UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

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

EMPIRICAL ASSESSMENT OF CLASSIFICATION ACCURACY OF LOCAL SVM

Nicola Segata and Enrico Blanzieri

March 2008

Technical Report # DISI-08-014

Empirical Assessment of Classification Accuracy of Local SVM

Nicola Segata and Enrico Blanzieri*

March 21, 2008

Abstract

The combination of maximal margin classifiers and k-nearest neighbors rule constructing an SVM on the neighborhood of the test sample in the feature space (called kNNSVM), was presented as a promising way of improving classification accuracy. Since no extensive validation of the method was performed yet, in this work we test the kNNSVM method on 13 widely used datasets using four different kernels obtaining good classification results. Moreover we present two artificial datasets in which kNNSVM performs substantially better than SVM with RBF kernel. Statistically significant testing of the method as well as the results on the artificial datasets, lead us to conclude that kNNSVM performs sensibly better than SVM.

1 Introduction

The idea of combining directly the state-of-the-art classification method of SVM with the simple but still popular and effective method of kNN has been presented in [2]. The algorithm is called kNNSVM, and it builds a maximal margin classifier on the neighborhood of a test sample in the feature space induced by a kernel function. An important property of kNNSVM which theoretically permits better generalization power is that it can have, for some values of k, a lower radius/margin bound with respect to SVM. In [14] is proposed a similar method in which however the distance function for the nearest neighbors rule is performed in the input space and it is approximated in order to improve the computational performances. An interesting method that includes locality in kernel machines for regression has been recently presented in [8], proposing a way of weighting the loss parameter with a kernel function inspired to kNN.

Even if the kNNSVM has been successfully applied on two specific classification tasks (remote sensing in [2] and visual category recognition in [14]), no extensive testing has been performed in order to assess the classification performance of the method against SVM for general classification problems and for different kernels. The issue is theoretically relevant because it would indicate locality as a way for improving SVM accuracies.

In this work we empirically compare the classification performance of kNNSVM and SVM on 13 datasets taken from different application domains and with 4 kernel functions. The comparison confirms the better classification capabilities assured by the lower radius/margin bound of kNNSVM. The RBF kernel is also studied with two artificial datasets. The paper is organized as follows. After recalling the kNN and SVM methods (Section 2) we describe the kNNSVM classifier (Section 3). In Section 4 we detail the empirical testing of kNNSVM with respect to SVM and in Section 5 we discuss the comparison of the methods with RBF kernel by means of two artificial datasets. Finally we draw some conclusions.

*N. Segata and E. Blanzieri are with the Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Italy. E-Mail: {segata, blanzier}@disi.unitn.it.

2 Nearest neighbors and SVM

k nearest neighbors classifier. Let assume to have a classification problem with samples (x_i, y_i) with $i = 1, \ldots, N$, $x_i \in \mathbb{R}^p$ and $y_i \in \{+1, -1\}$. Given a point x', it is possible to order the entire set of training samples X with respect to x'. This corresponds to define a function $r_{x'} : \{1, \ldots, N\} \to \{1, \ldots, N\}$ that reorders the indexes of the N training points:

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

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', namely the j-th nearest neighbor, $\|x_{r_{x'}(j)} - x'\|$ is its distance from x' and $y_{r_{x'}(j)}$ is its class with $y_{r_{x'}(j)} \in \{-1, 1\}$. In other terms: $j < k \Rightarrow \|x_{r_{x'}(j)} - x'\| \leq \|x_{r_{x'}(k)} - x'\|$.

Given the above definition, the majority decision rule of kNN for binary classification problems is defined by

$$kNN(x) = sign\left(\sum_{i=1}^{k} y_{r_x(i)}\right).$$

Support vector machines. SVMs [6] are classifiers based on statistical learning theory [13]. The decision rule is $SVM(x) = sign(\langle w, \Phi(x)\rangle_{\mathcal{F}} + b)$ where $\Phi(x) : \mathbb{R}^p \to \mathcal{F}$ is a mapping in a 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:

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

where $y_i \in \{-1, +1\}$ is the class label of the i-th nearest training sample. The presence of the slack variables ξ_i's allows some misclassification on the training set. Reformulating such an optimization problem with Lagrange multipliers α_i $(i = 1, \ldots, N)$, and introducing a positive definite kernel (PD) function[1] $K(\cdot, \cdot)$ 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:

