Geometrical and computational aspects of Spectral Support Estimation for novelty detection

Alessandro Rudi[a,b], Francesca Odone[b], Ernesto De Vito[c]

[a]*Robotics, Brain and Cognitive Sciences Department - Istituto Italiano di Tecnologia. Via Morego 30, 16163, Genova - Italy*
[b]*Dipartimento di Informatica Bioingegneria Robotica e Ingegneria dei Sistemi - University of Genova. Via Dodecaneso 35, 16146, Genova - Italy*
[c]*Dipartimento di Matematica - University of Genova. Via Dodecaneso 35, 16146, Genova - Italy and INFN - Sezione di Genova, Via Dodecaneso 33, 16146, Genova - Italy*

Abstract

In this paper we discuss the Spectral Support Estimation algorithm [1] by analyzing its geometrical and computational properties. The estimator is non-parametric and the model selection depends on three parameters whose role is clarified by simulations on a two-dimensional space. The performance of the algorithm for novelty detection is tested and compared with its main competitors on a collection of real benchmark datasets of different sizes and types.

1. Introduction

Support estimation emerged in the sixties in statistics with the seminal works of Rényi and Sulanke [2] and Geffroy [3], and in the last decades became crucial in different fields of machine learning and pattern recognition as, just to mention a few, one class estimation [4], novelty and anomaly detection [5, 6]. These problems find applications in different domains where it is difficult to gather negative examples (as it often happens in biological and biomedical problems) or when the negative class is not well defined (as in object detection problems in computer vision).

Email addresses: alessandro.rudi@iit.it (Alessandro Rudi), francesca.odone@unige.it (Francesca Odone), devito@dima.unige.it (Ernesto De Vito)

Support estimation deals with the following setting. The population data are represented by d-dimensional column vectors of features, but they live in a proper subset $C \subset \mathbb{R}^d$ distributed according to some probability distribution $p(x)dv(x)$, where dv is a suitable infinitesimal *volume* element of C. For example, C could be a curve in $\mathbb{R}^d$, dv is the arc length and $p(x)$ is the density distribution of the data on the curve. Both the set C and the distribution $p(x)dv(x)$ are known only through a training set $\{x_1, \ldots, x_n\}$ of examples drawn independently from the population according to $p(x)dv(x)$. The aim of support estimation is to find a subset $C_n \subset \mathbb{R}^d$ such that C_n is similar to C, if n is large enough.

In this paper we focus on support estimation of a probability distribution, that is, given a training set of examples, we would like to define a set which is a good estimator of the support of the distribution, *i.e.* the smallest (closed) subset having probability one. To this purpose we review the Spectral Support Estimation algorithm introduced in De Vito et al. [1] with an emphasis on its geometrical and computational properties and on its applicability to real novelty detection problems.

To have good estimators some geometrical a-priori assumption on C is needed. For example, if C is convex, a choice for C_n is the convex hull of the training set, as first proposed in Dümbgen and Walther [7]. If C is an arbitrary set with non-zero d-dimensional Lebesgue measure, Devroye and Wise [8] define C_n has the union of the balls of center x_i and radius ϵ with ϵ going to 0 when the number of data increases. A different point of view is taken by the so-called plug-in estimators. In such approach one first provides an estimator of the probability density and then C_n is defined as the region with high density [9].

However, in many applications the data approximatively live on a low dimensional submanifold, whose Lebesgue measure is clearly zero, and one may take advantage of this a priori information by using some recent ideas about dimensionality reduction, as for example manifold learning algorithms [10, 11, and references therein] and kernel Principal Component Analysis [12]. Based on this idea, Hoffmann [13] proposes a new algorithm for novelty detection, which can be seen as a support estimation problem. This point of view is further developed in De Vito et al. [1], where a new class of consistent estimators, called Spectral Support Estimators (SSE), is proposed.

The contribution of this paper is threefold. First, we review the SSE algorithm emphasizing its geometrical and computational aspects (while we refer the reader interested in its statistical properties to De Vito et al. [1]). Sec-

ond, we discuss the dependence of the algorithms on its hyper-parameters with the help of a thorough qualitative analysis on synthetic data. This analysis also allows us to show the quality of the estimated support, which adapt nicely and smoothly to the training data, similarly to kernel PCA [13]. Third, we show the appropriateness of the algorithm on a large choice of real data and compare its performances against well known competitors, namely K-Nearest Neighbours, Parzen windows [14], one class Support Vector Machines [4], and kernel PCA for novelty detection [13]. To make the match fair, for each algorithm we select the optimal choice for the hyper-parameters following a procedure developed in Rudi et al. [15].

To have an intuition of the SSE algorithm, suppose C is a r-dimensional linear subspace of $\mathbb{R}^d$. Consider the $d \times d$-matrix

$$T = \int_C xx' \, p(x) dx,$$

here the volume element dv of C is simply the r-dimensional Lebesgue measure dx. It is easy to check that the null space of T is the orthogonal complement of C in $\mathbb{R}^d$, that is, C is the linear span of all the eigenvectors of T with non-zero eigenvalues. Since a consistent estimator of T is the empirical matrix $T_n = \frac{1}{n} \sum_{i=1}^n x_i x_i'$, one can define C_n as the linear span of the eigenvectors of T_n whose eigenvalue is bigger than a threshold λ. As in supervised learning, the thresholding ensures a stable solution with respect to the noise. Now, if λ goes to zero when n increases, C_n becomes closer and closer to C, providing us with a consistent estimator. Furthermore, to test if a new point x of $\mathbb{R}^d$ belongs to C or not, a simple decision rule is given by $F_n(x) = \sum_{\ell=1}^r (u_\ell' x)^2$, where $u_1, \ldots, u_r$ are the eigenvectors spanning C_n. Indeed, $0 \leq F_n(x) \leq x'x$ for all $x \in \mathbb{R}^d$, but it is close to $x'x$ (that is, the norm of x is near to the projection of x over C_n) if and only if x is near to C. Note that if T_n is replaced by the covariance matrix, then C_n is nothing else than the principal component analysis.

