the same steps as with the artificial dataset (see Fig. 1). First we removed the points on the boundary. Secondly we re-trained our novelty detector based on the KMVCE using the remaining 793 points; Thirdly we scaled a newly obtained ellipse using a different η in order to achieve desirable errors of first and second kinds. For approximately the same rate of correct classification of the “Healthy” class ($\eta = 140$) our method performs better than the soft margin one-class SVM method, and when $\eta = 180$ or $\eta = 190$ the KMVCE method is significantly better than the LPND method (Campbell and Bennett, 2000). Note that the one-class SVM using a Gaussian kernel is equivalent to finding the hypersphere

Table 1

The percentage of correctly labelled data using LPND, one-class SVM and KMVCE methods

| Method | Healthy | Fault 1 | Fault 2 | Fault 3 | Fault 4 |
|---------------------|---------|---------|---------|---------|---------|
| LPND | 98.7% | 100% | 53.3% | 28.3% | 25.5% |
| one-class SVM | 97.9% | 100% | 84.3% | 57.3% | 61.1% |
| KMVCE, $\eta = 190$ | 99.8% | 100% | 79.2% | 50.5% | 52.4% |
| KMVCE, $\eta = 180$ | 99.6% | 100% | 82.9% | 54.4% | 57.7% |
| KMVCE, $\eta = 140$ | 97.7% | 100% | 93.9% | 72.8% | 76.8% |
| KMVCE, $\eta = 64$ | 79% | 100% | 100% | 97.5% | 98.8% |

around the data points (see Schölkopf and Smola (2001)) .

All three methods (LPND, 1-class SVM and kernel MVCE) were trained using cross-validation in such a way that about 2% of the data from the target population are rejected (see the column headed “Healthy” in Table 1) and such that all cases of “Fault 1” are detected. Thus all three methods behave equally in this sense while using different approaches to deal automatically with outliers.

Many authors regard a Gaussian kernel as a good choice for anomaly detection but the kernel choice depends on the problem under consideration. For example, if it is known that the target data cloud is ellipsoidal then we can use the simple inner product kernel (see Fig. 1). In a high-dimensional feature space, an application of the KMVCE is particular beneficial if the support of the distribution is nonconvex or multimodal.

6 Conclusions

We have proposed a new and very general Kernel Minimum Volume Covering Ellipsoid method and have shown how it can be applied to outlier detection. For example, if we use a different form of regularization the method can be further developed to obtain D -optimal experimental designs for the kernel ridge regression model. We have demonstrated that it is not always necessary to specify an accurate estimate of the proportion of outliers in advance. We have proposed a very simple but effective iterative algorithm that can be used for both anomaly detection and optimal experimental design. The KMVCE method has demonstrated better or similar performance on the real-life condition monitoring problem compared to the one-class SVM method and the

LPND method. For an appropriately chosen objective function the method of kernelisation proposed in this paper includes the Minimum Volume Covering Sphere as a special case and therefore it is also related to the one-class SVM model. The probability that a non-outlier is misidentified by the KMVCE is analyzed using bounds based on Rademacher complexity. In future work it would be interesting to see to what extent we can increase robustness of the proposed method using other methodologies to remove outliers.

Acknowledgments

This research was partially supported by the Data Information Fusion Defence Technology Center, United Kingdom, under DTC Projects 8.1: “Active multi-sensor management”. This work was also supported in part by the European Community, under the PASCAL Network of Excellence. The authors wish to thank reviewers for helpful comments.

References

- Barnett, V., & Lewis, T., 1994. *Outliers in Statistical Data*. 3rd ed, Wiley, Chichester.
- Bartlett, P.L. & Mendelson, S., 2002. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research* 3 463–482.
- Campbell, C. & Bennett, K.P., 2000. A Linear Programming Approach to Novelty Detection. In: *Advances in Neural Information Processing Systems*, Vol. 14, MIT Press, Cambridge, MA, 395–401.
- Croux, C., Haesbroeck, G., & Rousseeuw, P.J., 2002. Location Adjustment for the Minimum Volume Ellipsoid Estimator. *Statistical Computation*, 12(3) 191–200.
- Hardin, J. & Rocke, D.M., 2004. Outlier Detection in the Multiple Cluster Setting using the Minimum Covariance Determinant Estimator. *Computational Statistics & Data Analysis*, 44(4) 625–638
- Hawkins, D., 1980. *Identification of Outliers*. Chapman and Hall, London.
- Kiefer, J. & Wolfowitz, J., 1960. The Equivalence of Two Extremum Problems. *Canadian Journal of Mathematics*, 12 363–366.
- Kearns, M., 1997. A Bound on the Error of Cross Validation using the Approximation and Estimation Rates, with Consequences for the Training-test Split. *Neural Computation*, 9 1143–1161.
- Mercer, J., 1909. Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society of London, Series A*, 209 415–446.
- Nairac, A., Corbett-Clark, T., Ripley, R., Townsend, N. W., & Tarassenko, L., 1997. Choosing an Appropriate Model for Novelty Detection. In: *Proceed-*