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

The introduction of PD kernels avoids the explicit definition of the feature space $\mathcal{F}$ and of the mapping Φ [12]. Popular kernels are the linear (LIN) kernel, the radial basis function (RBF) kernel, and the homogeneous (HPOL) and inhomogeneous (IPOL) polynomial kernels. Their definition are:

$$\begin{array}{ll} k^{lin}(x, x') = \langle x, x'\rangle & k^{rbf}(x, x') = \exp \frac{\|x - x'\|^2}{\sigma} \\ k^{hpol}(x, x') = \langle x, x'\rangle^d & k^{ipol}(x, x') = (\langle x, x'\rangle + 1)^d \end{array}$$

The maximal separating hyperplane defined by SVM has been shown to have important generalization properties and nice bounds on the VC dimension [13]. In particular we refer to the following theorem:

[1] For convention we refer to kernel functions with the capital letter K and to the number of nearest neighbors with the lower-case letter k.

Theorem 1 ([13] 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 R^2/Δ^2.

3 The kNNSVM classifier

The method [2] 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. In order to classify a given point x' of the input 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 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 training points defined as:

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

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 thus $j < k \Rightarrow \|\Phi(x_{r_{x'}(j)}) - \Phi(x')\| \leq \|\Phi(x_{r_{x'}(k)}) - \Phi(x')\|$ because of the monotonicity of the quadratic operator. The computation is expressed in terms of kernels as:

$$\begin{array}{rl} & ||\Phi(x) - \Phi(x')||^2 = \\ = & \langle\Phi(x), \Phi(x)\rangle_{\mathcal{F}} + \langle\Phi(x'), \Phi(x')\rangle_{\mathcal{F}} + \\ & -2 \cdot \langle\Phi(x), \Phi(x')\rangle_{\mathcal{F}} = \\ = & K(x, x) + K(x', x') - 2 \cdot K(x, x'). \end{array} \tag{2}$$

In the case of the LIN kernel, the ordering function can be built using the Euclidean distance, whereas if the kernel is not linear, the ordering can be different. If the kernel is the RBF kernel the ordering function is equivalent to using the Euclidean metric.

The decision rule associated with the method is:

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

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

This method is rather computationally expensive because, for each test point, it is necessary to compute the k nearest neighbors in the transformed space, to train an SVM (which is however faster than an SVM trained on the complete training set) and finally to perform the SVM prediction step.

Table 1: The 13 datasets used in the experiments. The references to the sources are: UCI [1], TG99 [7], Statlog [11], CWH03a [10], TKH96a [9]. Number of classes, training set cardinality and number of features are reported.

DATASET NAME	SOURCE	#CL	#TR	#F
IRIS	UCI	3	150	4
WINE	UCI	3	178	13
LEUKEMIA	TG99	2	38	7129
LIVER	UCI	2	345	6
SVMGUIDE2	CWH03A	3	391	20
VEHICLE	STATLOG	4	846	18
VOWEL	UCI	11	528	10
BREAST	UCI	2	683	10
FOURCLASS	TKH96A	2	862	2
GLASS	UCI	6	214	9
HEART	STATLOG	2	270	13
IONOSPHERE	UCI	2	351	34
SONAR	UCI	2	208	60

In [14] the authors independently developed an approximated version based on a "crude" distance metric used to compute the neighborhood of the testing point which demonstrated to drastically speed up the method. However, our intention here is to assess the classification capabilities of the original formulation for which some theoretical properties are valid. In fact, considering kNNSVM as a local SVM classifier built in the features space, 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 R^2/Δ^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.

Theorem 2 (Blanzieri & Melgani, in press)**.** *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 and alternative and rigourous proof see [3]. □

As a consequence of Theorem 2, kNNSVM has the potential of improving over both 1NN and SVM for some $2 < k < N$.

4 Empirical testing of kNNSVM

We tested the performances of the kNNSVM classifier in comparison with the performances of SVM on the 13 datasets listed in Table 1. They are datasets extensively used in the machine learning

Table 2: 10 fold cross validation accuracies for SVM and kNNSVM with the LIN kernel.

