UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

OPERATORS FOR TRANSFORMING KERNELS INTO QUASI-LOCAL KERNELS THAT IMPROVE SVM ACCURACY

Nicola Segata and Enrico Blanzieri

July 2009

Technical Report # DISI-09-042

Note: updated version of DISI Technical Report DISI-08-009

Operators for transforming kernels into quasi-local kernels that improve SVM accuracy

Nicola Segata and Enrico Blanzieri *

June 24, 2009

Abstract

Motivated by the crucial role that locality plays in various learning approaches, we present, in the framework of kernel machines for classification, a novel family of operators on kernels able to integrate local information into any kernel obtaining *quasi-local* kernels. The quasi-local kernels maintain the possibly global properties of the input kernel and they increase the kernel value as the points get closer in the feature space of the input kernel, mixing the effect of the input kernel with a kernel which is local in the feature space of the input one. If applied on a local kernel the operators introduce an additional level of locality equivalent to use a local kernel with non-stationary kernel width. The operators accept two parameters that regulate the width of the exponential influence of points in the locality-dependent component and the balancing between the feature-space local component and the input kernel. We address the choice of these parameters with a data-dependent strategy. Experiments carried out with SVM applying the operators on traditional kernel functions on a total of 43 datasets with different characteristics and application domains, achieve very good results supported by statistical significance.

1 Introduction

Support Vector Machines [1] (SVM) are state-of-the-art classifiers and are now widely used and applied over a wide range of domains. Reasons for SVM's success are multiple, including the presence of an elegant bound on generalization error [2], the fact that SVM is based on kernel functions $k(\cdot, \cdot)$ representing the scalar product of the sample mapped in a Hilbert space and the relative lightweight computational cost of the model in the evaluation phase. For a review on SVM and kernel methods the reader can refer to [3].

Locality in classification plays a crucial role [4]. In the framework of statistical learning theory, in fact, selecting the local influence of the training points used to classify a test point (i.e. the level of locality of the classifier), allows one to find a lower minimization of the guaranteed risk (i.e. a bound on the probability of classification error) with respect to "global" approaches as shown in [5]. Local learning algorithms [4, 6] are based on this theoretical consideration and they locally adjust the separating surface considering the characteristics of each region of the training set, the assumption being that the class of a test point can be more precisely determined by the local neighbors rather than by the whole training set especially for non-evenly distributed datasets. Notice that one of the most popular classification methods,

*N. Segata and E. Blanzieri are with Department of Information and Telecommunication Technologies, University of Trento, Italy. E-Mail: {segata, blanzier}@dit.unitn.it.

the K-Nearest Neighbors (KNN)[1] algorithm, is deeply based on the notion of locality. In kernel methods, locality has been introduced with two meanings: i) as local relationship between the features, i.e. local feature dependence, adding prior information reflecting it, ii) as distance proximity between points, i.e. local points dependence, enhancing the kernel values for points that are close to each other and/or penalizing the points that are far from each other. The first meaning has been exploited by *locality-improved kernels*, the second by *local kernels* and *local SVM*.

Locality-improved kernels [3] take into account prior knowledge of the local structure in data such as local correlation between pixels in images. The way the prior information is integrated into the kernel depends on the specific task but, in general, the kernel increases similarity and correlation of selected features that are considered locally related. Locality-improved kernels were successfully applied on image processing [7] and on bioinformatics tasks [8] [9].

Local kernels are kernels such that, when the distance between a test point and a training point tends to infinity, the value of the kernel is constant and independent of the test point [10] [11]; if this condition is not respected the kernel is said to be *global*. A popular local kernel is the radial basis function (RBF) kernel that tends to zero for points whose distance is high with respect to a width parameter that regulates the degree of locality. On the other hand, distant points influence the value of global kernels (e.g. linear, polynomial and sigmoidal kernels). Local kernels and in particular the RBF kernel show very good classification capability but they can suffer from the curse of dimensionality problem [12] and they can fail with datasets that require non-linear long-range extrapolation. In this case, even if the tuning of the width parameter allows for the contribution of distant points, global kernel reflecting a particular conformation of the separating surface are generally preferred and permits better accuracies. An attempt to mix the good characteristics of local and global kernels is reported in [11] where RBF and polynomial kernels are considered for SVM regression.

Local SVM is a local learning algorithm and was independently proposed by Blanzieri and Melgani [13] [14] and by Zhang et al. [15] and applied respectively to remote sensing and visual recognition tasks. Other successful applications of the approach are detailed in [16] for general real datasets, in [17] for spam filtering and in [18] for noise reduction. The main idea of local SVM is to build at prediction time a sample-specific maximal marginal hyperplane based on the set of K-neighbors. In [13] it is also proved that the local SVM has chance to have a better bound on generalization with respect to SVM. However, local SVM suffers from the high computational cost of the testing phase that comprises for each sample the selection of the K nearest neighbors and the computation of the maximal separating hyperplane, and from the problem of tuning the K parameter. Although the first drawback prevents the scalability of the method for large datasets, some approximations of the approach have been proposed in order to improve the computational performances in [19] and [20]. In particular the approach we presented in [20] is asymptotically faster than SVM especially for non high-dimensional datasets basically maintaining the classification capabilities of KNNSVM, whereas the approach of [19] remains much slower than SVM and builds only local linear models.

Other ways of including locality in the learning process are based on the work of Amari and Wu [21] that modify the Riemannian geometry induced by the kernel in the input space introducing a quasi-conformal transformation on the kernel metric with a positive scalar function. Particular choices of such scalar functions permitted in [21] to increasing the margin of the separating hyperplane through a two steps SVM training under the empirical assumption that the support vectors (detected with a primary SVM training) are located mainly in proximity of the hyperplane. In the bioinformatics field, a different particular choice of the scalar function

[1] From now on, for notational reasons, we refer to the K parameter of KNN based methods with upper-case K, reserving lower-case k for denoting kernel functions.

permitted to the authors of [22] to reach high accuracy in classification of tissue samples from their microarray gene expression levels through a KNN based scheme. Locality has been also used as the key factor to combine multiple kernel functions using a non-stationary (i.e. non-global) fashion as detailed in [23].

In this work we present a family of operators that transform an arbitrary input kernel into a kernel which has a component that is local and universal in the feature space of the input kernel. This resulting new family of kernels, opportunely tuned, maintains the original kernel behaviour for non-local regions, while increasing the values of the kernel for pairs of points that fall in a local region. In this way we aim to take advantage of both locality information and the long-range extrapolation ability of global kernels, alleviating also the curse of dimensionality problem of the local kernels and balancing the compromise between interpolation and generalization capability. The operators systematically map the input kernel functions into kernels that maintain the positive definite property and exploit the locality in the feature space which is a generalization of the standard locality meaning and it is central in the notion of quasi-local kernels. In such a way we are able to introduce the power of local learning techniques in the standard kernel methods framework modifying only the kernel functions and thus overcoming the computational limitation of the original formulation of local SVM. In particular, if the operators are applied on a local kernel, it turns out that the new kernel has a conceptually different meaning of locality, basically similar to a local kernel with variable kernel width. We give a practical way of estimating the optimal additional parameters introduced in the resulting kernel functions starting from the optimized input kernel and the penalty parameter of SVM.

Although we are focusing here on the classification task, our operators on kernels can be theoretically applied for every kernel-based technique in which locality plays a crucial role. It is the case of many kernel-based subspace analysis techniques like dimensionality reduction, manifold learning and feature selection techniques which are gaining importance in the last few years. Some of the most popular techniques are intrinsically based on locality such as Local Learning Embedding (LLE) [24] which has a kernel-based version [25] and it is equivalent to kernel principal component analysis (kernel PCA) [26] for a particular kernel choice and kernel Local Discriminant Embedding (kernel LDE) [27]. Other non naturally local techniques, have their local counterparts: Fisher Discriminative Analysis (FDA) [28] and its kernel-based version [29] with Local Fisher Discriminative Analysis (LFDA) [30], Generalized Discriminant Analysis (GDA) [31] with locally linear discriminant analysis (LLDA) [32]. Global techniques such as ISOMAP [33,34] can adopt their kernel version using a local kernel to include locality. Other approaches are based on developing and learning kernels subject to local constraints, as for example in [35]. An interesting discussion on local and global approaches for non-linear dimensionality reduction fall beyond the kernel methods field and it is addressed in [36].

The paper is organized as follows. After recalling in section 2 some preliminaries on SVM, kernel functions and local SVM, in section 3 we present the new family of operators that produces quasi-local kernels. The artificial example presented in section 4 illustrates intuitively how the quasi-local kernels work. In section 5 we propose a first experiment on 23 datasets with the double purpose of investigating the classification performance and of identifying the most suitable quasi-local operators. The most promising quasi-local kernels are applied in the experiment of section 6 to 20 large classification datasets. Finally, in section 7, we draw some conclusions.

2 SVM and kernel methods preliminaries

Support vector machines (SVMs) are classifiers with sound foundations in statistical learning theory [2]. The decision rule of an SVM is $\mathrm{SVM}(x) = \mathrm{sign}(\langle w, \Phi(x)\rangle_{\mathcal{F}} + b)$ where $\Phi(x) : \mathbb{R}^p \to \mathcal{F}$ is a mapping in some transformed feature space $\mathcal{F}$ with inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$. The parameters $w \in \mathcal{F}$ and $b \in \mathbb{R}$ are such that they minimize an upper bound on the expected risk while minimizing the empirical risk. The minimization of the complexity term is achieved by minimizing the quantity $\frac{1}{2} \cdot \|w\|^2$, which is equivalent to maximizing the margin between the classes. The empirical risk term is controlled through the following set of constraints:

$$y_i \left(\langle w, \Phi(x_i)\rangle_{\mathcal{F}} + b\right) \geq 1 - \xi_i \quad \text{with } \xi_i \geq 0 \text{ and } i = 1, \ldots, N \tag{1}$$

where $y_i \in \{-1, +1\}$ is the class label of the i-th nearest training sample. Such constraints mean that all points need to be either on the borders of the maximum margin separating hyperplane or beyond them. The margin is required to be 1 by a normalization of distances. The presence of the slack variables ξ_i allows the search for a soft margin, i.e. a separation with possibly some training set misclassification, necessary to handle noisy data and non-completely separable classes. By reformulating such an optimization problem with Lagrange multipliers α_i $(i = 1, \ldots, N)$, and introducing a positive definite kernel function $k(\cdot,\cdot)$ that substitutes the scalar product in the feature space $\langle\Phi(x_i), \Phi(x)\rangle_{\mathcal{F}}$ the decision rule can be expressed as:

$$\mathrm{SVM}(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b\right)$$

where training points with nonzero Lagrange multipliers are called support vectors. The introduction of the positive definite (PD) kernels avoids the explicit definition of the feature space $\mathcal{F}$ and of the mapping Φ [3] [37], through the so-called kernel trick. A kernel is PD if it is the scalar product in some Hilbert space, i.e. the kernel matrix is symmetric and positive definite[2].

The maximal separating hyperplane defined by SVM has been shown to have important generalization properties and nice bound on the VC dimension [2]. In particular we refer to the following theorem:

Theorem 1 (Vapnik [2] p.139)**.** *The expectation of the probability of test error for a maximal separating hyperplane is bounded by*

$$EP_{error} \leq E\left\{\min\left(\frac{m}{l}, \frac{1}{l}\left[\frac{R^2}{\Delta^2}\right], \frac{p}{l}\right)\right\}$$

where l is the cardinality of the training set, m is the number of support vectors, R is the radius of the sphere containing all the samples, $\Delta = 1/|w|$ is the margin, and p is the dimensionality of the input space.

Theorem 1 states that the maximal separating hyperplane can generalize well as the expectation on the margin is large (since a large margin minimizes the $\frac{R^2}{\Delta^2}$ ratio).

2.1 Local and global basic kernels

Kernel functions can be divided in two classes: local and global kernels [11]. Following [10] we define the locality of a kernel as:

[2] In the present work, we frequently refer to PD kernels simply as kernels.

Definition 1 (Local kernel)**.** *A PD kernel k is a* local kernel *if, considering a test point x and a training point x_i, we have that*

$$\lim_{\|x-x_i\|\to\infty} k(x, x_i) \to c_i \tag{2}$$

with c_i constant and not depending on x. If a kernel is not local, it is considered to be global*.*

This definition captures the intuition that, in a local kernel, only the points that are enough close each other influences the kernel value. This does not directly implicate that the higher peak of the kernel value is in correspondence of points in the same position, although the most popular local kernel functions have this additional characteristic. In contrast, in a global kernel function, all the points are able to influence the kernel value regardless of their proximity.

In this work we will consider as baseline and as inputs of the operators we will introduce in the next section, the linear kernel k^{lin}, the polynomial kernel k^{pol}, the radial basis function kernel k^{rbf} and the sigmoidal kernel k^{sig}. We refer to these four kernels as *reference input kernels* and we recall here their definitions:

$$\begin{aligned} k^{lin}(x,x') &= \langle x, x'\rangle & \qquad k^{pol}(x,x') &= (\gamma^{pol}\cdot\langle x, x'\rangle + r^{pol})^d \\ k^{rbf}(x,x') &= \exp(-\gamma^{rbf}\cdot||x-x'||^2) & \qquad k^{sig}(x,x') &= \tanh(\gamma^{sig}\cdot\langle x, x'\rangle + r^{sig}) \end{aligned}$$

with $\gamma^{pol}, \gamma^{rbf}, \gamma^{sig} > 0,\ r^{pol}, r^{sig} \geq 0$ and $d \in \mathbb{N}$.