- ings of the 5th IEE International Conference on Artificial Neural Networks, Cambridge, 117–122.
- Rousseeuw, P.J. & Leroy, A.M., 1987. Robust Regression and Outlier Detection. Wiley-Interscience, New York.
- Schölkopf, B., Smola, A., & Müller, K.-R., 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319.
- Schölkopf, B., & Smola, A., 2001. Learning with Kernels. MIT Press, Cambridge, MA.
- Shawe-Taylor, J. & Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge, UK.
- Shawe-Taylor, J., Williams, C., Cristianini, N. & Kandola, J. S., 2002. On the Eigenspectrum of the Gram Matrix and Its Relationship to the Operator Eigenspectrum. In: Proceedings of the 13th International Conference on Algorithmic Learning Theory (ALT2002), Vol. 2533, 23–40.
- Stone, M., 1977, Asymptotics for and Against Cross-validation, *Biometrika*, 64, 29–35.
- Tax, D.M.J., & Duin, R.P.W., 1999. Data Domain Description by Support Vectors. In: Verleysen, M. (ed.), Proceedings of ESANN, Brussels, 251–256.
- Titterton, D.M., 1978. Estimation of Correlation Coefficients by Ellipsoidal Trimming. *Applied Statistics*, 27(3) 227–234.
- Vapnik, V., 1998. Statistical Learning Theory. Wiley, NY.
- Venables, W.N. & Ripley, B.D., 2002. Modern Applied Statistics with S. Fourth Edition, Springer, NY.
- Wahba, G., 1990. Splines Models for Observational Data. Series in Applied Mathematics, Vol. 59. SIAM, Philadelphia.
- Woodward, W.A., & Sain, S.R. Testing for Outliers from a Mixture Distribution when some Data are Missing. *Computational Statistics & Data Analysis*, 44(1–2) 193–210.

A Rademacher Complexity Analysis for Outlier Detection

A.1 A short Introduction to Rademacher Complexity Theory

We begin with the definition of Rademacher complexity; see for example Bartlett and Mendelson (2002) and Shawe-Taylor and Cristianini (2004) for an introductory exposition.

Definition 4 For a sample $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ generated by a distribution \mathcal{D} on a set X and a real-valued function class \mathcal{F} with a domain X , the empirical Rademacher complexity of \mathcal{F} is the random variable

$$\hat{R}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{2}{n} \sum_{i=1}^n \sigma_i f(\mathbf{x}_i) \right| \middle| \mathbf{x}_1, \dots, \mathbf{x}_n \right],$$

where Rademacher random variables $\sigma = \{\sigma_1, \dots, \sigma_n\}$ are independent binary $\{\pm 1\}$ -valued random variables such as $\sigma = 2(\beta - 0.5)$, in which random variables β have a Bernoulli(p) distribution with $p=0.5$. The Rademacher complexity of \mathcal{F} is

$$R_n(\mathcal{F}) = \mathbb{E}_{\mathcal{D}} [\hat{R}_n(\mathcal{F})] = \mathbb{E}_{\mathcal{D}, \sigma} \left[\sup_{f \in \mathcal{F}} \left| \frac{2}{n} \sum_{i=1}^n \sigma_i f(\mathbf{x}_i) \right| \right].$$

Note that the Rademacher complexity measures the degree to which the function class can align with random labels, a concept that gives an intuitive measure of capacity.

We use $\mathbb{E}_{\mathcal{D}}$ to denote expectation with respect to a distribution \mathcal{D} and \mathbb{E}_S for its empirical value on the sample S , assuming uniform sampling. The Rademacher complexity allows us to bound uniformly the expectations of the values of all functions in a function class based on the class complexity. We first quote a theorem giving some properties of Rademacher complexity (Shawe-Taylor and Cristianini, 2004).