More in general, if C is not a linear subspace the above algorithm does not work, as it happens in binary classification problems with linear Support Vector Machines if the two classes are not linearly separated. This suggests the use the kernel trick which requires a feature map Φ, mapping the input space $\mathbb{R}^d$ into the feature space $\mathcal{H}$, with $\Phi(C)$ a linear subspace of $\mathcal{H}$. This strong condition is satisfied by the *separating reproducing kernels* introduced in De Vito et al. [1].

The reminder of the paper is organized as follows. In Section 2 we review the SSE algorithm. In Section 3 we discuss how the algorithm is influenced by the choice of the parameters, supporting our theoretical analysis with simulations on synthetic data. In Section 4 we compare SSE with other methods from the literature on a vast selection of real datasets. Section 5 is left to a final discussion.

2. A spectral algorithm for support estimation

In this section we describe the SSE algorithm. We first set the mathematical framework, hence we introduce the algorithm by discussing its geometrical interpretation, the role of the separating kernels and we give some examples of separating kernels. We then derive a simple implementation of the algorithm and observe how the algorithm can be implemented in different methods according to a specific choice of a filter (similarly to what was done in Lo Gerfo et al. [16] for the supervised case).

2.1. The framework

We assume that the input space is $\mathbb{R}^d$ with the euclidean scalar product $x't$ between two column vectors x and t. The population lives on a closed subset $C \subset \mathbb{R}^d$ and is distributed according to some probability density p only defined on C, namely

$$p(x) > 0 \quad \forall x \in C \qquad \text{and} \qquad \int_C p(x)dv(x) = 1,$$

where again dv is the infinitesimal *volume* element of C. For any measurable subset E of $\mathbb{R}^d$, we set

$$\rho(E) = \int_{C \cap E} p(x)dv(x),$$

then ρ is a probability measure on $\mathbb{R}^d$ and C is the smallest closed subset of $\mathbb{R}^d$ such that $\rho(C) = 1$, namely C is the support of the measure ρ. In general, ρ does not have density with respect to the Lebesgue measure of $\mathbb{R}^d$, as it always happens if C is an r-dimensional sub-manifold with $r < d$. Further, we assume the measure ρ is unknown, but we have a training set $\{x_1, \dots, x_n\}$ sampled independently and identically distributed according to ρ.

Our aim is to find a closed subset $C_n \subset \mathbb{R}^d$ such that C_n is statistically consistent, *i.e.* it becomes similar to C with high probability when the number of examples n goes to infinity.

Since in the general case C is not a linear subspace, we consider a suitable feature map Φ from the input space $\mathbb{R}^d$ into a Hilbert space $\mathcal{H}$, whose scalar product will be denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. As a common practice for kernel machines, we state the condition on the feature map in terms of its reproducing kernel $K(x,t) = \langle \Phi(x), \Phi(t) \rangle_{\mathcal{H}}$. As usual, we identify $\mathcal{H}$ with the reproducing kernel Hilbert space associated with K, so that the elements of $\mathcal{H}$ are functions on $\mathbb{R}^d$, the feature map is given by $\Phi(x) = K(\cdot, x) \in \mathcal{H}$, and for any $f \in \mathcal{H}$ and $x \in \mathbb{R}^d$, $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$ [17].

In the case of SSE we need to assume K satisfies the following properties:

i) *Mercer*: the map $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is continuous, *i.e* K is a Mercer kernel;

ii) *Normalization*: for all $x \in X$ it holds that $K(x,x) = 1$;

iii) *Separability*: for any closed subset $T \subset \mathbb{R}^d$ and any point $x \notin T$ there exists $f \in \mathcal{H}$ such that $\langle f, \Phi(x) \rangle_{\mathcal{H}} \neq 0$ and $\langle f, \Phi(t) \rangle_{\mathcal{H}} = 0$ for all $t \in T$.

As shown in De Vito et al. [1] this assumption is crucial to prove the statistical consistency of the SSE algorithm.

The requirement that K is a Mercer kernel is very natural for kernel machines, whereas the normalization assumption simply makes the computation easy and, as shown in De Vito et al. [1], the separating property is preserved after normalization. The crucial requirement is the separability condition. Indeed, it implies that

$$\Phi(C) = \overline{\text{span}}\{\Phi(x) \mid x \in C\} \cap \Phi(\mathbb{R}^d),$$

which means that $\Phi(C)$ is the intersection of a linear space of $\mathcal{H}$ and $\Phi(\mathbb{R}^d)$, here $\overline{\text{span}}\{\Phi(x) \mid x \in C\}$ is the closed subspace generated by the family $\{\Phi(x)\}_{x \in C}$.

Examples of separating kernels are given in De Vito et al. [1], here we list two general purpose kernels that can be applied on a large class of problems:

a) Laplacian (Abel) kernel:

$$K(x,t) = \exp(-\frac{|x-t|}{\gamma}) \tag{1}$$

where $\gamma > 0$ and $|x-t|$ is the euclidean norm in $\mathbb{R}^d$;

b) ℓ_1-kernel:

$$K(x,t) = \exp(-\sum_{j=1}^{d} |x^j - t^j|/\gamma) \tag{2}$$

where $\gamma > 0$ and x^j and t^j are the j-th component of the vectors x and t, respectively.

Notice that the Gaussian kernel does not have the separating property since the functions in the corresponding reproducing kernel Hilbert space are analytic [17].

2.2. The algorithm: Tikhonov regularization

We are now in position to describe the SSE algorithm. Following the intuition discussed in the introduction, we replace the matrix T_n by with its kernel version. Now T_n is a linear operator on $\mathcal{H}$ defined by

$$\begin{aligned} T_n f &= \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \langle f, \Phi(x_i) \rangle_{\mathcal{H}} \\ T_n f(x) &= \frac{1}{n} \sum_{i=1}^{n} K(x, x_i) f(x_i), \end{aligned} \tag{3}$$

for all $f \in \mathcal{H}$, which is positive and with finite rank.