It is simple to show that, among the four input kernel listed above, the only local kernel is k^{rbf} since for $\|x - x_i\| \to \infty$ we have that $k^{rbf}(x, x_i) \to 0$ (i.e. a constant that does not depend on x), whereas k^{lin}, k^{pol} and k^{sig} are global.

For the radial basis function kernel k^{rbf} it is reasonable to set the parameter γ^{rbf} with the inverse of the squared median of the of $\|x_i - x_j\|$, namely the Euclidean distances between every pair of samples x_i [38]. This because $k^{rbf}(x, x')$ can be written explicitly introducing the kernel width as $\exp\left(-\frac{||x-x'||^2}{2\cdot\sigma^{rbf\,2}}\right)$ and in this way the distances are weighted with a value that is likely to be in same order of magnitude. More precisely, denoting with $q_h[\|x - x'\|^{\mathcal{Z}}]$ the h percentile of the distribution of the distance in the $\mathcal{Z}$ space between every pair of points x, x' in the training set, γ^{rbf} can be chosen as $\gamma^{rbf}_h = 1/(2 \cdot q_h^2[\|x - x'\|^{\mathbb{R}^p}])$. For h reasonable choices can be 10, 50 (i.e. the median) or 90 that should be in the same order of magnitude of the median, and 1 which enhances the local behaviour.

It is known that the linear, polynomial and radial basis function kernels are valid kernels since they are PD. It has been shown, however, that the sigmoidal kernel is not PD [3]; nevertheless it has been successfully applied in a wide range of domains as discussed in [39]. In [40] is showed that the sigmoidal kernel can be conditionally positive definite (CPD) for certain parameters and for specific inputs. Since CPD kernels can be safely used for SVM classification [41], the sigmoidal kernel is suitable for SVM only on a subset of the parameters and input space. In this work we use the sigmoidal kernel being aware of its theoretical limitations, which can be reflected in non-optimal solutions and convergence problems in the maximal margin optimization.

2.2 Local SVM

The method [13, 14] combines locality and searches for a large margin separating surface by partitioning the entire transformed feature space through an ensemble of local maximal margin hyperplanes. It can be seen as a modification of the SVM approach in order to obtain a local learning algorithm [4, 5] able to locally adjust the capacity of the training systems. The local learning approach is particularly effective for uneven distributions of training set samples in the input space. Although KNN is the simplest local learning algorithm, its decision rule

based on majority voting overlooks the geometric configuration of the neighbourhood. For this reason the adoption of a maximal margin principle for neighbourhood partitioning can result in a good compromise between capacity and number of training samples [42].

In order to classify a given point x' of the p-dimensional input feature space, we need first to find its K nearest neighbors in the transformed feature space $\mathcal{F}$ and, then, to search for an optimal separating hyperplane only over these K nearest neighbors. In practice, this means that an SVM classifier is built over the neighborhood of each test point x'. Accordingly, the constraints in (1) become:

$$y_{r_x(i)} \left(w \cdot \Phi(x_{r_x(i)}) + b \right) \geq 1 - \xi_{r_x(i)}, \text{ with } i = 1, \ldots, K$$

where $r_{x'} : \{1, \ldots, N\} \rightarrow \{1, \ldots, N\}$ is a function that reorders the indexes of the N training points defined recursively as:

$$\begin{cases} r_{x'}(1) = \underset{i=1,\ldots,N}{\operatorname{argmin}} \|\Phi(x_i) - \Phi(x')\|^2 \\ r_{x'}(j) = \underset{i=1,\ldots,N}{\operatorname{argmin}} \|\Phi(x_i) - \Phi(x')\|^2 \quad \text{with } i \neq r_{x'}(1), \ldots, r_{x'}(j-1) \text{ for } j = 2, \ldots, N \end{cases}$$

In this way, $x_{r_{x'}(j)}$ is the point of the set X in the j-th position in terms of distance from x' and the following holds: $j < K \Rightarrow \|\Phi(x_{r_{x'}(j)}) - \Phi(x')\| \leq \|\Phi(x_{r_{x'}(K)}) - \Phi(x')\|$ because of the monotonicity of the quadratic operator. The computation of the distance in $\mathcal{F}$ is expressed in terms of kernels as:

$$\begin{aligned} & ||\Phi(x) - \Phi(x')||^2 = \Phi^2(x) + \Phi^2(x') - 2\langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}} = \\ = \; & \langle \Phi(x), \Phi(x) \rangle_{\mathcal{F}} + \langle \Phi(x'), \Phi(x') \rangle_{\mathcal{F}} - 2\langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}} = k(x, x) + k(x', x') - 2k(x, x'). \end{aligned} \tag{3}$$

If the kernel is k^{rbf} or any polynomial kernels with degree 1, the ordering function is equivalent to use the Euclidean metric. For non-linear kernels (other than k^{rbf}) the ordering function can be different from that produced using the Euclidean metric.

The decision rule associated with the method is:

$$K\text{NNSVM}(x) = \text{sign} \left(\sum_{i=1}^{K} \alpha_{r_x(i)} y_{r_x(i)} k(x_{r_x(i)}, x) + b \right).$$

For $K = N$, the KNNSVM method is the usual SVM whereas, for $K = 2$, the method implemented with the linear kernel corresponds to the standard 1-NN classifier. Conventionally, in the following, we assume that also 1-NNSVM is equivalent to 1-NN.

The method can be seen as a KNN classifier implemented in the input or in a transformed feature space with a SVM decision rule or as a local SVM classifier. In this second case the bound on the expectation of the probability of test error becomes:

$$EP_{error} \leq E \left\{ \min \left(\frac{m}{K}, \frac{1}{K} \left[\frac{R^2}{\Delta^2} \right], \frac{p}{K} \right) \right\}$$

where m is the number of support vectors. Whereas the SVM has the same bound with $K = N$, apparently the three quantities increase due to $K < N$. However, in the case of KNNSVM the ratio $\frac{R^2}{\Delta^2}$ decreases because: 1) R (in the local case) is smaller than the radius of the sphere that contains all the training points; and 2) the margin Δ increases or at least remains unchanged. The former point is easy to show, while the second point (limited to the case of linear separability) is stated in the following theorem [14].

Theorem 2. *Given a set of N training points $X = \{x_i \in \mathbb{R}^p\}$, each associated with a label $y_i \in \{-1, 1\}$, over which is defined a maximal margin separating hyperplane with margin Δ_X, if for an arbitrary subset $X' \subset X$ there exists a maximal margin hyperplane with margin $\Delta_{X'}$ then the inequality $\Delta_{X'} \geq \Delta_X$ holds.*

Sketch of the proof. Observe that for $X' \subset X$ the convex hull of each class is contained in the convex hull of the same class in X. Since the margin can be seen as the minimum distance between the convex hulls of different classes and since given two convex hulls H_1, H_2 the minimum distance between them cannot be lower than the minimum distance between H_1' and H_2 with $H_1' \subseteq H_1$, we have the thesis. For an alternative and rigourous proof see [14]. □

As a consequence of Theorem 2 the KNNSVM has the potential of improving over both KNN and SVM as empirically shown in [13] for remote sensing, in [15] for visual applications, in [16] on 13 benchmark datasets, in [17] for spam filtering and in [18] for noise removal.

Apart from the SVM parameters (C and the kernel parameters), the only parameter of KNNSVM that needs to be tuned is the number of neighbors K. K can be estimated on the training set among a predefined series of natural numbers (usually a subset of the odd numbers between 1 and the total number of points) choosing the value that shows better predictive accuracy with a 10-fold cross validation approach. In this work, when we refer to the KNNSVM classifier we assume that K is estimated in this way.

3 Operators that transform kernels into quasi-local kernels

In this section we define the operators we use to integrate the locality information into existing kernels obtaining quasi-local kernels. We first introduce the framework of operators on kernel, then the quasi-local operators discussing their properties, definition, intuitive meaning and strategies to select their parameters.

3.1 Operators on kernels

An operator on kernels, generically denoted as $\mathcal{O}$, is a function that accepts a kernel as input and transforms it into another kernel, i.e. $\mathcal{O}$ is an operator on kernels if $\mathcal{O}\,k$ is a kernel (supposing that k is a kernel). More formally:

Definition 2 (Operators on kernels)**.** *Denoting with l_p a (possibly empty) list of parameters that can be real constants and real-valued functions and with l_k a (possibly empty) a-priori fixed-length list of PD kernels, $\mathcal{O}_{l_p}$ is an operator on kernels if $k(x, x') = (\mathcal{O}_{l_p}\, l_k)(x, x')$ with $x, x' \in \mathcal{X}$ is positive definite for every choice of PD kernels in l_k.*

An example of operator with an empty list of kernels that we can define is $(\mathcal{O}_f^{mul})(x, x') := f(x)f(x')$ which is a PD kernel for every real-valued function f. Also the identity function can be thought of as an operator on kernel such that $(\mathcal{I}\,k)(x, x') = k(x, x')$. Another example is the exponentiation operator defined as $(\mathcal{O}^e\,k)(x, x') := \exp(k(x, x'))$. Although the focus in this work is on the class of operators for quasi local kernels, notice that, defining operators based on known properties of kernel, it is possible to prove the PD property of a kernel rewriting it starting from known PD kernels applying only operators on kernels.

3.2 Operators for quasi-local kernels

Our operators produce kernels that we call *quasi-local* kernels, combining the input kernel with another kernel based on the distance in the feature space of the input kernel. The formal definition of quasi-locality will be discussed in subsection 3.6 but basically the class of quasi-local kernels comprises those kernels that combine an input kernel with a kernel which is local in the feature space of the input kernel. In the case of a global kernel as input of the operators, the intuitive effect of the *quasi-locality* of the resulting kernels is that they are not local for definition 1 but at the same time the kernel score is significantly increased for samples that are

close in the feature space of the input kernel. In this way the kernel can take advantage from both the locality in the feature space and the long-range extrapolation ability of the global input kernel.

We first construct a kernel to capture the locality information of any kernel function; such a family of kernels takes inspiration from the RBF kernel, substituting the Euclidean distance with the distance in the feature space.

$$k^{exp}(x, x') = \exp\left(-\frac{||\Phi(x) - \Phi(x')||^2}{\sigma}\right) \quad \sigma > 0$$

where Φ is a mapping between the input space $\mathbb{R}^p$ and the feature space $\mathcal{F}$. The feature space distance $||\Phi(x) - \Phi(x')||^2$ is dependent on the choice of kernel (see (3)):

$$||\Phi(x) - \Phi(x')||^2 = k(x, x) + k(x', x') - 2 \cdot k(x, x').$$

The k^{exp} kernel can be obtained with the first operator, named $\mathcal{E}_\sigma$, that accepts a positive parameter σ applied on a kernel k producing $\mathcal{E}_\sigma\, k = k^{exp}$. Explicitly, the $\mathcal{E}_\sigma$ operator is defined as:

$$(\mathcal{E}_\sigma\, k)(x, x') = \exp\left(\frac{-k(x, x) - k(x', x') + 2k(x, x')}{\sigma}\right) \quad \sigma > 0. \tag{4}$$

Note that $\mathcal{E}_\sigma\, k^{lin} = k^{rbf}$ so as a special case we have the RBF kernel. However, the kernels obtained with $\mathcal{E}_\sigma$ consider only the distance in the feature space without including explicitly the input kernel. For this reason $\mathcal{E}_\sigma\, k$ is not a quasi-local kernel.

In order to overcome the limitation of $\mathcal{E}_\sigma$ which completely drops the global information, the idea is to weight the input kernel with the local information to obtain a real quasi-local kernel. So we include explicitly the input kernel in the output of the following operator:

$$(\mathcal{P}_\sigma\, k)(x, x') = k(x, x') \cdot (\mathcal{E}_\sigma k)(x, x') \qquad \sigma > 0. \tag{5}$$

Observing that the $\mathcal{E}_\sigma\, k$ kernel can assume values only between 0 and 1 (since it is an exponential with negative exponent) and that the higher the distance in the feature space between samples the lower the value of the $\mathcal{E}_\sigma\, k$ kernel, the idea of $\mathcal{P}_\sigma$ is to exponentially penalize the basic kernel k with respect to the feature space distance between x and x'.

An opposite possibility is to amplify the values of input kernels in the cases in which the samples contain local information. This can be done simply by adding the $\mathcal{E}_\sigma\, k$ kernel to the input one.

$$(\mathcal{S}_\sigma\, k)(x, x') = k(x, x') + (\mathcal{E}_\sigma k)(x, x') \qquad \sigma > 0. \tag{6}$$

However, since $\mathcal{E}_\sigma$ gives kernels that can assume at most the value of 1 while the input kernel in the general case does not have an upper bound, it is reasonable to weight the $\mathcal{E}_\sigma$ operator with a constant reflecting the order of magnitude of the values that the input kernel can assume in the training set. We call this parameter η and the new operator is:

$$(\mathcal{S}_{\sigma,\eta}\, k)(x, x') = k(x, x') + \eta \cdot (\mathcal{E}_\sigma k)(x, x') \quad \sigma > 0, \eta \geq 0. \tag{7}$$

A different formulation of the $\mathcal{P}_\sigma$ operator that maintains the product form but adopts the idea of amplifying the local information is:

$$(\mathcal{PS}_\sigma\, k)(x, x') = k(x, x') \left[1 + (\mathcal{E}_\sigma k)(x, x')\right] \quad \sigma > 0, \eta \geq 0. \tag{8}$$

Also in this case the parameter η that controls the weight of the $\mathcal{E}_\sigma\, k$ kernel is introduced:

$$(\mathcal{PS}_{\sigma,\eta}\, k)(x, x') = k(x, x') \left[1 + \eta \cdot (\mathcal{E}_\sigma k)(x, x')\right] \quad \sigma > 0, \eta \geq 0. \tag{9}$$

The quasi-local kernels produced by the operators defined in Eq. 5 6, 7, 8, 9 are more complicated then the corresponding input kernels, since it is necessary to evaluate $k(x,x)$, $k(x',x')$, $k(x,x')$ and to perform a couple of addition/multiplication operation and an exponentiation instead of the evaluation of $k(x,x')$ only. However, this is a constant computational overhead in the kernel evaluation phase, that does not affect the complexity of the SVM algorithm either in the training or in the testing phase. Moreover, it is possible to implement a variant of the dot product that computes $\langle x,x\rangle$, $\langle x',x'\rangle$, $\langle x,x'\rangle$ with only one traversing of x and x' vectors, or precompute and store $\langle x,x\rangle$ for each sample in order to enhance the computational performances of the operators.