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.967	0.960	−0.007	
WINE	0.966	0.983	+0.017	
LEUKEMIA	**0.950**	0.925	−0.025	
LIVER	0.681	**0.739**	+0.058	√
SVMGUIDE2	0.816	**0.859**	+0.043	√
VEHICLE	0.799	**0.861**	+0.061	√
VOWEL	0.837	0.998	+0.161	√
BREAST	0.968	0.966	−0.001	
FOURCLASS	0.768	**1.000**	+0.232	√
GLASS	0.622	0.692	+0.071	√
HEART	0.826	0.822	−0.004	
IONOSPHERE	0.869	0.929	+0.060	√
SONAR	0.779	0.875	+0.096	√

community taken from the website of LibSVM [5] and belonging to different research fields and application domains. Seven datasets are for binary classification, while the others are multiclass with a number of classes ranging from 3 to 11. The cardinality of the training set is always under 1000 and the number of features varies from 2 to 7129. We do not test the performance of kNN with respect to kNNSVM because it has already been shown, for instance in [4], that SVMs perform generally better than kNN. Moreover it is also accepted that, for a fixed value of k, SVM performs better than the majority rule and thus, if the model selection is done correctly, kNNSVM performs better than kNN.

We evaluate the performances of the classifiers using the 10-fold cross validation (CV) classification accuracies considering the linear kernel (LIN), the radial basis function kernel (RBF), the homogeneous polynomial kernel (HPOL) and the inhomogeneous polynomial kernel (IPOL). The folds were randomly chosen during preprocessing. The model selection (performed on each fold) was performed with stratified 10-fold CV splitting randomly the data at each application. The C parameter of SVM is chosen in the set $\{1, 5, 10, 25, 50, 75, 100, 150, 300, 500\}$, the σ parameter of the RBF kernel among $\{2^{-10}, 2^{-9}, \ldots, 2^{9}, 2^{10}\}$ and the degree of the polynomial kernels is bounded to 5. The dimension of the neighborhood for the kNNSVM classifier, i.e. k, is chosen among the first 5 odd natural numbers followed by the ones obtained with a base-2 exponential increment from 9

Table 3: 10-fold cross validation accuracies for SVM and kNNSVM with the RBF kernel.

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.947	0.960	+0.013	
WINE	**0.994**	0.989	−0.006	
LEUKEMIA	0.708	0.875	+0.167	√
LIVER	0.722	0.728	+0.006	
SVMGUIDE2	0.836	0.844	+0.008	
VEHICLE	0.849	0.840	−0.008	
VOWEL	0.992	**1.000**	+0.008	
BREAST	0.968	**0.971**	+0.003	
FOURCLASS	0.999	1.000	+0.001	
GLASS	0.687	0.674	−0.013	
HEART	**0.830**	0.819	−0.011	
IONOSPHERE	**0.937**	0.935	−0.003	
SONAR	0.894	**0.904**	+0.010	

Table 4: 10-fold cross validation accuracies for SVM and kNNSVM with the HPOL kernel.

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	**0.973**	0.960	−0.013	
WINE	0.966	0.989	+0.023	√
LEUKEMIA	0.950	0.925	−0.025	
LIVER	0.713	0.739	+0.026	√
SVMGUIDE2	0.816	0.841	+0.026	
VEHICLE	0.837	0.857	+0.020	√
VOWEL	0.979	0.998	+0.019	√
BREAST	0.968	0.965	−0.003	
FOURCLASS	0.811	1.000	+0.189	√
GLASS	0.720	**0.720**	+0.001	
HEART	0.822	0.822	0.000	
IONOSPHERE	0.892	0.929	+0.037	√
SONAR	0.880	0.890	+0.010	

and the cardinality of the training set, namely in {1, 3, 5, 7, 9, 11, 15, 23, 39, 71, 135, 263, 519, $|training\_set|$}. To assess the statistical significance of the differences between the 10-fold CV of SVM and kNNSVM we use the two-tailed paired t-test ($\alpha = 0.05$) on the two sets of fold accuracies. We used LibSVM [5] for SVM (adopting the one-against-all strategy for multiclass classification problems) and as the base for our implementation of kNNSVM.

The 10-fold CV classification results for the four kernels are reported in Tables 2, 3, 4 and 5. The best achieved accuracy results for each dataset are in bold (considering also the non reported decimal values). In case of multiple best results the simpler method is considered (with SVM simpler than kNNSVM and LIN kernel simpler than RBF, HPOL and IPOL kernels).