Furthermore, given $\lambda > 0$ we introduce the regularized operator

$$P_n = (\lambda \, \mathrm{I} + T_n)^{-1} T_n,$$

where I is the identity operator. Note that, if f is an eigenfunction of T_n with eigenvalue σ, then $P_n f = \frac{\sigma}{\sigma+\lambda} f$, and the range of P_n is a *regularized version* of the linear span of the eigenfunctions of T_n with non-zero eigenvalue (Tikhonov regularization).

The empirical decision rule is given by

$$\begin{aligned} F_n(x) &= 1 - (\langle \Phi(x), \Phi(x) \rangle_{\mathcal{H}} - \langle P_n \Phi(x), \Phi(x) \rangle_{\mathcal{H}}) \\ &= \left\langle (\lambda \, \mathrm{I} + T_n)^{-1} T_n \Phi(x), \Phi(x \right\rangle_{\mathcal{H}} \end{aligned} \tag{4}$$

where, by construction, $0 \leq F_n(x) \leq 1$ for all $x \in \mathbb{R}^d$. The corresponding set estimator C_n is defined by

$$C_n = \{x \in \mathbb{R}^d \mid F_n(x) \geq 1 - \tau\},$$

where $\tau > 0$ is a threshold parameter. As it happens for kernel machines, the computation of F_n reduces to a finite dimensional problem, since it holds that

$$F_n(x) = k'_{n.x}(\lambda n \, \mathrm{I_n} + \mathbf{K}_n)^{-1} k_{n.x}, \tag{5}$$

where $\mathbf{K}_n$ is the $n \times n$ Gram matrix whose (i,j)-entry is $K(x_i, x_j)$ and $\mathbf{k}_{n,x}$ is the n-dimensional column vector whose i-th element is $K(x, x_i)$. The corresponding pseudo-code is listed in Algorithm 1, which shows that SSE algorithm can easily implemented in a few lines.

Algorithm 1 Spectral Support Estimator with a Tikhonov filter controlled by a regularization parameter `lambda` and a kernel implemented by the function `kernel`– Matlab Code.

```
function [rK] = learn_support(X,lambda,r);
    K=kernel(X,X);
    rK=pinv(K+lambda*eye(size(K,1))); % Tikhonov filter
end

function [Fn] = decision_rule(X, x,rK) % Eq. (3)
    kx=kernel(X,x);
    Fn = sum((rK*kx).*kx,1);
 end

%----------------------
% main script
.....
rk=learn_support(X,lambda,r);  % training set in matrix X
y=decision_rule(X,x,rK); % test datum x
y >= 1- tau; % membership to the set Cn
```

The Singular Value Decomposition of the Gram matrix $\mathbf{K}_n$ provides an alternative procedure to compute F_n. We start off from the SVD of $\mathbf{K}_n$

$$\mathbf{K}_n = \Big[v_1 | \ldots | v_r\Big] \; \mathrm{diag}(\sigma_1, \ldots, \sigma_r) \; \Big[v'_1 | \ldots | v'_r\Big],$$

where r is the rank of $\mathbf{K}_n$ and $v_1, \ldots, v_r \in \mathbb{R}^n$ are the eigenvectors with non-zero eigenvalue (and $v'_\ell v_\ell = 1$). Then, for any $\ell = 1, \ldots, r$ we define the

out-of-sample extension $f_\ell \in \mathcal{H}$ to be

$$f_\ell(x) = \frac{1}{\sqrt{n\sigma_\ell}} \sum_{i=1}^{n} K(x, x_i) v_\ell^i = \frac{1}{\sqrt{n\sigma_\ell}} \mathbf{k}'_{n,x} v_\ell.$$

Finally $T_n f_\ell = \frac{\sigma_\ell}{n}$, f_ℓ, $\langle f_\ell, f_{\ell'} \rangle_{\mathcal{H}} = \delta_{\ell,\ell'}$ leading us to

$$F_n(x) = \sum_{j=1}^{r} \frac{\sigma_\ell}{\sigma_\ell + \lambda} |f_\ell(x)|^2. \tag{6}$$

2.3. Other spectral filters

Equation (6) makes it clear that the regularization parameter λ reduces the contribution of the eigenvectors v_ℓ with small, but non-zero eigenvalue and, hence, ensures the algorithm to be stable against the noise. Clearly other regularized operators P_n can be considered by replacing the Tikhonov filter $\sigma/(\lambda+\sigma)$ with a low-pass (in the frequencies domain) filter $r_\lambda : [0,1] \to [0,1]$. In this case the corresponding decision rule will be given by

$$F_n(x) = \sum_{j=1}^{r} r_\lambda(\sigma_\ell) |f_\ell(x)|^2. \tag{7}$$

The filter is defined only on the unit interval since the normalisation condition on the kernel implies that the eigenvalues of T_n are between 0 and 1. The requirement that $r_\lambda(\sigma)$ is also in $[0,1]$ ensures that also F_n takes values in the unit interval. A discussion about the mathematical assumptions of the filter can be found in Lo Gerfo et al. [16], here we list a few possible examples:

- Truncated SVD: $r_\lambda(\sigma) = \begin{cases} 0 & \sigma < \lambda \\ 1 & \sigma \geq \lambda \end{cases}$
- Spectral cut-off: $r_\lambda(\sigma) = \begin{cases} \frac{\sigma}{\lambda} & \sigma < \lambda \\ 1 & \sigma \geq \lambda \end{cases}$
- Landweber: $r_\lambda(\sigma) = \sigma \sum_{k=0}^{m} (1-\sigma)^k$.

Kernel PCA (KPCA) corresponds to the Trunctated SVD (TSVD) filter, but in KPCA the Gram matrix is usually replaced by covariance matrix in the feature space, see Hoffmann [13]. Notice that, in our setting the separating property justifies the SSE algorithm from a geometrical point of view therefore we do not need to have data with zero mean and thus, unlike KPCA, the normalization of the data is not required.