Intuitively all the kernels produces by $\mathcal{S}_\sigma$, $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_\sigma$ and $\mathcal{S}_{\sigma,\eta}$ (Eq. 5 6, 7, 8, 9) are quasi-local since they combine the original kernel with the locality information in its feature space. We will formalise this in subsection 3.6, while in the following subsection we will prove that the operators preserve the PD property of the input kernel.

3.3 The operators for quasi-local kernels preserve the PD property of the input kernels

We recall three well-known properties of PD kernels (for a comprehensive discussion of PD kernels refer to [3] or [37]):

Proposition 1 (Some properties of PD kernels)**.**

(i) *the class of PD kernels is a convex cone, i.e. if* $\alpha_1,\alpha_2 \geq 0$ *and* k_1, k_2 *are PD kernels then* $\alpha_1 k_1 + \alpha_2 k_2$ *is a PD kernel;*

(ii) *the class of PD kernels is closed under pointwise convergence, i.e. if* $k(x,x') := \lim_{n\to\infty} k_n(x,x')$ *exists for all* x, x', *then* k *is a PD kernel;*

(iii) *the class of PD kernels is closed under pointwise product, i.e. if* k_1, k_2 *are PD kernels, then* $(k_1 k_2)(x,x') := k_1(x,x')\cdot k_2(x,x')$ *is a PD kernel.*

The introduced operators preserve the PD property of the kernels on which they are applied, as stated in the following theorem.

Theorem 3. *If* k *is a PD kernel, then* $\mathcal{O}\,k$ *with* $\mathcal{O} \in \{\mathcal{E}_\sigma, \mathcal{P}_\sigma, \mathcal{S}_\sigma, \mathcal{S}_{\sigma,\eta}, \mathcal{PS}_\sigma, \mathcal{PS}_{\sigma,\eta}\}$ *is a PD kernel.*

Proof. It is straightforward to see that, for a PD kernel k, all the kernels resulting from the introduced operators can be obtained using properties (i) and (iii) of Proposition 1, provided that $\mathcal{E}_\sigma\,k$ is a PD kernel. So the only thing that remains to prove is that $\mathcal{E}_\sigma\,k$ is PD. Decomposing the definition of $(\mathcal{E}_\sigma\,k)(x,x')$ into three exponential functions we obtain:

$$(\mathcal{E}_\sigma\,k)(x,x') = \exp\left(\tfrac{2k(x,x')}{\sigma}\right)\exp\left(\tfrac{-k(x,x)}{\sigma}\right)\exp\left(\tfrac{-k(x',x')}{\sigma}\right)$$

that can be written as:

$$(\mathcal{E}_\sigma\,k)(x,x') = (\mathcal{O}^e\,2k/\sigma)(x,x')\cdot f(x)f(x')$$

where $\mathcal{O}^e\,2k/\sigma$ is the exponentiation of the $2k/\sigma$ kernel, and f is a real valued function such that $f(x) = \exp(-k(x,x)/\sigma)$. The first term is the exponentiation of a kernel multiplied by a non-negative constant and, since the kernel exponentiation can be seen as the limit of the series expansion of the exponential function which is the infinite sum of polynomial kernels, for property (ii) we conclude that $\mathcal{O}^e\,2k/\sigma$ is a PD kernel. Moreover, recalling from the definition of PD kernels, that the product $f(x)f(x')$ is a PD kernel for all the real-valued functions f defined in the input space [37] we conclude that $\mathcal{E}_\sigma\,k$ is a PD kernel. □

Obviously, if the input of $\mathcal{E}_\sigma$ is not a PD kernel, also the resulting function cannot be, in the general case, a PD kernel since the exponentiation operator maintains the PD property only for PD kernels. So, in the case of the sigmoidal kernel as input kernel, the resulting kernel is still not ensured to be PD.

3.4 Properties of the operators

In order to understand how the operators modify the original feature space of the input kernel we study the distances in the feature space of the quasi-local kernels. The new feature space introduced by kernels produced by the operators is denoted with $\mathcal{F}_\mathcal{O}$, the corresponding mapping function with $\Phi_\mathcal{O}$ and the distance between two input points mapped in $\mathcal{F}_\mathcal{O}$ with $dist_{\mathcal{F}_\mathcal{O}}(x,x') = m(\Phi_\mathcal{O}(x), \Phi_\mathcal{O}(x'))$ where m is a metric in $\mathcal{F}_\mathcal{O}$. Applying the kernel trick for distances, we can express the squared distances in $\mathcal{F}_\mathcal{O}$ as:

$$dist^2_{\mathcal{F}_\mathcal{O}}(x,x') = \|\Phi_\mathcal{O}(x) - \Phi_\mathcal{O}(x')\|^2 = (\mathcal{O}\,k)(x,x) + (\mathcal{O}\,k)(x',x') - 2(\mathcal{O}\,k)(x,x'). \tag{10}$$

For $\mathcal{O} = \mathcal{E}_\sigma$, since it is clear that $dist_\mathcal{F}(x,x) = 0$ for every x, we can derive $dist_{\mathcal{F}_{\mathcal{E}_\sigma}}$ as follows:

$$\begin{aligned} dist^2_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x') &= \exp\left(-\tfrac{dist^2_\mathcal{F}(x,x)}{\sigma}\right) + \exp\left(-\tfrac{dist^2_\mathcal{F}(x',x')}{\sigma}\right) - 2\exp\left(-\tfrac{dist^2_\mathcal{F}(x,x')}{\sigma}\right) = \\ &= 2\left[1 - exp\left(-\tfrac{dist^2_\mathcal{F}(x,x')}{\sigma}\right)\right]. \end{aligned} \tag{11}$$

Note that $dist^2_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x') \leq 2$ for every pair of samples, and so the distances in $\mathcal{F}_{\mathcal{E}_\sigma}$ are bounded even if they are not bounded in $\mathcal{F}$.

Substituting $\mathcal{P}_\sigma$, $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ in Eq. 10, an taking into account Eq. 11, the distances in $\mathcal{F}_\mathcal{O}$ for the quasi-local kernels are:

$$\begin{aligned} dist^2_{\mathcal{F}_{\mathcal{P}_\sigma}}(x,x') &= dist^2_\mathcal{F}(x,x') + k(x,x')\, dist^2_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x'); \\ dist^2_{\mathcal{F}_{\mathcal{S}_{\sigma,\eta}}}(x,x') &= dist^2_\mathcal{F}(x,x') + \eta \cdot dist^2_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x'); \\ dist^2_{\mathcal{F}_{\mathcal{PS}_{\sigma,\eta}}}(x,x') &= (1+\eta)\, dist^2_\mathcal{F}(x,x') + \eta \cdot k(x,x')\, dist^2_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x') = \\ &= dist^2_\mathcal{F}(x,x') + \eta \cdot dist^2_{\mathcal{F}_{\mathcal{P}_\sigma}}(x,x'). \end{aligned} \tag{12}$$

We can notice that the distances in $\mathcal{F}_{\mathcal{E}_\sigma}$ and in $\mathcal{F}_{\mathcal{S}_{\sigma,\eta}}$ do not contain explicitly the kernel function but they are based only on the distances in $\mathcal{F}$. So we can further analyse the behaviour of the distances in $\mathcal{F}_{\mathcal{E}_\sigma}$ and $\mathcal{F}_{\mathcal{S}_{\sigma,\eta}}$ with the following proposition.

Proposition 2. *The operators $\mathcal{E}_\sigma$ and $\mathcal{S}_{\sigma,\eta}$ preserve the ordering on distances in $\mathcal{F}$. Formally*

$$dist_\mathcal{F}(x,x') < dist_\mathcal{F}(x,x'') \Rightarrow dist_{\mathcal{F}_\mathcal{O}}(x,x') < dist_{\mathcal{F}_\mathcal{O}}(x,x'')$$

for $\mathcal{O} \in \{\mathcal{E}_\sigma, \mathcal{S}_{\sigma,\eta}\}$ and for every sample x, x', x''.

Proof. It follows directly from the observations that $dist_{\mathcal{F}_{\mathcal{E}_\sigma}}(x,x')$ and $dist_{\mathcal{F}_{\mathcal{S}_{\sigma,\eta}}}(x,x')$ are defined with strictly increasing monotonic functions, Eq. 11 and the second equation in Eq. 12 respectively, and that $dist_\mathcal{F}$ is always non-negative. □

This means that $\mathcal{E}_\sigma\, k$ kernel determines the same neighborhoods as k and that the $\mathcal{E}_\sigma\, k$ exploits the locality information weighting the influence of the neighbors of a point in the feature space of k maintaining the property that points at distance d in the feature space of k influence the $\mathcal{E}_\sigma\, k$ kernel score more than any other more distant points. In other words $\mathcal{E}_\sigma\, k$ modifies the influence of the points using the features space distances but the ordering on the weights is the same of the ordering on distances in the input space.

The $\mathcal{E}_\sigma\, k$ kernel has also an interesting property regarding the class of universal kernels. Roughly speaking, universal kernels, introduced in [43] and further discussed in [44–46], are kernels that permits to optimally approximate the Bayes decision rule or, equivalently, to learn an arbitrary continuous function uniformly on any compact subset of the input space. Applying Proposition 8 and Corollary 10 in [43], it turns out that $\mathcal{E}_\sigma\, k$ is universal in the feature space of k. Intuitively this happens because $\mathcal{E}_\sigma\, k$ builds an k^{rbf} kernel, which is universal, in the feature space of k. This means that, regardless of the universality of the input kernel, the $\mathcal{E}_\sigma$ always finds a space on which the resulting kernel is universal.

3.5 Connections between $\mathcal{E}_\sigma\ k^{rbf}$ and k^{rbf} with variable kernel width

Since k^{rbf} is a local kernel, a question that naturally arises concerns the behaviour of $\mathcal{E}_\sigma\ k^{rbf}$, i.e. the quasi-local transformation of a local kernel. In particular the point is to understand if k^{rbf} and $\mathcal{E}_\sigma\ k^{rbf}$ exploit the same notion of locality. If it is the case, this would mean that $\mathcal{E}_\sigma\ k^{rbf}$ and k^{rbf} are basically equivalent and identify the same features space, possibly under certain parameter settings. This question is addressed by the following Proposition.

Proposition 3. *There not exist two constant $\sigma, \gamma^{rbf} \in \mathbb{R}$ with $\sigma > 0$ and $\gamma \geq 0$, such that, for every $x, x' \in X$ with X with at least 3 distinct points, the following holds:*

$$k^{rbf}(x, x') = (\mathcal{E}_\sigma\ k^{rbf})(x, x') \tag{13}$$

Proof. Suppose, by contradiction, that there exist $\sigma, \gamma^{rbf} \in \mathbb{R}$ such that, for every $x, x' \in X$, Eq. 13 holds. It can be rewritten as:

$$\exp(-\gamma^{rbf} \cdot \|x - x'\|^2) =$$
$$= \exp\left(-\frac{\exp(-\gamma^{rbf} \cdot \|x - x\|^2) + \exp(-\gamma^{rbf} \cdot \|x' - x'\|^2) - 2 \cdot \exp(-\gamma^{rbf} \cdot \|x - x'\|^2)}{\sigma}\right)$$

Since $\exp(-\gamma^{rbf} \cdot \|x - x\|^2) = 1$, we can obtain:

$$-\gamma^{rbf} \cdot \|x - x'\|^2 = \frac{-2 + 2 \cdot \exp(-\gamma^{rbf} \cdot \|x - x'\|^2)}{\sigma},$$

from which we have

$$\exp(-\gamma^{rbf} \cdot \|x - x'\|^2) = 1 - \frac{\gamma^{rbf}\sigma}{2} \cdot \|x - x'\|^2$$

that can be written as:

$$k^{rbf}(x, x') = 1 - \frac{\gamma^{rbf}\sigma}{2} \cdot \|x - x'\|^2.$$

Since, with respect to the square of the Euclidean distance $\|x - x'\|^2$, $k^{rbf}(x, x')$ is a negative exponential function, whereas $1 - \|x - x'\|^2 \cdot \frac{\gamma^{rbf}\sigma}{2}$ is a non-increasing linear function, the two function can have no more than 2 points in common. Because σ and γ^{rbf} are constant, while $\|x - x'\|^2$ is not constant, it is straightforward to conclude that $k^{rbf}(x, x') \neq 1 - \|x - x'\|^2 \cdot \frac{\gamma^{rbf}\sigma}{2}$ at least for some $x, x' \in X$. In this way we get a contradiction thus proving the proposition. □

From this proposition we can conclude that $\mathcal{E}_\sigma\ k^{rbf}$ cannot be emulated by k^{rbf} and thus it introduces an higher degree of locality. Intuitively an increased level of locality can be introduced locally adjusting the local parameters. In the specific case of k^{rbf} this intuition can be applied permitting to the width parameter $(1/\gamma^{rbf})$ to be locally adaptive, as proposed for example in [47]. The following proposition demonstrate that $\mathcal{E}_\sigma\ k^{rbf}$ is equivalent to k^{rbf} with variable kernel width.