We can notice that kNNSVM performs substantially better than SVM in a considerable number of datasets without cases of significant losses in accuracies. Considering all the kernels, kNNSVM improves the SVM performances in 34 cases (65%) and the improvements are significant in 19 cases (37%) while for the 15 cases in which it reduces the accuracies of SVM the differences are never significant. Overall kNNSVM produces 8 times the best result against the 5 of SVM. In particular for kNNSVM with the LIN kernel we have 9 datasets in which kNNSVM achieve better 10-fold CV accuracies (8 significant), and 8 for the polynomial kernels (6 significant for the HPOL kernel and 4 for the IPOL kernel). In the case of RBF kernel we have 8 improvements but only one is significant;

Table 5: 10 fold cross validation accuracies for SVM and kNNSVM with the IPOL kernel.

DATASET	SVM	kNNSVM	DIFF	$p < 0.05$
IRIS	0.973	0.967	−0.007	
WINE	0.966	0.994	+0.028	√
LEUKEMIA	0.950	0.925	−0.025	
LIVER	0.701	0.733	+0.032	√
SVMGUIDE2	0.826	0.857	+0.031	√
VEHICLE	0.847	0.848	+0.001	
VOWEL	0.989	0.998	+0.009	√
BREAST	0.968	0.962	−0.006	
FOURCLASS	0.998	1.000	+0.002	
GLASS	0.701	0.706	+0.006	
HEART	0.822	0.822	0.000	
IONOSPHERE	0.912	0.929	+0.017	
SONAR	0.875	0.890	+0.015	

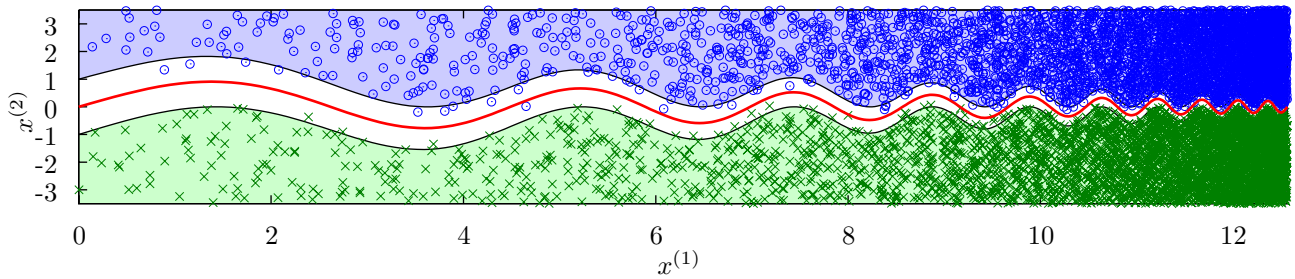

Figure 1: The DECSIN artificial dataset. The black lines denote the limit of the points of the two classes without noise, the red line denotes the optimal separation of the two classes.

this is due both to the fact that in two cases we reach the perfect classification without the possibility to improve significantly the SVM results and to the fact that the SVM with RBF kernel has already a high classification accuracy. We further discuss the comparison between SVM and kNNSVM with RBF kernel in the next section.

Considering the presented data about classification performances of kNNSVM, we can conclude that the application of kNNSVM is able to systematically and significantly improve the classification accuracy of SVM at least for the linear and polynomial kernels.

5 kNNSVM on artificial datasets

In order to show that there are situations in which kNNSVM has chance to improve on SVM with RBF kernel we built two artificial datasets.

The DECSIN artificial dataset. The first dataset is shown in Figure 1. It is a two feature dataset built starting from the following parametric function:

$$\begin{cases} u(t) = \dfrac{t}{1 + c \cdot t} \\ v(t) = \dfrac{sin(t)}{1 + c \cdot t} \end{cases} \qquad c = \frac{1}{5 \cdot \pi}, \quad t \in [0, 20\pi]$$

considering $y_i = +1$ if $x_i^{(1)} = u(t)$ and $x_i^{(2)} > v(t)$, and $y_i = -1$ if $x_i^{(1)} = u(t)$ and $x_i^{(2)} < v(t)$ where $x_i^{(j)}$ denotes the j-th component of the vector $x_i = (u(t), v(t))$. The reticulum of points is defined with a minimum distance of $\frac{1}{1+c \cdot t}$ from $v(t)$, increases the resolution as $\frac{1}{1+c \cdot t}$ on both axes and the samples are modified by a gaussian noise with zero mean and variance of $\frac{0.25}{1+c \cdot t}$.