Theorem 5 *Let $\mathcal{F}, \mathcal{F}_1, \dots, \mathcal{F}_n$ and \mathcal{G} be classes of real functions. Then the following hold:*

- (i) *if $\mathcal{F} \subseteq \mathcal{G}$, then $\hat{R}_n(\mathcal{F}) \leq \hat{R}_n(\mathcal{G})$;*
- (ii) *for every $c \in \mathbb{R}$, $\hat{R}_n(c\mathcal{F}) = |c| \hat{R}_n(\mathcal{F})$;*
- (iii) *if $\mathcal{A} : \mathbb{R} \rightarrow \mathbb{R}$ is Lipschitz with constant L and satisfies $\mathcal{A}(0) = 0$, then $\hat{R}_n(\mathcal{A} \circ \mathcal{F}) \leq 2L \hat{R}_n(\mathcal{F})$.*

Theorem 6 *Fix $\delta \in (0, 1)$ and let \mathcal{F} be a class of functions mapping from S to $[0, A]$. Let $\{\mathbf{x}_i\}_{i=1}^n$ be drawn independently according to a probability distribution \mathcal{D} . Then, with probability at least $1 - \delta$ over random draws of samples of size n , every $f \in \mathcal{F}$ satisfies*

$$\mathbb{E}_{\mathcal{D}} [f(\mathbf{x})] \leq \mathbb{E}_S [f(\mathbf{x})] + \hat{R}_n(\mathcal{F}) + 3A \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

For the proof the reader is referred to Bartlett and Mendelson (2002) and Shawe-Taylor and Cristianini (2004) with the slight adaptation that the range is scaled from the standard $[0, 1]$ to $[0, A]$. Application of the standard result to an appropriate scaling of the functions together with part (ii) of Theorem 5 gives the result. Theorem 6 gives a uniform bound on the difference between the empirical and true expectation of any function drawn from a class in terms of its Rademacher complexity.

Given a training set S , the class of functions that we will primarily be considering are linear functions $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle$ with weights \mathbf{w} of bounded norm:

$$\left\{ x \rightarrow \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) : \alpha' \mathbf{K} \alpha \leq B^2 \right\} \subseteq \{ \mathbf{x} \rightarrow \langle \mathbf{w}, \phi(\mathbf{x}) \rangle : \|\mathbf{w}\| \leq B \} = \mathcal{F}_B,$$

where ϕ is the feature mapping corresponding to the kernel function κ and \mathbf{K} is the corresponding kernel matrix for the sample S . The following result bounds the Rademacher complexity of this kind of linear function class.

Theorem 7 *Bartlett and Mendelson (2002)*. *If $\kappa : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$ is a kernel, and $\mathbf{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is a random sample from \mathbf{X} , then the empirical Rademacher complexity of the class \mathcal{F}_B satisfies*

$$\hat{R}_n(\mathcal{F}) \leq \frac{2B}{n} \sqrt{\text{tr}(\mathbf{K})}.$$

The theorem shows that for linear functions the two factors that determine the Rademacher complexity are the trace of the kernel matrix and the bound on the norm of the weight vectors \mathbf{w} . In the next subsection we show how the expected misclassification probability for normal observations can be bounded using an underlying linear function class in a kernel-defined feature space.

A.2 Probabilistic bound using Rademacher complexity for the KMVCE

The novelty detection approach models the support of the distribution that generated the normal data. As described in the introduction the region is made as small as possible to guard against abnormal data being classified as normal, naturally, it increases the chance that it may misclassify normal data as abnormal. This section provides a probabilistic bound for this occurring. Following the approach adopted in the analysis of Shawe-Taylor et al. (2002) we can view $d(\phi(\mathbf{x}), \boldsymbol{\alpha})$ as a linear function in the space defined by the kernel $\hat{\kappa}(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{x}, \mathbf{z})^2$, since

$$\begin{aligned} d(\phi(\mathbf{x}), \boldsymbol{\alpha}) &= \sum_{i=1}^m \lambda_i^{-1} (\mathbf{u}_i' \phi(\mathbf{x}))^2 = \sum_{i=1}^m \lambda_i^{-1} \mathbf{u}_i \phi(\mathbf{x}) \phi(\mathbf{x})' \mathbf{u}_i \\ &= \left\langle \sum_{i=1}^m \lambda_i^{-1} \mathbf{u}_i \mathbf{u}_i, \phi(\mathbf{x}) \phi(\mathbf{x})' \right\rangle_F =: \langle \mathbf{w}, \phi(\mathbf{x}) \phi(\mathbf{x})' \rangle_F, \end{aligned}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product and

$$\hat{\kappa}(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \phi(\mathbf{x})', \phi(\mathbf{z}) \phi(\mathbf{z})' \rangle_F = (\phi(\mathbf{z})' \phi(\mathbf{x}))^2 = \kappa(\mathbf{x}, \mathbf{z})^2.$$