Proposition 4. *There exists a real-valued function $f(\sigma, \gamma^{rbf}, \|x - x'\|)$ such that the following holds for each $x, x' \in X$:*

$$\exp\left(-\frac{\|x - x'\|^2}{f(\sigma, \gamma^{rbf}, \|x - x'\|)}\right) = (\mathcal{E}_\sigma\ k^{rbf})(x, x') \tag{14}$$

Proof. We can easily find such function f isolating it from Eq. 14:

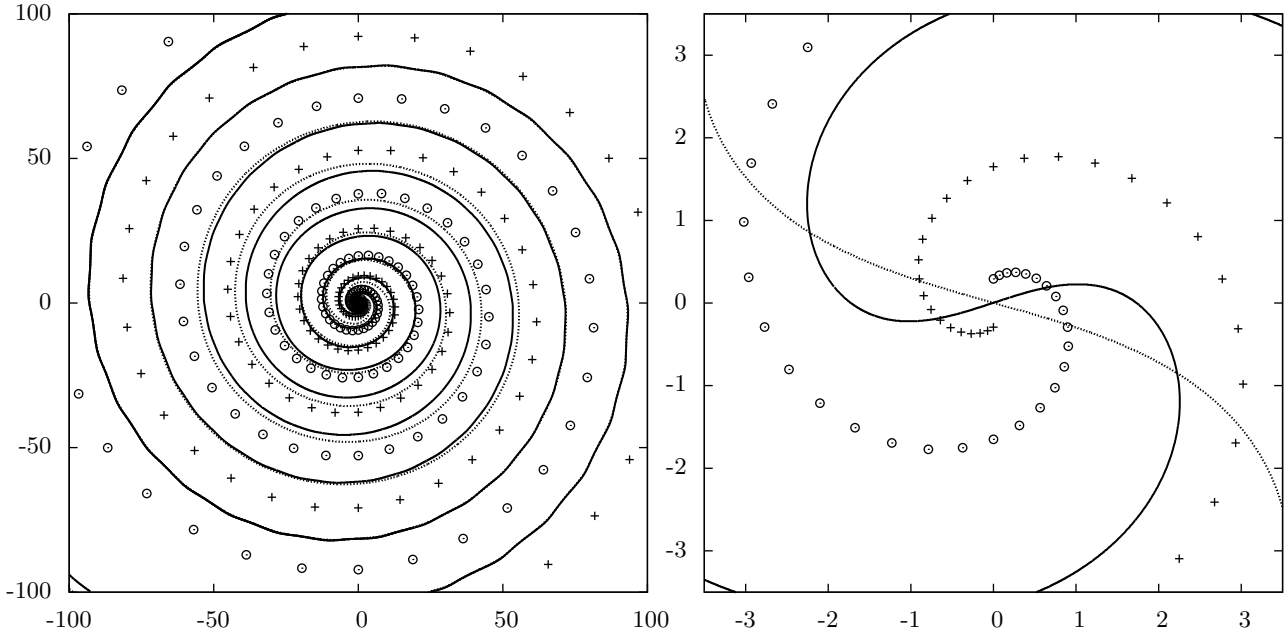
$$\exp\left(-\frac{\|x - x'\|^2}{f(\sigma, \gamma^{rbf}, \|x - x'\|)}\right) = \exp\left(\frac{-2 + 2 \cdot \exp(-\gamma^{rbf} \cdot \|x - x'\|^2)}{\sigma}\right)$$

obtaining:

$$f(\sigma, \gamma^{rbf}, \|x - x'\|) = \frac{\sigma}{2} \cdot \frac{\|x - x'\|^2}{1 - \exp(-\gamma^{rbf} \cdot \|x - x'\|^2)}. \tag{15}$$

□

We thus found the function regulating the variable k^{rbf} width. It can be shown that Eq. 15 has always positive derivative, meaning that it always grows as the distance between samples grows. This causes the kernel width to be lower for close points and higher for distant points, thus permitting to alleviate the tradeoff between over- and under-fitting on which a uniform kernel width is based. The variable kernel width is particularly crucial in presence of data with uneven densities.



(a) The whole two spiral dataset

(b) The two spiral dataset zoomed.

Figure 1: The behaviour of k^{rbf} (the dotted line) and $\mathcal{E}_\sigma\ k^{rbf}$ (the filled line) on the two spiral problem (the samples of the two classes are denoted by + and ⊙ symbols). The model parameters are obtained with 20-fold CV. The best training set accuracy is 0.823 for k^{rbf} and 0.907 for $\mathcal{E}_\sigma\ k^{rbf}$.

We illustrate these considerations with the application of k^{rbf} and $\mathcal{E}_\sigma\ k^{rbf}$ on the two spirals artificial dataset shown in Figure 1. Both k^{rbf} and $\mathcal{E}_\sigma\ k^{rbf}$ are applied with the best parameters obtained with a grid search 20-fold CV on $C, \gamma^{rbf}, \sigma \in \{2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\}$. The

best training accuracy of k^{rbf} is 0.823 whereas $\mathcal{E}_\sigma\ k^{rbf}$ reaches 0.907, meaning that the quasi-local kernel approach is able to find a better decision function. This is evident also graphically, in fact, while in the peripheral regions of the datasets (see Figure 1(a)) both classifiers find a good decision function, whereas in the central region (see Figure 1(b)) k^{rbf} starts to clearly underfit the data.

3.6 Quasi-local kernels

In this section, we formally introduce the notion of quasi-local kernels, and we show that kernels produced by the $\mathcal{S}_\sigma$, $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_\sigma$ and $\mathcal{S}_{\sigma,\eta}$ are quasi-local kernels. Firstly, we introduce the concept of locality with respect to a function:

Definition 3. *Given a PD kernel k with implicit mapping function $\Phi : \mathbb{R}^p \mapsto \mathcal{F}$ (namely $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$), and a function $\Psi : \mathbb{R}^p \mapsto \mathcal{F}_\Psi$, k* is local with respect to Ψ *if there exists a function $\Omega : \mathcal{F}_\Psi \mapsto \mathcal{F}$ such that the following holds:*

1. $\langle \Phi(x), \Phi(x_i) \rangle = \langle \Omega(\Psi(x)), \Omega(\Psi(x_i)) \rangle$ *for all $x, x_i \in \mathbb{R}^p$*
2. $\lim\limits_{\|u - v_i\|_{\mathcal{F}_\Psi} \to \infty} \langle \Omega(u), \Omega(v_i) \rangle = c_i$ *with $u = \Psi(x)$, $v_i = \Psi(x_i)$ for some $x, x_i \in \mathbb{R}^p$ and c_i constant and not depending on u.*

In other terms, the notion of locality referred to samples in input space (Definition 1), is modified here in order to consider the locality in any space accessible from the input space through a corresponding mapping function. Notice that, as particular cases, we have that every local kernel is local with respect to the identity function and with respect to its own implicit mapping function.

With the next theorem we see that the $\mathcal{E}_\sigma$ formally respect the idea of producing kernels that are local with respect to the feature space of the input kernel.

Theorem 4. *If k is a PD kernel with the implicit mapping function $\Phi : \mathbb{R}^p \mapsto \mathcal{F}$, then $\mathcal{E}_\sigma\, k$ is local with respect to Φ.*

Proof. We have already shown that $\mathcal{E}_\sigma\, k$ is a PD kernel given that k is a PD kernel (see Theorem 3). It remains to show that $\mathcal{E}_\sigma\, k$ is local with respect to Φ.

First we need to show that (Definition 3 point 1), denoted with $\Phi' : \mathbb{R}^p \mapsto \mathcal{F}'$ the implicit mapping function of $\mathcal{E}_\sigma\, k$, there exists a function $\Omega : \mathcal{F} \mapsto \mathcal{F}'$ such that $\Phi'(x) = \Omega(\Phi(x))$. Taking as $\Omega : \mathcal{F} \mapsto \mathcal{F}'$ the implicit mapping of the kernel $\exp\left(-\frac{\|u - v_i\|}{\sigma}\right)$ with $u = \Phi(x)$, $v_i = \Phi(x_i)$ with $x, x_i \in \mathbb{R}^p$ we have

$$\langle \Omega(u), \Omega(v_i) \rangle = \exp\left(-\frac{\|u - v_i\|}{\sigma}\right). \tag{16}$$

Using the hypothesis on u and v_i it becomes:

$$\exp\left(-\frac{\|\Phi(x) - \Phi(x_i)\|}{\sigma}\right) = \langle \Omega(\Phi(x)), \Omega(\Phi(x_i)) \rangle. \tag{17}$$

The implicit mapping function of $\mathcal{E}_\sigma\, k$ is Φ' and so

$$\langle \Phi'(x), \Phi'(x_i) \rangle = (\mathcal{E}_\sigma\, k)(x, x_i) \tag{18}$$

Moreover since $(\mathcal{E}_\sigma\, k)(x, x_i) = \exp\left(-\frac{\|\Phi(x) - \Phi(x_i)\|}{\sigma}\right)$ for definition of $\mathcal{E}_\sigma$ (see Eq. 4), substituting Eq. 17 into Eq. 18 we conclude that

$$\langle \Phi'(x), \Phi'(x_i) \rangle = \langle \Omega(\Phi(x)), \Omega(\Phi(x_i)) \rangle.$$

Second, we need to show that (Definition 3 point 2) $\langle \Omega(u), \Omega(v_i) \rangle \to c_i$ with c_i constant for $\|\Omega(u) - \Omega(v_i)\| \to \infty$. From the Eq. 16, it is clear that, as the distance between $u = \Phi(x)$ and $v_i = \Phi(x_i)$ tend to infinity, the kernel value is equal to the constant 0 regardless of x. □

Now we can define the quasi-locality property of a kernel.

Definition 4 (Quasi-local kernel)**.** *A PD kernel k is a quasi-local kernel if $k = f(k^{inp}, k^{loc})$ where k^{inp} is a PD kernel with implicit mapping function $\Phi : \mathbb{R}^p \mapsto \mathcal{F}$, k^{loc} is a PD kernel which is local with respect to Φ and f is a function involving legal and non trivial operations on PD kernels.*

For legal operations on kernels we mean operations preserving the PD property. For non trivial operations we intend operations that always maintain the influence of all the input kernels in the output kernel; more precisely a function $f(k_1, k_2)$ does not introduce trivial operations if there exists two kernels k' and k'' such that $f(k', k_2) \neq f(k_1, k_2)$ and $f(k_1, k'') \neq f(k_1, k_2)$. Notice that the k^{inp} kernel of the definition corresponds to the input kernel of the operator that produces the quasi-local kernel k.

Theorem 5. *If k is a PD kernel, then $\mathcal{S}_\sigma\, k$, $\mathcal{S}_{\sigma,\eta}\, k$, $\mathcal{PS}_\sigma\, k$ and $\mathcal{S}_{\sigma,\eta}\, k$ are quasi-local kernels.*

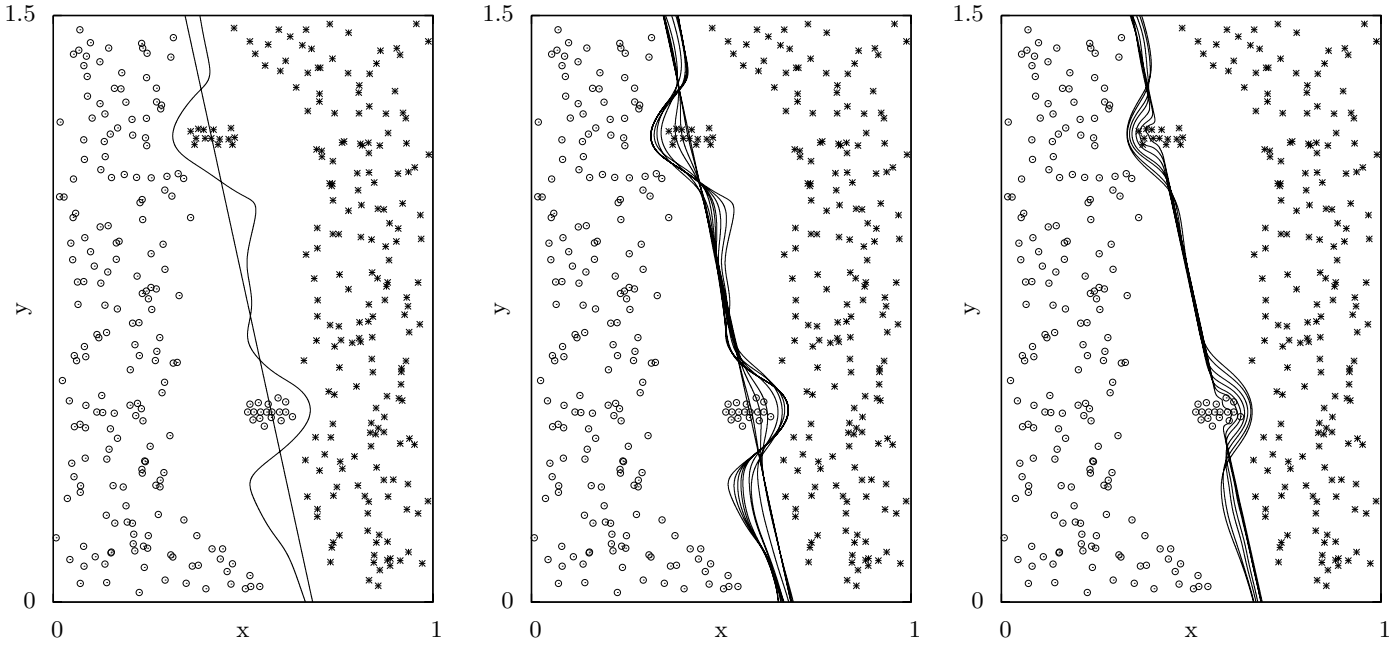
Proof. Theorem 4 already states that $\mathcal{E}_\sigma\, k$ is a PD kernel which is local with respect to the implicit mapping function Φ of the kernel k which is PD for hypothesis. It is easy to see that all the kernels resulting from the introduced operators can be obtained using properties (i) and (iii) of Proposition 1 starting from the two PD kernels k and $\mathcal{E}_\sigma\, k$, and thus $\mathcal{S}_\sigma\, k$, $\mathcal{S}_{\sigma,\eta}\, k$, $\mathcal{PS}_\sigma\, k$ and $\mathcal{S}_{\sigma,\eta}\, k$ are PD kernels obtained with legal operations. Moreover, the properties (i) and (iii) of Proposition 1 introduce multiplications and sums between kernels and between kernels and constant. The sums introduced by the operators are always non trivial because they always consider positive addends, and so it is for the multiplications because they never consider null factors (the introduced constants are non null for definition). □