We conclude by observing that, from the implementation point of view, the adoption of a different filter simply amounts at changing line 3 of Algorithm 1.

2.4. Computational cost

We discuss the computational cost in the worst case when $\mathbf{K}_n$ is a full matrix. With the Tikhonov filter, for each choice of λ and γ, the complexity is of order n^3. The cost of the SVD is also of order n^3, though the constant is worse, but it provides a solution for all the values of the regularization parameter λ, the so-called *regularization path*. This means that, to the price of a single SVD, we obtain different solutions for different λ (see Eq. 6). Also, from Eq. (7) we notice that the same complexity holds true for an arbitrary filter.

Moreover for TSVD, one needs to compute only the eigenvectors whose eigenvalues are bigger than λ, and one can use some approximation probabilistic algorithms, see for example Halko et al. [18], to further reduce the complexity. Instead, for the Landweber filter, an easy computation shows that

$$F_n(x) = \mathbf{k}'_{n,x}\mathbf{a}(x),$$

where the coefficient vector $\mathbf{a}$ can be evaluated iteratively by setting $\mathbf{a}^0(x) = 0$, and

$$\mathbf{a}^\ell(x) = \mathbf{a}^{\ell-1}(x) + \frac{1}{n}(\mathbf{k}_{n,x} - \mathbf{K}_n\mathbf{a}^{\ell-1}(x))$$

for $\ell = 1, \dots, m$, so that the complexity is of the order $n^2 m$, where m is the number of iterations.

2.5. Final remark on the separating property

We conclude the section with a remark on the need for the separating property of a kernel. Eq. (5) and the definition of C_n are meaningful also if a reproducing kernel K does not have the separating property. In this latter case, C_n is a consistent estimator of $\overline{\text{span}}\{\Phi(x) \mid x \in C\} \cap \Phi(\mathbb{R}^d)$, which is in

general bigger than C, unless C is a *good set* for K. For example, if K is the linear kernel, the SSE algorithm is consistent only if C is a linear subspace of $\mathbb{R}^d$. Hence the separating kernels play the same role as the universal kernels in supervised learning (however the two notions are not equivalent, since there exist universal kernels, which are not separating as the gaussian kernel). Next section will explicitly show this feature when adopting a linear kernel on a non linear subspace.

3. The parameters choice

In this section we give a qualitative evaluation on synthetic data of the effect of parameters choice on the SSE algorithm.

The SSE algorithm described in the previous section depends on two parameters: the regularization parameter λ and the threshold τ. Also we have the parameters of the kernel, for instance γ the width of the Laplacian kernel. We first note that, if we set $\lambda = \tau = 0$, the separability property implies that $C_n = \{x_1, \dots, x_n\}$, so that we are over-fitting the data. On the contrary, if λ goes infinity, $F_n(x)$ becomes equal to 0 so that C_n is empty or $\mathbb{R}^n$ (depending on the choice of the parameter τ) and we are over-smoothing the data, as it happens, if τ goes to 1 since $F_n(x) \geq 0$.

In general, λ and τ are connected. For large λ we obtain smooth $F_n(x)$ taking values in a small sub-set of $[0, 1]$ close to 0. In this case, for a wide range of τ we obtain an empty estimate of the support, thus the choice of τ is critical to obtain a non-empty estimate. Conversely, for small λ $F_n(x)$ will be oscillating and will take values in the whole range $[0, 1]$. In this case different τ produce very different non-empty support. In this case it is critical to choose a solution among the ones available.

In the following we discuss the role of these parameters on two meaningful simulations of a linear and a non linear subspace C. To do so, we apply the SSE algorithm with the Tikhonov filter clarifying the meaning of the parameters involved and their effect on the estimation of a support and discuss the effect of choosing between a linear and a non linear kernel in both cases.

As a separable non-linear kernel we choose the Laplacian kernel given by Eq. (1). In this case we notice how λ and γ are again tightly connected and play similar roles.

We consider a space $X = [0, 1]^2$ and a probability density ρ the support of which is a parametric curve

$$X_\rho = \{(x^1, x^2) \in [0, 1]^2 | x^1 = x^1(t) \quad x^2 = x^2(t), t \in I\}$$

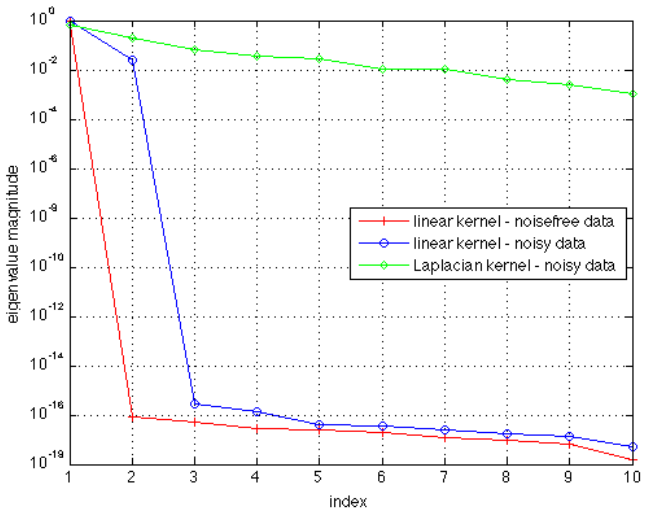

Figure 1: *Support estimation of a linear segment.* Decay of the eigenvalues of the Gram matrix with linear and Laplacian kernels.

where I is a compact interval sampled with a uniform distribution. The training set $S = \{x_1, \ldots, x_n\}$ is i.i.d. according to ρ. Additive uniform noise may be added to the data to model a more realistic situation where the measured data do not always belong exactly to the support.