We applied on this artificial dataset the SVM and kNNSVM with the RBF kernel as shown in Figure 2. We can notice that SVM with RBF has serious problem of under- and over-fitting depending on the value of the σ parameter (with C fixed to 1). In fact, if the σ parameter is too high ($\sigma = 1$ in the upper image of Figure 2) the separating hyperplane is close to the optimal separation in the leftmost region of the dataset, but it reduces to a straight line in the rightmost region clearly underfitting the data. Conversely, if the width parameter is too low ($\sigma = 1/50$ in the second image of Figure 2) there are problems of overfitting in the leftmost region. An intermediate value of the width parameter ($\sigma = 1/10$ in the third image) reaches an unsatisfactory compromise because, even if the central region of the dataset is correctly separated, there are both problems of underfitting (in the leftmost region) and underfitting (in the rightmost region). Acting on the C parameter of SVM is not resolutive because in all the three cases the number of misclassified points is very low.

We applied to the same dataset the kNNSVM method with RBF kernel. In order to avoid the validation of σ for every local application of SVM, we chose to automatically estimating it with the 0.1 percentile of the distribution of the distances between every pair of samples in each local training

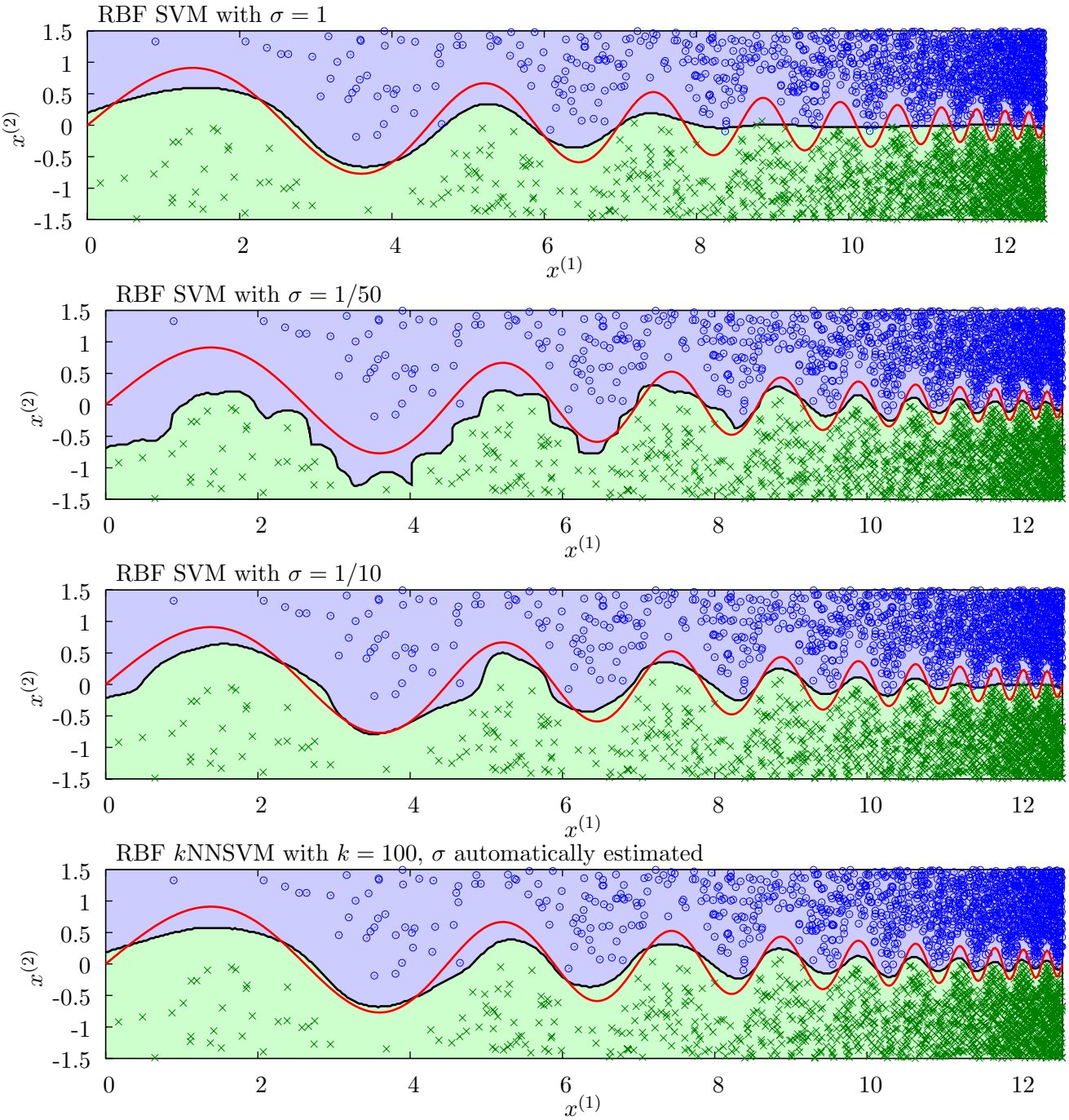

Figure 2: The behaviour of SVM and kNNSVM with RBF kernel on the DECSIN dataset (reported here on the $[-1.5, 1.5]$ interval on the y axis). We can notice that SVM has problems of under- or over-fitting depending on the σ parameter, whereas kNNSVM has a decision function close to the optimal separation in every region of the dataset. kNNSVM estimates σ as the 0.1 percentile of the distribution of the distance between every pair of points; this estimation for SVM gives $\sigma = 0.6$ which produces a separation very similar to the one with $\sigma = 1$ thus underfitting the data in the leftmost region of the dataset (this underfitting is still present with $\sigma = 1/10$ as shown in the third image).

set. Setting C=1 and k=100, we can notice (last image of Figure 2) that the separation produced by kNNSVM is close to the optimal separation in every region of the dataset without the over- or under-fitting problems seen for SVM.

The 2SPIRAL artificial dataset The second artificial dataset is based on the two spiral problem. The two classes are defined with the following function:

$$\begin{cases} x^{(1)}(t) = c \cdot t^d \cdot sin(t) \\ x^{(2)}(t) = c \cdot t^d \cdot cos(t) \end{cases} \qquad d = 2.5, \quad t \in [0, 10\pi]$$