Both quasi-local kernels and KNNSVM classifiers are based on the notion of locality in the feature space. However, two main theoretical differences can be found between them. The first is that in KNNSVM locality is included directly, considering only the points that are close to the testing point, while for the quasi-local kernels the information of the input kernel is balanced with the local information. The second consideration concerns the fact that KNNSVM has a variable but hard boundary between the local and non local points, while $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ produce kernels whose locality decreases exponentially but in a continuous way.

3.7 Parameter choice and empirical risk minimization for quasi-local kernels

There are two parameters for the operators on kernels through which we obtain the quasi-local kernels: σ, which is present in $\mathcal{E}_\sigma$ and consequently in all the operators, and η, which is present in $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ ($\mathcal{S}_\sigma$ and $\mathcal{PS}_\sigma$ can be seen as special cases of $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ with $\eta = 1$).

The role of these two parameters will be better illustrated in the next section. Here we propose a strategy for choosing their values. The idea is that a quasi-local operator is applied on an already optimized kernel in order to further enhance the classification capability introducing locality. Notice that, in general, it would be possible to estimate the input kernel parameters, the SVM penalty parameter C and the operator parameters at the same time, but this is in contrast with the above idea. Ideally the operators can accept a kernel matrix without knowledge about the kernel function from which it is generated. So the approach we adopt here is to apply the operators on a kernel for which the parameters are already set, thus requiring only one parameter optimization (for $\mathcal{E}_\sigma$, $\mathcal{P}_\sigma$ and $\mathcal{PS}_\sigma$) or two (for $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$).



(a) k^{lin} and k^{rbf} (b) $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ varying η (c) $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ varying σ

Figure 2: The separating hyperplanes for a two-feature hand-built artificial datasets defined by the application of the SVM (all with $C = 3$) with (a) linear kernel k^{lin} and RBF kernel k^{rbf} (with $\gamma^{rbf} = 150$), (b) the $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ quasi-local kernel with fixed σ ($\sigma = 1/150 = 1/\gamma^{rbf}$) and variable η ($\eta = 10^6, 50, 10, 1, 0.5, 0.1, 0.05, 0.03, 0.01, 0.005, 0.001, 0.000001$), and (c) the $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ quasi-local kernel with fixed η ($\eta = 0.05$) and variable σ ($\sigma = 1/5000, 1/2000, 1/1000, 1/500, 1/300, 1/150, 1/100$).

Moreover, we provide some data-dependent estimations of σ η permitting the reduction of the number of parameters values that need to be optimized (3 for η and 4 for σ).

The dataset-dependent estimation of σ take inspiration from the γ^{rbf} estimation, since σ and γ^{rbf} play a similar role of controlling the width of the kernel. However, differently from the k^{rbf} kernel, the $\mathcal{E}_\sigma$ operator uses distances in the feature space $\mathcal{F}$ (except for the special case $k = k^{lin}$). More precisely, remembering that the data-dependent values of γ^{rbf} are obtained with $\gamma^{rbf}_h = 1/(2 \cdot q^2_h[\|x - x'\|^{\mathbb{R}^p}])$ where $q_h[\|x - x'\|^{\mathcal{Z}}]$ denotes the h percentile of the distribution of the distance in the $\mathcal{Z}$ space between every pair of points x, x', the σ parameter can be estimated using $\sigma_h = 2 \cdot q^2_h[\|x - x'\|^{\mathcal{F}}]$ with $h \in \{1, 10, 50, 90\}$ as for the γ^{rbf} case. For η we adopt $\eta_h = q_h[\|x - x'\|^{\mathcal{F}}]$ with $h \in \{10, 50, 90\}$.

We thus have a total of 12 quasi-local parameter configurations, meaning that the model selection for quasi-local kernels in this scenario requires no more than 12 cross-validation runs to choose the best parameters. Notice that, comparing the cross-validation best values of the input kernel and quasi-local kernels, we implicitly test also the $\eta = 0$ case. Since $\mathcal{S}_{\sigma,\eta}\, k$ and $\mathcal{PS}_{\sigma,\eta}\, k$ with $\eta = 0$ are equivalent to k, $\mathcal{S}_{\sigma,\eta}\, k$ and $\mathcal{PS}_{\sigma,\eta}\, k$ have the possibility to reduce to k as a special case. In our empirical evaluation we will highlight the cases in which $\eta = 0$ is selected.

4 Intuitive behaviour of quasi-local kernels

The operators on kernels defined in the previous section aim to modify the behaviour of an input kernel k in order to produce a kernel more sensitive to local information in the feature space, maintaining however the original behaviour for regions in which the locality is not important. In addition the η and σ parameters control the balance between the input kernel k and its local reformulation $\mathcal{E}_\sigma\, k$, in other words the effects of the local information.

These intuitions are highlighted in Figure 2 with an example that illustrates the effects of the $\mathcal{S}_{\sigma,\eta}$ operator on the linear kernel k^{lin} using a two-feature hand-built artificial dataset. Notice that this example is not limited to the combination of k^{lin} and k^{rbf}, because it represents the intuition of what happens in the feature space of the original kernel applying the $\mathcal{S}_{\sigma,\eta}$ operator. The transformed kernel is:

$$(\mathcal{S}_{\sigma,\eta}\, k^{lin})(x, x') = k^{lin}(x, x') + \eta \cdot (\mathcal{E}_\sigma\, k^{lin})(x, x') = k^{lin}(x, x') + \eta \cdot k^{rbf}(x, x') \qquad (19)$$

with $\gamma^{rbf} = 1/\sigma$. So the $\mathcal{S}_{\sigma,\eta}$ operator on the k^{lin} kernel gives a linear combination of k^{lin} and k^{rbf}. Figure 2(a) show the separate behaviours of the global term k^{lin} alone and of the local term $\mathcal{E}_\sigma\, k^{lin} = k^{rbf}$ alone. Figure 2(b) illustrates what happens when the local and the global terms are combined with different values of η and a fixed σ. Figure 2(c) shows the behaviour of the separating hyperplane with a fixed balancing factor η but varying the σ parameter.

The η parameter regulates the influence on the separating hyperplane of the local term of the quasi-local kernel. In fact, in Figure 2(b), we see that all the planes lie between the input kernel (k^{lin}, obtained with $\eta \to 0$ from $\mathcal{S}_{\sigma,\eta}\, k^{lin}$) and the local reformulation of the same kernel (obtained with $\eta = 10^6$ from $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ which behaves as k^{rbf} since the high value of η partially hides the effect of the global term). Moreover, since σ is low, the modifications induced by different values of η are global, influencing all the regions of the separating hyperplane.

We can observe in Figure 2(c), on the other hand, that σ regulates the magnitude of the distortion from the linear hyperplane for the region containing points close to the plane itself. The σ parameter in the $\mathcal{E}_\sigma\, k^{lin}$ term of $\mathcal{S}_{\sigma,\eta}\, k^{lin}$ has a similar role to the K parameter in the local SVM approach (i.e. it regulates the range of the locality), even though K defines an hard boundary between local and non local points instead of a negative exponential one. It is important to emphasize that in the central region of the dataset the separating hyperplane remains linear, highlighting that the kernel resulting from the $\mathcal{S}_{\sigma,\eta}$ operator differs from the input kernel only where the information is local.

The example simply illustrates the intuition behind the proposed family of quasi-local kernels, and in particular how the input kernel behaviour in the feature space is maintained for the regions in which the information is not local, so it is not important here to analyse the classification accuracy. However, kernels that are sensitive to important local information but retain properties of global input kernels, can also be obtained from very elaborated and well tuned kernels defined on high-dimensionality input spaces. In the following two sections we investigate the accuracy performances of the quasi-local kernels in a number of real datasets using a data-dependent method of choosing η and σ parameters.

5 Experiment 1

The goal of the first experiment is to compare the accuracy of SVM with quasi-local kernels against SVM with traditional kernels and kNNSVM. The evaluation is carried out on 23 non-large datasets.

Table 1: The 23 datasets for Experiment 1 ordered by training set size.

Name	brief description	# classes	train. size	# features
leukemia	Cancer classification, originally from [49]	2	38	7129
iris	A well known pattern recognition dataset	3	150	4
wine	wine recognition from chemical data, preproc. as [50]	3	178	13
sonar	discrimination between different sonar signals	2	208	60
glass	types of glass classification	6	214	9
heart	heart disease prediction, originally from Statlog [51]	2	270	13
liver	liver disorders prediction from alcohol consumption data	2	345	6
ionosphere	classification of radar signals from the ionosphere	2	351	34
bioinf	(or *svmguide2*) bioinformatics data originally from [52]	3	391	20
vowel	automatic recognition of British English vowels	11	528	10
breast	Wisconsin breast cancer data	2	683	10
australian	australian credit approval, originally from Statlog [51]	2	690	14
diabetes	Pima indians diabetes data	2	768	8
vehicle	vehicle recognition [50], originally from Statlog [51]	4	846	18
fourclass	a 4 class problem [53] transformed to a 2 class problem	2	862	2
splice	primate splice-junction gene sequences data	2	1000	60
numer	German credit risk data, originally from Statlog [51]	2	1000	24
vehicle2	(or *svmguide3*) vehicle data originally from [52]	2	1243	21
a1a	Adult dataset preprocessed as done by [54]	2	1605	123
dna	DNA problem preprocessed as done in [55]	3	2000	180
segment	image segmentation data originally from Statlog [51]	7	2310	19
w1a	web page classification, originally from [54]	2	2477	300
astro	(or *svmguide1*) astroparticle application from [52]	2	3089	4

5.1 Experimental procedure

Table 1 lists the 23 datasets from different sources and scientific fields used in this experiment; we took all the freely available datasets from the LibSVM repository [48] with training set with no more than 3500 samples. Some datasets are multiclass and the number of features ranges from 2 to 7129.

The reference input kernels for the quasi-local operators considered are the linear kernel k^{lin}, the polynomial kernel k^{pol}, the radial basis function kernel k^{rbf} and the sigmoidal kernel k^{sig}. The quasi-local kernels we tested are those resulting from the application of the $\mathcal{E}_\sigma$, $\mathcal{P}_\sigma$, $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_{\sigma,\eta}$ operators on the reference input kernels. We also evaluated the accuracy of the KNNSVM classifier with the same reference input kernels.

The methods are evaluated using 10-fold cross validation. The assessment of statistical significant difference between two methods on the same dataset is performed with the two-tailed paired t-test ($\alpha = 0.05$) on the two sets of fold accuracies. Although the count of positive and negative significative difference can be used to establish if a method performs better than another on multiple datasets, it has been shown [56] that the Wilcoxon signed-ranks test [57] is a theoretically safer (with respect to parametric tests it does not assume "normal distributions or homogeneity of variance") and empirically stronger test. For this reason the final assessment of statistical significance difference on the 23 datasets is performed with the Wilcoxon signed-ranks test (in case of ties, the rank is calculated as the average rank between them).

The model selection is performed on each fold with a inner 5-fold cross validation as follows. For all the methods tested, the regularization parameter C is chosen in $\{2^{-5}, 2^{-4}, \ldots, 2^{9}, 2^{10}\}$. For the polynomial kernel we adopt the widely used homogeneous polynomial kernel ($\gamma^{pol} = 1$, $r^{pol} = 0$), selecting a degree non higher than 5. The choice of γ^{rbf} for the RBF kernel is done adopting γ_h^{rbf} where h is chosen among $\{1, 10, 50, 90ua\}$ as described in 2.1. For the sigmoidal kernel, r^{sig} is set to 0, whereas γ^{sig} is chosen among $\{2^{-7}, 2^{-6}, \ldots, 2^{-1}, 2^{0}\}$. For the quasi-local kernels we use the C and kernel parameters found by the model selection described

above for each input kernel, whereas σ is chosen using $\sigma_h^{\mathcal{F}}$ with $h \in \{1, 10, 50, 90\}$ and η using $\eta_{,h}^{\mathcal{F}}$ with $h \in \{10, 50, 90\}$ and (implicitly) $\eta = 0$ as described in Section 3.7 through a 5-fold cross validation on the same folds used for model selection on the input kernels. Finally, the value of K for KNNSVM is automatically chosen on the training set between $\mathcal{K} = \{1, 3, 5, 7, 9, 11, 15, 23, 39, 71, 135, 263, 519, 1031\}$ (the first 5 odd natural numbers followed by the ones obtained with a base-2 exponential increment from 9) as described in section 2.2.