3.1. Linear support

First we consider a training set $S = \{x_1, \ldots, x_n\}$ $(n = 10)$ sampled on a probability distribution ρ the support of which is a segment, more precisely the unit segment on the x-axis,

$$X_\rho = \{(x^1(t), x^2(t)) \mid t \in [0,1]\} \quad \text{with} \quad \begin{cases} x^1(t) = t \\ x^2(t) = 0 \end{cases}$$

With a normalized linear kernel $K(x,t) = x't/\sqrt{x'x\,t't}$ we obtain a Gram matrix $\mathbf{K}_n$ where each entry $K_{ij} = \cos\theta_{ij}$ and θ_{ij} is the angle between x_i and x_j. If data belong exactly to the segment, then $\cos\theta_{ij} = 1$ for each i and j, and the Gram matrix will have only one non-zero eigenvalue $\sigma_1 = n$; thus $\mathbf{K}_n/n$ will have $\sigma_1 = 1$ has it can be noticed in Figure 1. We observe the highest eigenvalue equal to 1, all the others are negligible. In the absence of

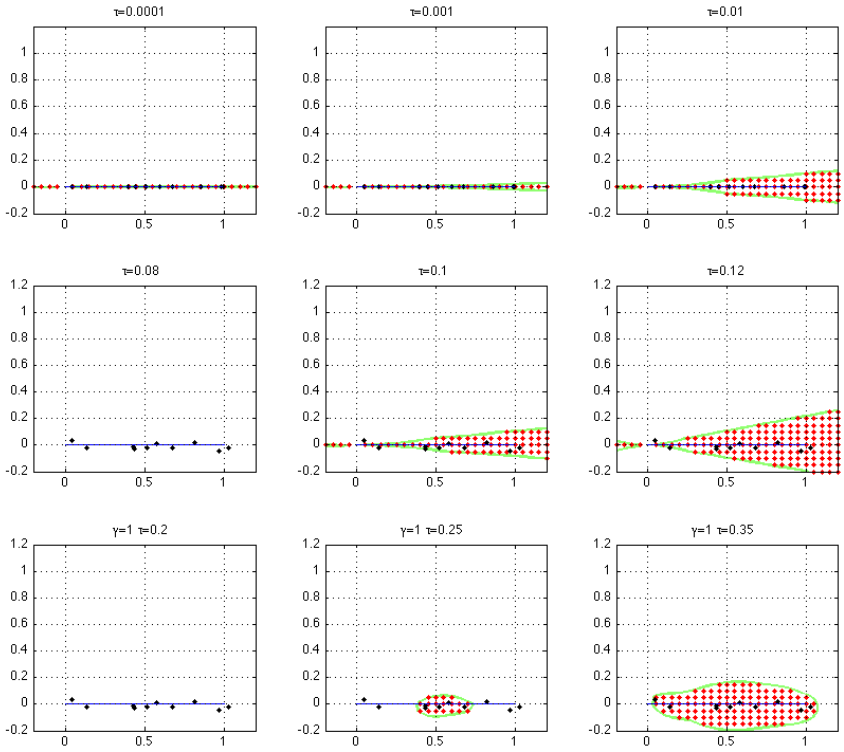

Figure 2: *Support estimation of a linear segment.* The set C_n for different values of τ. Row I: linear kernel and noise-less data. Row II: linear kernel with noisy data. Row III: Laplacian kernel with noisy data.

noise we do not need to add any regularization to the SSE algorithm, thus $\lambda = 0$. Figure 2 (row I) shows different supports that we have estimated by changing the offset value of τ: in the figure the green lines are the boundaries of the estimated support. Red dots are an even sampling of the space X within the estimated support. With $\tau = 10^{-4}$ we obtain a good estimate of the support, where a segment is approximated with a line. As τ increases, regions not belonging to the support are included in the estimate.

In the presence of additive noise on the data sampled from ρ, such data will not be exactly on the ideal support. We perturbate data with additive noise: $\hat{x}_i = x_i + \epsilon\xi$ with ξ uniformly distributed in $[-1, 1]$. The spectrum of the Gram matrix will change, as shown in Figure 1: here we have two eigenvalues that are not negligible and two eigenvectors thanks to which the surface $F_n(x)$ corresponds to a hyperplane. In this case it is not possible to approximate X_ρ regardless the choice of τ. We then look for a regularized solution, $\lambda > 0$. With $\lambda = 0.1$ we attenuate the effect of noise on the previously discussed case and obtain a good approximation (see Figure 2, row II, the estimate obtained with $\tau = 0.1$). Notice the shrinking effect of regularization on the estimated surface $F_n(x)$: in this case, for $\tau = 0.08$ we still obtain an empty support.

We conclude this section by analysing the results obtained with a non-linear Laplacian kernel; in this case we have a further parameter to take into account, γ. In these experiments we choose $\gamma = 1$ as an appropriate choice for points lying on a simple surface in the space $X = [0, 1]^2$. With a Laplacian kernel we consider a more complex set of functions in our approximation. This is suggested by the slower decay of the Gram matrix eigenvalues (see again Figure 1): in this case 7 eigenvalues out of 10 are above 10^{-2}. The non linearity of the Laplacian kernel provides a higher flexibility, but requires a higher number of samples to achieve a good approximation. FigureFigure 2 (row III) shows that different choices of τ fail to approximate the support satisfactorily.

3.2. Lissajous curve support

Now we consider a probability density ρ with a Lissajous curve as a support X_ρ:

$$X_\rho = \{(x^1(t), x^2(t)) \,|\, t \in [0, 2\pi]\} \quad \text{with} \quad \begin{cases} x^1(t) = \frac{1}{2}(\cos(t + \frac{\pi}{2}) + 1) \\ x^2(t) = \frac{1}{2}(\sin(2t) + 1) \end{cases}$$

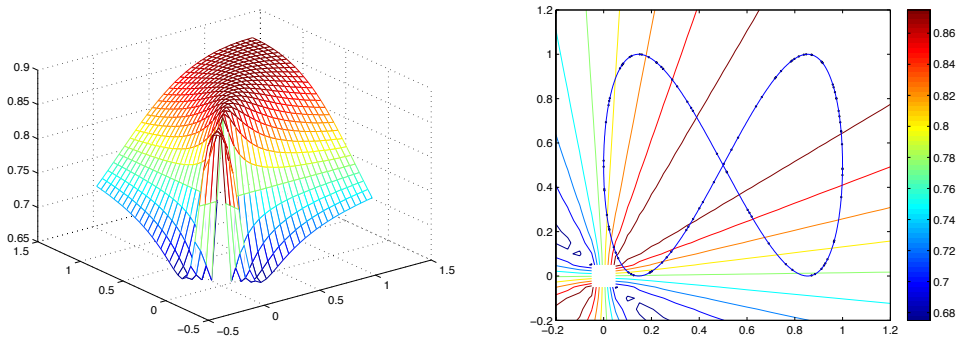

Figure 3: *Curve with linear kernel.* Left: graph of F_n. Right: level sets of F_n. The plots show how the estimated surface is not appropriate for the support considered.