The norm of the weight vector \mathbf{w} is given by

$$\begin{aligned}\|\mathbf{w}\|^2 &= \left\langle \sum_{i=1}^m \lambda_i^{-1} \mathbf{u}_i \mathbf{u}_i', \sum_{j=1}^m \lambda_j^{-1} \mathbf{u}_j \mathbf{u}_j' \right\rangle_F \\ &= \sum_{i=1}^m \lambda_i^{-2} \|\mathbf{u}_i\|^2 = \sum_{i=1}^m \lambda_i^{-2}.\end{aligned}$$

Consider the function

$$g(z) = \min((z - m)_+ / \gamma, 1),$$

which has range $[0, 1]$, Lipschitz constant $1/\gamma$ and satisfies $g(0) = 0$. Note that

$$P(d(\phi(\mathbf{x}), \boldsymbol{\alpha}) > m + \gamma) \leq \mathbb{E}_{\mathcal{D}}[g(d(\phi(\mathbf{x}), \boldsymbol{\alpha}))],$$

while

$$\mathbb{E}_S[g(d(\phi(\mathbf{x}), \boldsymbol{\alpha}))] \leq \frac{1}{n\gamma} \sum_{i=1}^n (d(\phi(\mathbf{x}_i), \boldsymbol{\alpha}) - m)_+.$$

Combining the above we obtain the following.

Theorem 8 Fix $\delta \in (0, 1)$, $m \in \mathbb{N}$ and constant $C > 0$. Then, with probability at least $1 - \delta$ over random draws of samples $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of size n , if the output of the ellipsoid algorithm satisfies

$$\sum_{i=1}^m \lambda_i^{-2} \leq C^2,$$

then

$$\begin{aligned}P(d(\phi(\mathbf{x}), \boldsymbol{\alpha}) > m + \gamma) &\leq \frac{1}{n\gamma} \sum_{i=1}^n (d(\phi(\mathbf{x}_i), \boldsymbol{\alpha}) - m)_+ \\ &\quad + \frac{4C}{n\gamma} \sqrt{\sum_{i=1}^n \kappa(\mathbf{x}_i, \mathbf{x}_i)^2} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.\end{aligned}\tag{A.1}$$

Proof: The result follows from an application of Theorem 6 to the function $g(d(\phi(\mathbf{x}), \boldsymbol{\alpha}))$ which, by the condition, belongs to the class

$$\mathcal{F} = \{\mathbf{x} \mapsto g(\langle \mathbf{w}, \phi(\mathbf{x}) \phi(\mathbf{x})' \rangle_F)\}.$$

The above observations relate the true and empirical expectations to the quantities in the bound, while the Rademacher complexity can be bounded through an application of Theorems 5 and 7. \square

The regularization achieved by the dimension reduction is dictated by the form of the bound which shows that the term summing the inverse squares of the eigenvalues needs to be truncated to achieve a good overall bound. Therefore, the bound indicates the trade-off between the size of the first term on the right-hand side of the inequality, $\frac{1}{n\gamma} \sum_{i=1}^n (d(\phi(\mathbf{x}_i), \boldsymbol{\alpha}) - m)_+$, and the complexity, given by the eigenvalue sum. Interestingly the bound is not prescriptive about the dimension, which will depend on the way in which the data fill the space. What it does tell us is that the directions in which the data variation is just noise must be truncated.

Note that, if we use the Gaussian kernel (see (2), (3)), then $\kappa(\mathbf{x}, \mathbf{x}) = 1$ for any \mathbf{x} and therefore the bound can be simplified in this case:

$$P(d(\phi(\mathbf{x}), \boldsymbol{\alpha}) > m + \gamma) \leq \frac{1}{n\gamma} \sum_{i=1}^n (d(\phi(\mathbf{x}_i), \boldsymbol{\alpha}) - m)_+ + \frac{4C}{\gamma\sqrt{n}} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

The analysis suggests that we should truncate the eigenvector expansion before λ_k becomes small, as this will ensure that the probability of misclassification of the normal data is kept small.