We used LibSVM library [48] version 2.84 for SVM training and testing, extending it with a object-oriented architecture for kernel calculation and specification. For the quasi-local kernels we store the values of $\langle x, x \rangle$ for each sample in order to obtain the quasi-local kernel value computing only one scalar product, i.e. $\langle x, x' \rangle$, instead of three. The KNNSVM implementation is based on the same version of LibSVM. The experiments are carried out on multiple Intel® Xeon™ CPU 3.20GHz systems, setting the kernel cache dimension to 1024Mb and interrupting the experiments that are not terminated after 72 hours.

5.2 Results

Table 2 reports the 10-fold cross validation accuracy of SVM with the four considered input kernels, SVM with the quasi-local kernels obtained applying the $\mathcal{E}_\sigma$, $\mathcal{P}_\sigma$, $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_{\sigma,\eta}$ operators and KNNSVM. Some KNNSVM results are missing due to the computational effort of the method, corresponding to the cases in which KNNSVM does not terminates within 72 hours. The $^+$ and $^-$ denotes quasi-local kernel and KNNSVM results that are significatively better (and worse) than the corresponding input kernels according to the two-tailed paired t-test ($\alpha = 0.05$). The total number of datasets in which quasi-local kernels and KNNSVM perform better (and worse) than the corresponding input kernels are reported, with the number of significative differences in parenthesis. The last row reports the Wilcoxon signed-ranks tests to assess the significant improvements of quasi-local kernels over corresponding input kernels on all the datasets (for KNNSVM only on the datasets for which the results are present). The cases in which, for $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_{\sigma,\eta}$, the model selection chose $\eta = 0$ for all the 10 folds thus giving the same results of the input kernels, are underlined. In bold, are highlighted the best 10-fold cross validation accuracy achieved for a specific dataset among all the methods and kernels.

5.3 Discussion

The KNNSVM results basically confirm the earlier assessment [16], although the model selection is performed here differently; KNNSVM is able to improve, according to the Wilcoxon signed-rank test, the classification generalization accuracy of SVM with the k^{lin} kernel (10 two-tailed paired t-test significant improvements, 1 deteriorations) and k^{sig} kernel (8 two-tailed paired t-test significant improvements, 1 deteriorations). Instead we do not have evidence of improved generalization accuracy on the benchmark datasets for the k^{pol} kernel and k^{rbf} kernel, although we showed in [16] that, for k^{rbf}, there are scenarios in which KNNSVM can be particularly indicated.

$\mathcal{E}_\sigma\, k$ seems to perform significantly better than k for k^{sig} and for k^{lin} (although without statistical evidence), whereas there are no overall improvement for k^{rbf}, and for k^{pol} the accuracies are deteriorated. These results are probably due to the choice of not re-performing model selection for $\mathcal{E}_\sigma\, k$ in particular for the C parameter. In fact $\mathcal{E}_\sigma$ is the only operator that does not contain the input kernel explicitly in the resulting one, and thus the optimal parameters can be very different. This is confirmed by the fact that $\mathcal{E}_\sigma\, k^{lin}$ is equivalent to k^{rbf} but their results, as reported in Table 2, appears to be are very different.

The results of $\mathcal{P}_\sigma\, k$ are slightly better than $\mathcal{E}_\sigma\, k$. According to the Wilcoxon signed-rank test, it is better than k for $k = k^{sig}$ and $k = k^{rbf}$, but not for k^{rbf} and k^{pol}. In total, the

Table 2: Experiment 1. 10-fold CV accuracy of SVM with the input and quasi-local kernels and of KNNVM.

dataset	$k = k^{lin}$						$k = k^{rbf}$					
	k	$\mathcal{E}_\sigma k$	$\mathcal{P}_\sigma k$	$\mathcal{S}_{\sigma,\eta} k$	$\mathcal{PS}_{\sigma,\eta} k$	KNNSVM	k	$\mathcal{E}_\sigma k$	$\mathcal{P}_\sigma k$	$\mathcal{S}_{\sigma,\eta} k$	$\mathcal{PS}_{\sigma,\eta} k$	KNNSVM
leukemia	**.947**	.763$^-$	**.947**	**.947**	**.947**	.921	**.947**	.895	.921	**.947**	**.947**	.895
iris	.967	.960	.960	**.973**	**.973**	.960	.960	.953	.960	**.973**	.967	**.973**
wine	.972	.972	.978	.978	.978	.966	.972	.972	.978	.978	.972	.966
sonar	.745	.755	.880$^+$	.870$^+$	.894$^+$	.899$^+$	.894	.899	**.904**	.894	.894	.865$^-$
glass	.640	.692	.710$^+$	.687	.701$^+$	.710$^+$	.682	.668	.678	.715$^+$	.706$^+$	.734$^+$
heart	**.833**	.822	.811	.826	.826	.811	.819	.822	.807	.822	.819	.815
liver	.687	.722	.713	.733$^+$	.722	.733$^+$	.719	.733	.725	.725	.728	.725
ionosphere	.883	.943$^+$	.943$^+$	.929	.949$^+$	.940$^+$	.940	**.954**	**.954**	.946	.952	.943
bioinf	.818	.841	.821	.834	.826	.841	.831	.836	.841	.831	.839	.854
vowel	.848	.989$^+$	.989$^+$	.992$^+$	.991$^+$	**.996**$^+$	.992	.994	.994	**.996**	**.996**	**.996**
breast	.958	.966	.965	.966	.962	.971	.969	.969	.968	.971	.972	.971
australian	.848	.848	.839	.846	.848	.864	.843	.851	.843	.852	.848	.851
diabetes	.766	.766	.754	.772	.775	**.779**	.772	.763	.770	.766	.762	.768
vehicle	.800	.849$^+$	.853$^+$	.855$^+$	.859$^+$	**.866**$^+$	.857	.856	.849	.849	.857	.853
fourclass	.774	.987$^+$	.922$^+$	.988$^+$	.950$^+$	**1.00**$^+$	**1.00**	.998	.999	**1.00**	**1.00**	.999
splice	.800	.774	.872$^+$	.848$^+$	.884$^+$	-	**.885**	.882	.882	.881	.880	-
numer	.769	.747	.698$^-$	.765	.770	-	.760	.757	.750$^-$	.761	.765	.757
vehicle2	.829	.846	.841	.847$^+$	.848	.828	.843	.849	.838	.845	.846	.840
a1a	**.833**	.800$^-$	.802$^-$	.831	.832	-	.828	.831	.827	.831	.827	-
dna	.952	.558$^-$	.960$^+$	.953	.959$^+$	.936$^-$	.958	.960	**.962**	.959	.959	-
segment	.959	.971$^+$	.972$^+$	.975$^+$	.971	.975$^+$	.972	.972	**.976**	.973	.975	-
w1a	.981	.973$^+$	.979	**.981**	.981	.979	.981	**.981**	.981	.980	.980	-
astro	.955	.967$^+$	.968$^+$	.967$^+$	.969$^+$	**.971**$^+$	.966	.967	.968	.969$^+$	.969$^+$	-
# pos. diff.		12(7)	15(10)	18(9)	19(9)	13(10)		11(0)	10(0)	15(2)	13(2)	9(1)
# neg. diff.		8(2)	7(2)	4(0)	2(0)	7(1)		9(0)	11(1)	4(0)	4(0)	8(1)
Wsr test			✓	✓	✓	✓				✓	✓	

dataset	$k = k^{pol}$						$k = k^{sig}$					
	k	$\mathcal{E}_\sigma k$	$\mathcal{P}_\sigma k$	$\mathcal{S}_{\sigma,\eta} k$	$\mathcal{PS}_{\sigma,\eta} k$	KNNSVM	k	$\mathcal{E}_\sigma k$	$\mathcal{P}_\sigma k$	$\mathcal{S}_{\sigma,\eta} k$	$\mathcal{PS}_{\sigma,\eta} k$	KNNSVM
leukemia	**.947**	.711$^-$	.763$^+$	**.947**	**.947**	**.947**	.711	.711	.711	.658	.658	.789
iris	**.973**	.960	.960	.947	.947	.960	.960	.967	.953	.960	.960	.960
wine	.961	.961	.978	.966	.961	.966	.972	.983	**.989**	.972	.978	.966
sonar	.851	.716$^-$	.861	.846	.846	.885	.750	.899$^+$	.880$^+$	.894$^+$	.870$^+$	.885$^+$
glass	.701	.701	.701	.706	.696	.720	.626	.678$^+$	.664	.682$^+$	.696$^+$	**.738**$^+$
heart	.819	.785	.796	**.833**	.822	.811	.830	.811	.815	**.833**	.830	.819
liver	.725	.690	.716	.728	.722	.730	.672	.733$^+$	.716	**.739**$^+$	.704	.722$^+$
ionosphere	.900	.766$^-$	.926	.923	.926	.934	.872	.943$^+$	.946$^+$	.949$^+$	**.954**$^+$	.943$^+$
bioinf	.821	.770	.818	.818	.824	**.857**$^+$	.829	.795	.836	.831	.839	.852
vowel	.973	.987	.987	.991$^+$	.992$^+$	.994$^+$	.799	.991$^+$	.991$^+$	.991$^+$	.992$^+$	**.996**$^+$
breast	.963	.962	.960	.958	.960	.956	.975	.975	**.978**	.972	.969	.958$^-$
australian	.851	.854	.843	.851	.851	.852	.849	.849	.854	.851	.848	**.868**$^+$
diabetes	.767	.760	**.779**	.766	.763	.766	.758	.768	.773	.759	.767	**.779**$^+$
vehicle	.846	.818	.833	.846	.839	-	.787	.852$^+$	.839$^+$	.851$^+$	.830$^+$	-
fourclass	.799	.997$^+$	.964$^+$	.995$^+$	.959$^+$	.998$^+$	.776	**1.00**$^+$	.911$^+$	**1.00**$^+$	.818$^+$	.999$^+$
splice	.862	.828	.878	.862	.876	-	.805	.876$^+$	.865$^+$	.867$^+$	.841$^+$	-
numer	.766	.741	.723	.767	**.771**	-	.766	.734$^-$	.751	.766	.755	.758
vehicle2	.850	.797$^-$	.844	**.851**	.850	-	.822	.845	.846$^+$	.846$^+$	.826	-
a1a	.828	.814	.809	.827	.830	-	**.833**	.828	.826	.822	.822	-
dna	.958	.910	.958	.958	.958	-	.949	.960$^+$	.959$^+$	.956$^+$	.956	-
segment	.970	.966	.973	.972	.972	-	.947	.974$^+$	.972$^+$	.972$^+$	.975$^+$	-
w1a	.980	.974$^-$	.981	.980	.980	-	.981	.980	.980	.979	.979	-
astro	.968	.967	.965	.965	.968	.969	.954	.967$^+$	.968$^+$	**.971**$^+$	.970$^+$	-
# pos. diff.		6(1)	10(2)	10(2)	10(2)	10(3)		15(11)	17(10)	16(12)	15(9)	10(8)
# neg. diff.		15(5)	12(0)	7(0)	8(0)	4(0)		5(1)	5(0)	4(0)	6(0)	4(1)
Wsr test								✓	✓	✓	✓	✓

- + and - denotes quasi-local kernel and KNNSVM results that are significatively better (or worse) than the corresponding input kernels according to the two-tailed paired t-test ($\alpha = 0.05$);
- **# pos. diff.** and **# neg. diff.** denote, for each quasi-local kernel and KNNSVM methods, the number of datasets in which they perform better (or worse) than the corresponding input kernels. In parenthesis are reported the statistically significative differences;
- **Wsr test** marks the cases in which the Wilcoxon signed-ranks tests states that the improvements of quasi-local kernels over corresponding input kernels on all the datasets are significant ($\alpha = 0.05$);
- underlined are the cases in which, for $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$, the lowest empirical risk is achieved with $\eta = 0$ for all the 10 folds;
- in **bold**, are highlighted the best 10-fold cross validation accuracies achieved for a specific dataset among all methods and kernels.

kernels obtained with $\mathcal{P}_\sigma$ achieve the best accuracies for 8 datasets, meaning that this operator is able to reach very good results but the improvements are not systematic for all the input kernels. It is possible to notice that the classification results of $\mathcal{P}_\sigma\, k$ are very similar to the kNNSVM ones (both improve significantly over SVM with the k^{lin} and k^{sig} but not for k^{rbf} and k^{pol}).

The best results are clearly achieved by the $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ operators without significative differences between them. According to the Wilcoxon signed-rank test they significantly improve the generalization accuracy for k^{lin}, k^{rbf} and k^{sig}. Moreover, they are the only operators that never cause significant 10-fold CV losses according to the statistical two-tailed paired t-test, while the number of improvements are impressive at least for k^{lin} and k^{sig}. The only kernel that seems not to take a decisive advantage from the two operators is k^{pol} that, together with the results noticed for KNNSVM with the same input kernel, lead us to argue that, at least for non-large datasets, locality is not a crucial point for the polynomial kernels. Comparing $\mathcal{S}_{\sigma,\eta}\, k$ and $\mathcal{PS}_{\sigma,\eta}$ with KNNSVM we can notice that the operator approach performs better in terms of 10-fold CV accuracies (especially for k^{rbf}).