In this case it is apparent a linear kernel, without the separating property, will not estimate accurately a non linear support regardless the parameter choice, as it can be seen in Figure 3. The Laplacian kernel is a more appropriate choice, but in this case parameter estimation also involves the kernel width γ. In what follows we discuss how different choices of the parameter γ affect the estimated surface. To this purpose we set $\lambda = 10^{-3}$ for all experiments.

For *large values of* γ the Gram matrix will become a matrix of 1s with only one eigenvalue greater than 0. With $\gamma = 10$ the estimated $F_n(x)$ is very smooth and close to 1. This can be appreciated in Figure 4 (row I): small changes on τ produce very different solutions, making the choice of an appropriate offset very difficult.

For *small values of* γ the obtained Gram matrix is very sparse. In this case it is easier to set a reasonable τ, although the obtained estimates are very tight on the training data and are prone to overfitting. Figure 4 (row II) shows the estimate obtained for different values of τ and a small $\gamma = 0.05$. The offsets τ are large since most of the estimated support has small values, close to 0.

The choice of an appropriate value of γ may be guided from the data. We choose to estimate the median of median values of the distance of a given point from all the others. The obtained results with the estimated $\gamma = 0.67$ are shown again in Figure 4 (row III) This intermediate γ appears to be appropriate even in the presence of noisy data (Figure 4 (row IV)). Indeed, a

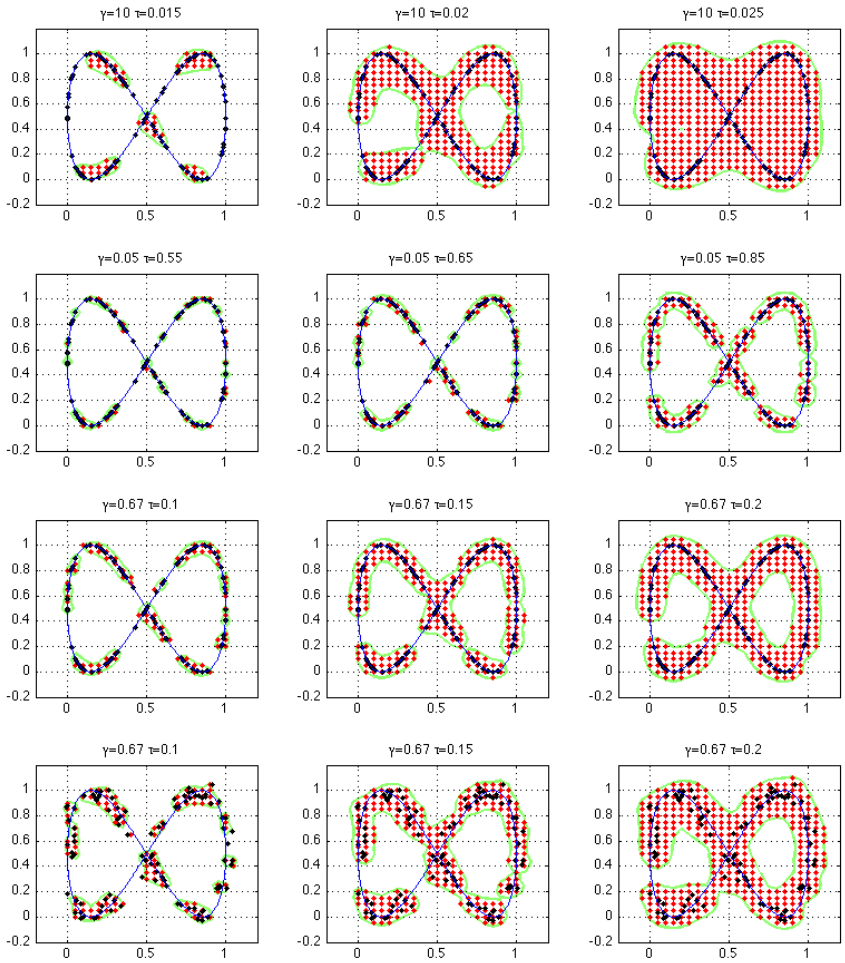

Figure 4: *Support estimation of a curve with a Laplacian kernel.* The set C_n for different values of τ. Row I: large γ. Row II: small γ. Row III: appropriate γ. Row IV: noisy data.

nonlinear kernel with an appropriate choice of parameters produce estimates which are not influenced by the presence of moderate noise.

4. Real datasets

In this section we carry out a thorough experimental analysis on a selection of benchmark datasets of different size (n), dimension of the environment space (d) and nature. Lacking specific benchmarks for support estimation, we consider the closely related novelty detection problem, and start from benchmark multi-class datasets *learning one class at a time*.

Table 1 summarizes the characteristics of each dataset and highlights the variety of the different application settings considered: the MNIST [1] and the USPS [2] datasets, widely used benchmarks on handwritten digits; the COIL dataset [3], an image library for view-based object recognition; BIO, set of data from molecular biology (see Ray et al. [19]). Finally, a selection of different datasets from the UCI benchmark [4].

Table 1: The datasets with their sizes and how they have been used in the experiments: number of trials and percentage of data assigned to training, validation, and test in each experimental section.