We do not discuss directly the computational performances of the operators in this experiment. However, we can notice that they are much faster, as expected, than KNNSVM since, in total, 25 KNNSVM results are missing due to computational difficulties (the computation does not finish within 72 hours) whereas SVM with input and quasi-local kernels always terminate in a reasonable time.

6 Experiment 2

The second experiment applies the SVM with the quasi-local kernels that, in the exploratory Experiment 1, seem to achieve better accuracy values, i.e. $\mathcal{S}_{\sigma,\eta}\, k$ and $\mathcal{PS}_{\sigma,\eta}\, k$. The aim of this experiment is to verify if these kernels are able to improve the input kernel classification accuracy in a considerable number of large datasets without worsening dramatically the computational performances.

6.1 Experimental procedure

The 20 datasets used in the second experiment are summarized in Table 3; they are all the datasets with more than 3500 samples available on the LibSVM repository [48] (except the *mushrooms* dataset for which perfect classification is already easily achievable for all the input kernels) and the UCI datasets for classification with only numerical values, available test labels, and more than 3500 training samples. The datasets are quite large and for this reason kernels resulting from the four chosen operators with the four input kernels are simply trained on the training set and tested on the testing set. If no separate testing sets are provided we use 75% of available data (randomly selected) for training and the remaining 25% for testing, apart for the *covertype* from which we randomly selected 100000 samples leaving the remaining 481012 in the testing set for computational reasons. We normalized the data in the range $[0, 1]$. With this approach the t-tests are not suitable, and the best way to assess statistical significance is the Wilcoxon signed rank test as detailed in [57].

The model selection is performed with 10-fold CV with the same approach of Experiment 1 and with the same ranges of parameter values. We do not test the KNNSVM classifier because of the computational weight of the method.

Table 3: The 20 datasets for Experiment 2 ordered by training set size.

Name	brief description	# classes	train. size	test. size	# features
optdigit	optical recognition of handwritten digits	10	3823	1797	64
blocks	segmented page blocks classification	5	4107	1368	10
satimage	Landsat satellite data, orig. from Statlog [51]	6	4435	2000	36
musk2	musks/non-musks molecule prediction, ver. 2	2	4949	1649	166
isolet	spoken letter prediction	26	6238	1559	617
usps	handwritten text recognition	10	7291	2007	256
magic	high energy gamma particles detection	2	14265	4755	10
letter	letter recognition, orig. from Statlog [51]	26	15000	5000	16
news20	newsgroup classification, preproc. as [58]	20	15935	3993	62061
protein	protein classification task	3	17766	6621	357
rcv1	two class version of Reuters Corpus Vol. I	2	20242	677399	47236
mnist1	handwritten digits bloblem, preproc. as [59]	10	21000	49000	780
a9a	Adult dataset preprocessed as done by [54]	2	32561	16281	123
shuttle	the shuttle dataset, orig. from Statlog [51]	7	43500	14500	9
w8a	web page classification, orig. from [54]	2	49749	14951	300
ijcnn1	IJCNN 2001 challenge, preproc. as [59]	2	49990	91701	22
connect4	connect4 result prediction (binary encoding)	3	50668	16889	126
acoustic	vehicle classification from acoustic sensors	3	78823	19705	50
acoustic	vehicle classification from seismic sensors	3	78823	19705	50
covertype	two classes forest cover type prediction	2	100000	481012	54

Table 4: Experiment 2. Generalization accuracy of SVM with the input and quasi-local kernels.

dataset	$k = k^{lin}$			$k = k^{rbf}$			$k = k^{pol}$			$k = k^{sig}$		
	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$
optdigit	.9672	.9777	**.9855**	.9816	.9816	.9816	.9750	.9750	.9822	.9666	.9839	.9800
blocks	.9678	.9715	.9715	.9635	.9635	.9635	.9671	.9722	**.9759**	.9591	.9686	.9686
satimage	.8580	.9150	.9180	.9190	**.9230**	.9210	.8880	.9120	.9105	.8570	.9215	.9135
musk2	.9551	.9982	.9982	.9970	.9976	**.9988**	.9970	.9970	.9964	.9260	.9951	.9715
isolet	.9596	.9609	.9673	.9666	.9679	.9679	.9628	.9628	**.9686**	.8012	.8518	.8621
usps	.9357	.9472	.9522	.9527	.9547	**.9557**	.9397	.9422	.9452	.9243	.9532	.9507
magic	.7868	.8694	.8751	.8755	.8763	.8755	**.8776**	.8763	**.8776**	.7865	.8700	.8574
letter	.8512	.9774	.9766	.9748	.9776	**.9778**	.9556	.9692	.9708	.8516	.9768	.9740
news20	.8550	.8550	.8550	.8257	.8257	.8257	.7626	.7744	.7626	**.8610**	**.8610**	**.8610**
protein	.6865	.6868	**.7041**	.7026	.7005	.6987	.6919	.6926	.6919	.6865	.6981	.6874
rcv1	.9605	.9570	**.9637**	.9455	.9426	.9405	.9545	.9478	.9545	.9604	.9542	.9622
mnist1	.9367	.9525	.9747	.9735	.9749	**.9754**	.9708	.9710	.9733	.9044	.9547	.9530
a9a	.8498	**.8511**	.8498	.8502	.8509	.8502	.8477	.8479	.8477	.8498	.8498	.8496
shuttle	.9794	**.9993**	.9992	.9990	.9992	.9992	.9987	**.9993**	.9988	.9757	.9991	.9988
w8a	.9868	.9944	**.9945**	.9910	.9914	.9919	.9924	.9944	.9924	.9858	.9909	.9886
ijcnn1	.9218	.9787	.9748	.9758	.9814	**.9824**	.9676	.9665	.9676	.9203	.9786	.9631
connect4	.7591	.8421	.8600	**.8623**	.8623	.8623	.8441	.8441	.8588	.7476	.8074	.7939
acoustic	.7024	.7997	.8001	.7987	.7999	**.8004**	.7984	.7986	.7993	.7020	.7988	.7845
seismic	.7281	.7694	.7697	.7698	.7698	.7698	.7658	.7658	.7658	.6976	**.7701**	.7486
covertype	.7629	.9098	.9121	.9077	**.9202**	.9187	-	-	-	.6286	.8732	.8629
# pos. diff.		18	18		13	11		14	10		17	18
# neg. diff.		1	0		2	2		3	1		1	1
Wsr test		✓	✓		✓	✓		✓	✓		✓	✓
avg rank	9.70	5.35	3.70	5.50	3.88	3.88	8.10	7.05	6.55	10.60	5.75	7.95

- + and - denoting statistical significance on single datasets are not present here (differently from Table 2) because, due to the dimension of the problems of this experiment, we have single testing sets and thus t-test are not applicable;
- **# pos. diff.** and **# neg. diff.** denote, for each quasi-local kernel, the number of datasets in which they perform better (or worse) than the corresponding input kernels;
- **Wsr test** marks the cases in which the Wilcoxon signed-ranks tests states that the improvements of quasi-local kernels over corresponding input kernels on all the datasets are significant ($\alpha = 0.05$);
- underlined are the cases in which, for $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$, the lowest empirical risk is achieved with $\eta = 0$;
- in **bold**, are highlighted the best generalization accuracies achieved for a specific dataset among all methods and kernels;
- missing values correspond to kernels for which model selection was not completed because for some parameter values the training time for a single fold takes more than 72 hours.
- **avg rank** reports the average rank of the methods.

Table 5: Experiment 2. Training times (in seconds) of SVM with the input and quasi-local kernels.

dataset	$k = k^{lin}$			$k = k^{rbf}$			$k = k^{pol}$			$k = k^{sig}$		
	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$
optdigit	1	1	1	2	2	2	1	1	1	1	3	3
blocks	1	1	1	1	1	1	2	1	1	1	2	2
satimage	1	2	2	3	3	4	9	**2**	**2**	2	6	4
musk2	16	**5**	**4**	4	4	4	5	7	4	8	7	8
isolet	27	32	53	50	61	70	31	32	59	123	71	72
usps	9	14	11	11	15	17	7	10	20	12	39	13
magic	164	129	138	99	146	154	3867	2360	6666	63	99	99
letter	12	12	14	14	24	28	13	12	23	21	46	37
news20	284	325	407	359	436	435	561	639	668	271	430	472
protein	354	377	431	440	499	532	410	460	552	424	611	590
rcv1	108	124	189	165	196	208	269	296	337	129	214	219
mnist1	93	99	131	124	153	174	94	101	155	213	185	203
a9a	220	290	460	196	263	270	219	280	*970*	259	590	547
shuttle	79	**6**	**5**	6	6	6	27	**4**	**8**	101	**30**	69
w8a	1292	**84**	**81**	59	90	95	40	*131*	53	55	157	124
ijcnn1	189	109	102	95	85	93	298	267	634	290	282	388
connect4	1219	1380	2419	1608	2469	3225	2913	3255	2985	1403	2074	2194
acoustic	7794	10972	10120	3143	4180	5398	2661	3459	3741	4544	8296	5689
seismic	10045	19704	21466	4160	5909	7144	5670	7190	8340	3958	8992	6164
covertype	19106	36914	*97532*	15506	22319	24694				2745	6283	5546
# pos. diff.		5(3)	5(3)		1(0)	1(0)		6(2)	4(2)		5(1)	3(0)
# neg. diff.		12(0)	13(1)		14(0)	15(0)		12(1)	14(1)		15(1)	16(0)

- **# pos. diff.** and **# neg. diff.** denote, for each quasi-local kernel, the number of datasets in which they are faster (or slower) than the corresponding input kernels. In parenthesis are reported the differences greater than 3 times;
- in **bold**, are the cases in which the quasi-local kernels are at least three times faster than the corresponding input kernel;
- in *italic*, are the cases in which the quasi-local kernels are at least three times slower than the corresponding in put kernel;
- missing values correspond to kernels for which model selection was not completed because for some parameter values the training time for a single fold takes more than 72 hours.

6.2 Results

Table 4 shows the generalization accuracy results of the input kernels k and of the quasi-local kernels $\mathcal{S}_{\sigma,\eta}k$ and $\mathcal{PS}_{\sigma,\eta}k$ on all the 20 datasets listed in Table 3. We report the number of datasets in which quasi-local kernels perform better (or worse) than the corresponding input kernels, the Wilcoxon signed rank test to asses the statistical significance of differences between them, and the average rank of each method. The cases for which model selection for quasi-local kernels chooses $\eta = 0$ thus obtaining the same model of the SVM with the input kernel are underlined. In bold are highlighted the best generalization accuracies achieved for each dataset. Notice that the results regarding the *covertype* dataset with the K^{pol} kernel are missing because of its excessive computational weight (especially for high degrees of the kernel) causes the model selection to take more than 72 hours to be completed.

The training and testing times, expressed in seconds, are reported in Table 5 and Table 6 respectively. We point out the number of times SVM with quasi local kernels are faster and slower than the corresponding input kernels and (in parenthesis) the number of times SVM with quasi local kernels are three times faster and slower than the corresponding input kernels (these big variations are highlighted in bold and italic).

6.3 Discussion

Quasi-local kernels perform better than the corresponding input kernels in terms of generalization accuracy with statistical significance as reported in Table 4, for all the input kernels

Table 6: Experiment 2. Testing times (in seconds) of SVM with the input and quasi-local kernels.

dataset	$k = k^{lin}$			$k = k^{rbf}$			$k = k^{pol}$			$k = k^{sig}$		
	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$	k	$\mathcal{S}_{\sigma,\eta}k$	$\mathcal{PS}_{\sigma,\eta}k$
optdigit	1	1	1	1	1	2	1	1	1	1	3	2
blocks	1	1	1	1	1	1	1	1	1	1	1	1
satimage	1	1	2	2	3	2	1	1	2	2	4	4
musk2	1	2	1	1	1	1	1	1	2	3	3	2
isolet	20	22	25	26	28	29	21	21	26	40	35	35
usps	4	8	6	6	6	8	4	5	7	5	15	7
magic	7	6	7	7	6	7	4	8	7	15	25	28
letter	15	17	19	19	25	26	10	19	22	27	45	42
news20	56	64	72	69	77	77	71	81	82	60	90	94
protein	107	117	125	125	142	150	119	133	151	130	189	188
rcv1	3355	3893	5932	5194	6195	6558	8142	9038	10284	4055	6827	6978
mnist1	419	451	533	529	617	667	368	398	552	820	763	787
a9a	64	89	93	84	107	107	64	89	98	117	238	238
shuttle	14	**1**	**1**	1	1	1	1	1	1	34	**6**	20
w8a	2	*18*	*20*	12	20	22	4	*26*	11	6	*34*	*24*
ijcnn1	239	156	103	166	162	160	28	50	26	456	443	700
connect4	280	273	235	237	274	306	187	222	291	371	559	596
acoustic	589	542	540	534	618	627	461	586	580	854	1134	1256
seismic	546	566	567	561	662	673	477	608	670	791	1195	1279
covertype	6813	4619	4129	4643	5076	4926				7982	17135	18469
# pos. diff.		6(1)	5(1)		2(0)	1(0)		0(0)	1(0)		4(1)	4(0)
# neg. diff.		11(1)	11(1)		13(0)	14(0)		13(1)	15(0)		14(1)	15(1)

- **# pos. diff.** and **# neg. diff.** denote, for each quasi-local kernel, the number of datasets in which they are faster (or slower) than the corresponding input kernels. In parenthesis are reported the differences greater than 3 times;
- in **bold**, are the cases in which the quasi-local kernels are at least three times faster than the corresponding input kernel;
- in *italic*, are the cases in which the quasi-local kernels are at least three times slower than the corresponding in put kernel;
- missing values correspond to kernels for which model selection was not completed because for some parameter values the training time for a single fold takes more than 72 hours.

taken into account. The number and the magnitude of the improvements are particularly large for the k^{lin} and k^{sig} input kernels. This because they are global kernels that in general (a part for k^{lin} in presence of a high-dimensional problems) are not able to achieve very high accuracy results, and thus the addition of the local information is almost always crucial. We can notice that, for these large datasets, the operators are able to improve the generalization accuracies also for he k^{pol} kernel differently from Experiment 1. Looking at the average ranks of all the methods, we can see that the methods achieving the best results are $\mathcal{PS}_{\sigma,\eta}\, k^{lin}$, $\mathcal{PS}_{\sigma,\eta}\, k^{rbf}$ and $\mathcal{S}_{\sigma,\eta}\, k^{rbf}$. On the other hand, apart k^{rbf} whose average rank is near the mean position (6), the other three input kernels have the worst average ranks. Looking at the best result for each dataset (bold values in Table 4), we can notice that $\mathcal{PS}_{\sigma,\eta}\, k^{rbf}$ is the kernel that permits the highest number of best generalization accuracies (about for one third of datasets), whereas the input kernels rarely achieve the best results. Compared to $\mathcal{S}_{\sigma,\eta}$, $\mathcal{PS}_{\sigma,\eta}$ seems to be a more "extreme" approach in the sense that it achieves the best results more frequently but at the same time there are more cases in which $\eta = 0$ is selected meaning that the input kernel has an higher training set accuracy. For this reason we can hypothesize that $\mathcal{PS}_{\sigma,\eta}$ introduces an higher level of locality than $\mathcal{S}_{\sigma,\eta}$. From the above considerations, we can conclude that the $\mathcal{S}_{\sigma,\eta}$ and $\mathcal{PS}_{\sigma,\eta}$ operators are able to significantly improve the generalization ability of traditional kernels, and, in particular, the kernels that show the best accuracies and can be thus indicated as good candidate kernels for general classification problems, are $\mathcal{PS}_{\sigma,\eta}\, k^{lin}$, $\mathcal{PS}_{\sigma,\eta}\, k^{rbf}$ and $\mathcal{S}_{\sigma,\eta}\, k^{rbf}$.

Observing the computational performances of quasi-local kernels in Table 5 and Table 6,

we can notice that both the training and testing times are slightly higher than input kernels. This is not surprising as the quasi-local transformation introduce inevitably and systematically a considerable overhead in kernel computation. However, there is a consistent number of cases in which quasi-local kernels are faster than the corresponding input kernel. This is due to the fact that quasi-local kernels can have more discriminative power and thus they can execute the SVM margin maximization with a smaller number of optimization steps. In general, from the results, we can conclude that the quasi-local kernels are very rarely more than three times slower in comparison with the input kernels, and in few cases they are more than three times faster. This means that although they introduce a certain overhead on kernel computation, the SVM performances are not dramatically deteriorated by the quasi-local transformation of kernel functions.

7 Conclusions

In this paper, we have presented a novel family of operators on kernels that add locality information to the input kernel. The resulting kernels are called quasi-local kernels since they balance the global information of the original kernel (if it is a non-local kernel) with the local kernel with respect to the distance in the feature space. The intuition is that the resulting kernels are able to maintain the original kernel behaviour for regions in which the information is not local, adapting instead the separating hyperplane following the local distribution of the data. We formally characterize the class of quasi-local kernels, showing that they are assured to be positive-definite. Moreover, we showed that the $\mathcal{E}_\sigma$ operator, on which the quasi-local kernels are based, defines the same neighborhoods as the input kernel, that, applied to the k^{rbf} its behaviour is equivalent to a k^{rbf} with variable kernel width and we detailed a data-dependent strategy to choose the operator parameters.

The empirical evaluation on a total of 43 datasets carried out transforming the optimized input kernel performing a reduced model selection (no more than 12 parameter choices), showed that the quasi-local kernel are able to significantly improve the classification accuracies of the input kernels. In particular, $(\mathcal{S}_{\sigma,\eta}\, k)(x, x') = k(x, x') + \eta \cdot \exp\left(\frac{-k(x,x)-k(x',x')+2k(x,x')}{\sigma}\right)$ and $(\mathcal{P}_{\sigma,\eta}\, k)(x, x') = k(x, x') \cdot \left(1 + \eta \cdot \exp\left(\frac{-k(x,x)-k(x',x')+2k(x,x')}{\sigma}\right)\right)$ showed solid statistical evidence of improved generalization capability over input kernels especially for large datasets. Considering the k^{lin}, k^{rbf}, k^{pol} and k^{sig} input kernels, the present work suggests that the best classification accuracies are achieved by $\mathcal{P}_{\sigma,\eta}\, k^{rbf}$, $\mathcal{P}_{\sigma,\eta}\, k^{lin}$ and $\mathcal{S}_{\sigma,\eta}\, k^{rbf}$. We also showed that the computational performances of quasi-local kernels are not dramatically deteriorated with respect to the corresponding input kernels.

Generally speaking, the idea highlighted in this work is that, especially for large and complex problems, the true class boundary reflects a global behaviour that can be estimated using a proper kernel function but is very likely to have local adaptations and modifications. These local anomalies can be detected and introduced in the learning process mainly relying on the sample distribution of the subregions. Combining global and high-level information with local and data-dependent analysis can be seen as a strategy that aims to "attack complex worlds" which is, according to a recent interview with prof. Vapnik[3], the main challenge machine learning still has to address.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, no. 3, pp. 273–297, 1995.

[3] Vladimir Vapnik, "Learning Has Just Started", Computational Learning Theory (COLT) available at `http://www.learningtheory.org`

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2000.

[3] B. Schölkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[4] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput*, vol. 4, no. 6, pp. 888–900, 1992.

[5] V. Vapnik and L. Bottou, "Local Algorithms for Pattern Recognition and Dependencies Estimation," *Neural Comput*, vol. 5, no. 6, pp. 893–909, 1993.

[6] B. V. Dasarathy, *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos: IEEE Computer Society Press, 1990, 1990.

[7] B. Scholkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," *Adv Neural Inf Process Syst*, vol. 10, pp. 640–646, 1998.

[8] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K. Müller, "Engineering support vector machine kernels that recognize translation initiation sites," *Bioinformatics*, vol. 16, no. 9, pp. 799–807, 2000.

[9] Y. Fu, Q. Yang, R. Sun, D. Li, R. Zeng, C. Ling, and W. Gao, "Exploiting the kernel trick to correlate fragment ions for peptide identification via tandem mass spectrometry," *Bioinformatics*, vol. 20, no. 12, pp. 1948–1954, 2004.

[10] Y. Bengio, O. Delalleau, and N. Le Roux, "The curse of dimensionality for local kernel machines," Departement dinformatique et recherche operationnelle, Universite de Montreal, Tech. Rep. 1258, 2005.

[11] G. Smits and E. Jordaan, "Improved SVM regression using mixtures of kernels," *Proc. of the 2002 International Joint Conference on Neural Networks (IJCNN'02)*, vol. 3, 2002.

[12] Y. Bengio, O. Delalleau, and N. Le Roux, "The curse of highly variable functions for local kernel machines," *Adv Neural Inf Process Syst*, vol. 18, pp. 107–114, 2006.

[13] E. Blanzieri and F. Melgani, "An adaptive SVM nearest neighbor classifier for remotely sensed imagery," *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS-2006)*, pp. 3931–3934, 2006.

[14] ——, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans Geosci Remote Sens*, vol. 46, no. 6, pp. 1804–1811, 2008.

[15] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, vol. 2, 2006.

[16] N. Segata and E. Blanzieri, "Empirical assessment of classification accuracy of Local SVM," Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Italy, Tech. Rep. DISI-08-014, 2008.

[17] E. Blanzieri and A. Bryl, "Evaluation of the highest probability SVM nearest neighbor classifier with variable relative error cost," in *CEAS 2007*, Mountain View, California, Aug 2007.

[18] N. Segata, E. Blanzieri, S. Delany, and P. Cunningham, "Noise reduction for instance-based learning with a local maximal margin approach," Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Italy, Tech. Rep. DISI-08-056, 2008, under submission.

[19] H. Cheng, P. Tan, and R. Jin, "Localized Support Vector Machine and Its Efficient Algorithm," *Proc. SIAM IntlConf. Data Mining*, 2007.

[20] N. Segata and E. Blanzieri, "Fast local support vector machines for large datasets," Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Italy, Tech. Rep. DISI-08-063, 2008.

[21] S. Wu and S. Amari, "Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers," *Neural Process Lett*, vol. 15, no. 1, pp. 59–67, 2002.

[22] H. Xiong, Y. Zhang, and X. Chen, "Data-dependent kernel machines for microarray data classification," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 4, no. 583-595, p. 1, 2007.

[23] D. Lewis, T. Jebara, and W. Noble, "Nonstationary kernel combination," in *Proceedings of the 23rd international conference on Machine learning*. ACM Press New York, NY, USA, 2006, pp. 553–560.

[24] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

[25] D. DeCoste, "Visualizing Mercel kernel feature spaces via kernelized locally linear embedding," in *Proceedings of the Eighth International Conference on Neural Information Processing (ICONIP-01)*, 2001.

[26] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput*, vol. 10, no. 5, pp. 1299–1319, 1998.

[27] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 2005, pp. 846–853 vol. 2.

[28] R. Fisher, "The use of multiple measurements in taxonomic problems," *Ann Eugen*, vol. 7, no. 2, pp. 179–188, 1936.

[29] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, 1999, pp. 41–48.

[30] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *ICML '06: Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA: ACM Press, 2006, pp. 905–912.

[31] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput*, vol. 12, no. 10, pp. 2385–2404, 2000.

[32] T.-K. Kim and J. Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE Trans Pattern Anal Mach Intell*, vol. 27, no. 3, pp. 318–327, 2005.

[33] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction." *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[34] H. Choi and S. Choi, "Robust kernel isomap," *Pattern Recognit*, vol. 40, no. 3, pp. 853–862, March 2007.

[35] X. He, S. Yan, Y. Hu, and H.-J. Zhang, "Learning a locality preserving subspace for visual recognition," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 385–392 vol.1.

[36] V. De Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Adv Neural Inf Process Syst*, vol. 15, 2003, pp. 705–712.

[37] N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press New York, NY, USA, 1999.

[38] A. J. Smola, Personal communication.

[39] B. Schölkopf, *Support Vector Learning*. R. Oldenbourg Verlag, 1997.

[40] H. Lin and C. Lin, "A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods," National Taiwan University, Tech. Rep., 2003.

[41] B. Schölkopf, "The kernel trick for distances," *Adv Neural Inf Process Syst*, vol. 13, pp. 301–307, 2001.

[42] V. Vapnik, "Principles of risk minimization for learning theory," in *NIPS*, 1991, pp. 831–838.

[43] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J Mach Learn Res*, vol. 2, pp. 67–93, 2002.

[44] ——, "Support Vector Machines are Universally Consistent," *J Compl*, vol. 18, no. 3, pp. 768–791, 2002.

[45] ——, "Consistency of support vector machines and other regularized kernel classifiers," *IEEE Trans Inf Theory*, vol. 51, no. 1, pp. 128–142, 2005.

[46] C. Micchelli, Y. Xu, and H. Zhang, "Universal Kernels," *J Mach Learn Res*, vol. 7, pp. 2651–2667, 2006.

[47] Q. Chang, Q. Chen, and X. Wang, "Scaling gaussian rbf kernel width to improve svm classification," in *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on*, vol. 1, 2005, pp. 19–22.

[48] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[49] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, p. 531, 1999.

[50] M. Duarte and Y. Hen Hu, "Vehicle classification in distributed sensor networks," *J Parallel Distr Comput*, vol. 64, no. 7, pp. 826–838, 2004.

[51] R. King, C. Feng, and A. Sutherland, "Statlog: comparison of classification algorithms on large real-world problems," *Appl Artif Intell*, vol. 9, no. 3, pp. 289–333, 1995.

[52] C. Hsu, C. Chang, C. Lin *et al.*, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, Tech. Rep., 2003.

[53] T. Ho and E. Kleinberg, "Building projectable classifiers of arbitrary complexity," *Proc. of the 13th International Conference on Pattern Recognition (ICPR-96)*, vol. 2, p. 880, 1996.

[54] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," *MIT Press Cambridge, MA, USA*, pp. 185–208, 1999.

[55] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans Neural Netw*, vol. 13, no. 2, pp. 415–425, 2002.

[56] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J Mach Learn Res*, vol. 7, pp. 1–30, 2006.

[57] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[58] K. Lang, "Newsweeder: Learning to filter netnews." in *Proc. of the 12th International Machine Learning Conference*, 1995.

[59] K. M. Lin and C. J. Lin, "A study on reduced support vector machines." *IEEE Trans Neural Netw*, vol. 14, no. 6, pp. 1449–1459, 2003.