Dataset	class. #	samples #	dimension	Training (pos)	Validation (pos - neg)	Test (pos - neg)
MNIST	10	60.000	28×28	1/2	1/3	1/6
COIL	20	1.440	128×128	1/3	1/6	1/2
BIO	2	176	120	3/8	1/8	1/2
USPS	10	10.000	16×16	1/3	1/3	1/3
UCI-Cancer	2	569	32	1/3	1/6	1/2
UCI-Heart	9	303	75	1/3	1/6	1/2
UCI-Telesc.	2	19.020	11	3/8	1/8	1/2
UCI-RedW.	6	1.599	12	1/3	1/6	1/2
UCI-Madel.	2	4.000	500	1/3	1/6	1/2

[1] http://yann.lecun.com/exdb/mnist/
[2] http://www.gaussianprocess.org/gpml/data/
[3] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php
[4] http://archive.ics.uci.edu/ml/datasets.html

We compare the SSE algorithm with well known methods from the literature, in particular Parzen windows [14], one class Support Vector Machines [4], and kernel PCA for novelty detection [13], and K Nearest Neighboors.

In all the experiments we adopt the Laplacian kernel given by Eq. (1) as a non linear separable kernel which we expect to be appropriate for a variety of problems, similarly to the Gaussian kernel for the supervised learning domain. As for the filter we adopt spectral cut-off, after we observed experimentally that different filters produced comparable results on different datasets.

4.1. Model selection

In this section we consider the problem of choosing appropriate parameters in real scenarios and high dimensional data, when the only knowledge on the problem is the availability of a dataset. It is well known that to date there are no model selection methods for one-class learning. Thus, lacking a more effective method, in our experiments we use a cross-validation scheme, assuming we have a (possibly small) set of examples which do not belong to the support we are estimating, we will use in the validation phase only. In particular, we perform model selection with an optimization of cross validation proposed in Rudi et al. [15]. The method, which is inspired by scale-space theory and statistical learning theory, adaptively learns the error function in the parameters space by sampling and refining the approximation only on the regions of stable minima. This approach has been shown to be more effective and computationally advantageous than classical grid search over the parameter space.

In accordance to a novelty detection setting, we perform the support estimation training phase from a set of positive examples, learning one class at a time. Then, we perform model selection on a validation set of positive and negative examples to the purpose of choosing the hyper-parameters that maximize the accuracy. Finally, we perform tests on both positives and negatives to evaluate the effectiveness of the method to both false positives and false negatives.

In the specific case of the SSE algorithm with a Laplacian kernel and a spectral cut-off filter, model selection accounts for the choice of three optimal parameters: the kernel parameter γ, the regularization parameter λ which is a threshold on the Gram matrix eigenvalues, and the threshold τ on the decision rule.

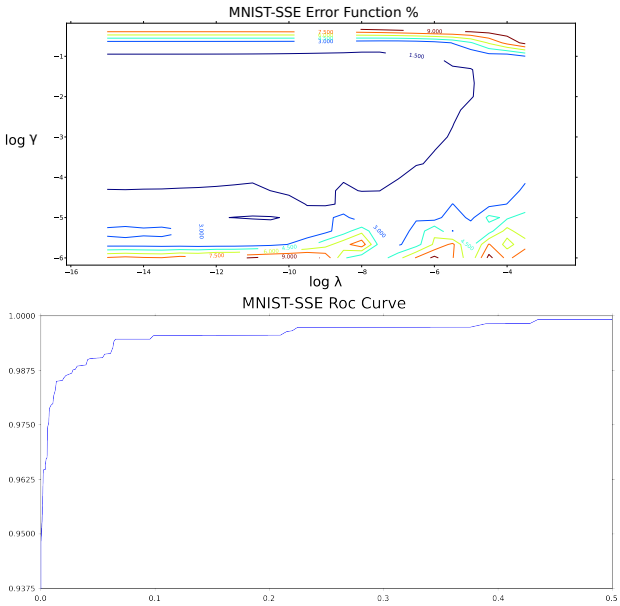

Figure 5: The effect of parameter choice on the error function for the MNIST dataset. Left: error function obtained by varying λ and γ. Right: ROC curve obtained by varying τ (see text)

As we pointed out in Section 3 the different parameters are somewhat connected. Figure 5 shows the effect of changing the parameters value on the MNIST dataset. On the top the error function obtained for different λ and γ is reported: for each (λ, γ) pair the error is obtained with the optimal τ. The plot highlights the link between the two parameters: one may obtain a similar result by changing either the λ or the γ value. Also, it can be noticed that the error function is smooth and the minimum error is associated to a wide area, showing that the parameter choice is not critical for the method on this set of data. On the bottom we show the ROC curve computed by varying the threshold τ, on a fixed ($\lambda = 3.8 * 10^{-5}$, $\gamma = 0.144$) pair. The figure shows the curve corresponding to the optimal pair (that is, the pair corresponding to the minimum error).

4.2. Method assessment

For each dataset and each algorithm we performed 100 trials. All the accuracy values concerning each trial are collected and are presented in Figure 6. Since most of the benchmark datasets we adopted are composed of multiple classes, given a dataset, we consider a class at a time and estimate its support from positive examples only. Then we evaluate the membership of each test datum to the support estimated. In this case we have a false positive if an example belonging to one of the other classes is associated to the estimated support, while a false negative if an example of the class considered falls outside the support. The results we report are the average of false positives and false negatives for all the different classes. For each dataset five box-plot are shown, one for each algorithm. Given the dataset and the algorithm, the box plot represents the statistical properties of the classification error, calculated on each trial and each class: on each box the central mark is the median, the edges of each box the 25th and the 75th percentiles, while outliers are represented as separate crosses.

The SSE algorithm consistently exhibits very good and stable performances, and turns out being the best performing algorithm on the MNIST, COIL, BIO, UCI Telescope, UCI Madelon.

5. Discussion

The paper presented an extensive discussion on the computational and geometrical properties of a recently proposed algorithm for Support Estimation, the so called Spectral Support Estimation (SSE) algorithm. The main

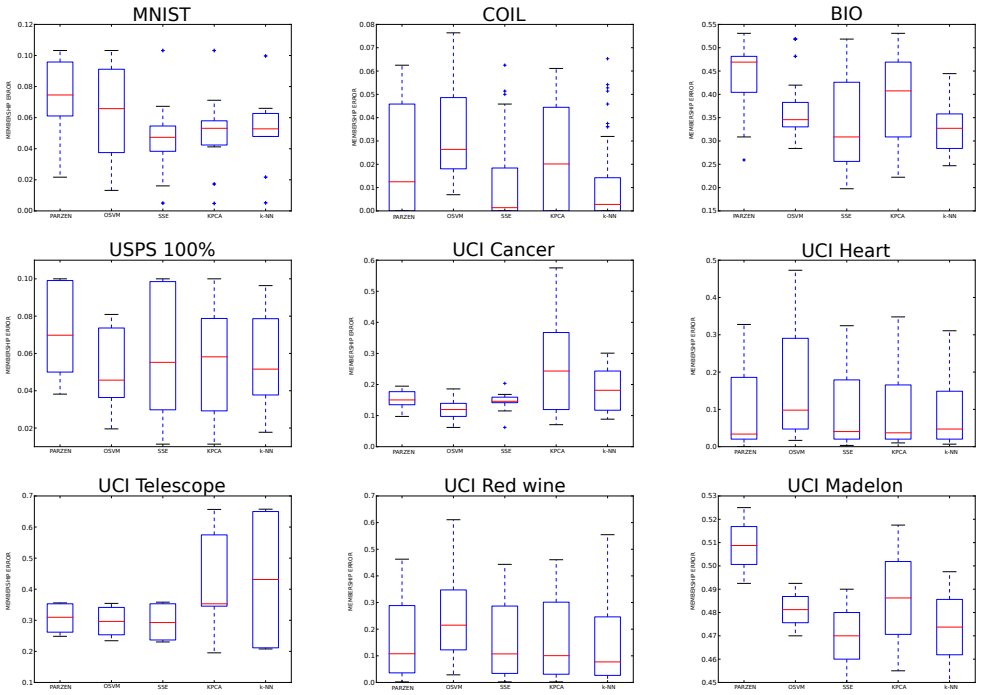

Figure 6: Comparative analysis of the SSE method with other support estimation algorithms on various datasets (see text). The box-plots describe the statistics of the errors of different trials and all the classes of each dataset, whose support is learned one at a time. In red the average error.

aim of the paper was to review the estimator providing a geometrical interpretation, and highlighting the properties of the kernel functions to be adopted. A further aim of the paper was to gain a deeper understanding of its computational aspects, deriving a simple implementation of the algorithm and discussing parameter selection and computational complexity issues. Simulations allowed us to illustrate the relationship between hyper-parameters and the effect of parameter choices. Real data were instead adopted to show the appropriateness of the method as a novelty detection algorithm with respect to its well known competitors (KNN, KPCA, Parzen windows, 1C-SVM) on a large selection of problems of different size and nature.

References

[1] E. De Vito, L. Rosasco, A. Toigo, Spectral Regularization for Support Estimation, Advances in Neural Information Processing Systems, NIPS Foundation (2010) 1–9.

[2] A. Rényi, R. Sulanke, Über die konvexe Hülle von n zufällig gewählten Punkten, Probability Theory and Related Fields 2 (1) (1963) 75–84.

[3] J. Geffroy, Sur un Probleme d'estimation Géométrique, Publ. Inst. Statist. Univ. Paris 13 (1964) 191–210.

[4] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

[5] M. Markou, S. Singh, Novelty detection: a review–part 1: statistical approaches, Signal Processing 83 (12) (2003) 2481–2497.

[6] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys (CSUR) 41 (3) (2009) 15.

[7] L. Dümbgen, G. Walther, Rates of convergence for random approximations of convex sets, Adv. in Appl. Probab. 28 (2) (1996) 384–393, ISSN 0001-8678.

[8] L. Devroye, G. Wise, Detection of abnormal behavior via nonparametric estimation of the support, SIAM Journal on Applied Mathematics (1980) 480–488.

[9] A. Cuevas, R. Fraiman, A plug-in approach to support estimation, Ann. Statist. 25 (6) (1997) 2300–2312, ISSN 0090-5364.

[10] D. L. Donoho, C. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, Proc. Natl. Acad. Sci. USA 100 (10) (2003) 5591–5596 (electronic), ISSN 1091-6490.

[11] M. Belkin, P. Niyogi, V. Sindhwani, Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples, J. Mach. Learn. Res. 7 (2006) 2399–2434, ISSN 1532-4435.

[12] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10 (5) (1998) 1299–1319.

[13] H. Hoffmann, Kernel PCA for novelty detection, Pattern Recognition 40 (3) (2007) 863–874.

[14] L. Tarassenko, P. Hayton, N. Cerneaz, M. Brady, Novelty detection for the identification of masses in mammograms, in: Artificial Neural Networks, 1995., Fourth International Conference on, IET, 442–447, 1995.

[15] A. Rudi, G. Chiusano, A. Verri, Adaptive Optimization for Cross Validation, In Proceedings of ESANN 2012, in: The 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - Proceedings, ESANN, 442–447, 2012.

[16] L. Lo Gerfo, L. Rosasco, F. Odone, E. De Vito, A. Verri, Spectral algorithms for supervised learning, Neural Comput. 20 (7) (2008) 1873–1897.

[17] I. Steinwart, A. Christmann, Support vector machines, Information Science and Statistics, Springer, New York, ISBN 978-0-387-77241-7, 2008.

[18] N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288, ISSN 0036-1445.

[19] S. Ray, B. M, C. Herbert, Y. Takeda-Uchimura, A. Boxer, K. Blennow, L. F. Friedman, Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins, Nature Medicine 13 (2007) 1359 – 1362.