using $c = 1/500$ for the first class ($y_i = +1$) and $c = -1/500$ for the second class ($y_i = -1$). The points are sampled with intervals of $\pi/30$ on the t parameter.

Although no noise is added to the data, also in this case SVM with RBF kernel exhibits problems of under- and over-fitting. In fact, if we choose a value of σ that separates well the data in the peripheral regions of the dataset it underfits the data in the central region (first image of Figure 3) and viceversa. In particular in order to classify perfectly the training set, SVM with RBF kernel and $C = 1$ needs to set $\sigma < 1/77750$ dramatically overfitting the data for peripheral regions (it is evident also with $\sigma = 1/10000$ in the second image of Figure 3), while kNNSVM is able to classify correctly all the training samples maintaining a good separation in all the dataset (last image of Figure 3).

So, even if the classification performances of kNNSVM with RBF kernel was not particularly positive for the benchmark datasets of the previous section, we showed here that, at least when the data has variable spatial resolution, it can have substantial advantages with respect to SVM with RBF kernel.

6 Conclusions

In this paper we empirically tested the classification performances of kNNSVM which can be seen as a SVM classifier built on the neighborhood in the feature space of the testing sample and for which there is the theoretical advantage of a lower radius/margin bound. We found that, in comparison with standard SVM, kNNSVM introduces a significant gain in the classification accuracy in a considerable number of datasets using the linear and polynomial (homogeneous and inhomogeneous) kernels. The strategy to find the k parameter proved to be effective enough to produce the effect guaranteed by the favourable bound. For the RBF kernel the improvements are less marked, but we presented two artificial datasets in which kNNSVM with RBF kernel behaves substantially better than SVM with the same kernel. So, even if the computational effort of its general formulation is considerable and thus some approximations of the method are desirable, kNNSVM has the possibility to sensibly improve the classification accuracies of a wide range of classification problems.

References

[1] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*, 2007. University of California, Irvine, School of Information and Computer Sciences. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[2] E. Blanzieri and F. Melgani. An adaptive SVM nearest neighbor classifier for remotely sensed imagery. *IEEE Int Conf on Geoscience and Remote Sensing Symposium (IGARSS-2006)*, pages 3931–3934, 2006.

[3] E. Blanzieri and F. Melgani. Nearest neighbor classification of remote sensing images with maximal margin principle. *IEEE Trans Geosci Rem Sens*, in press.

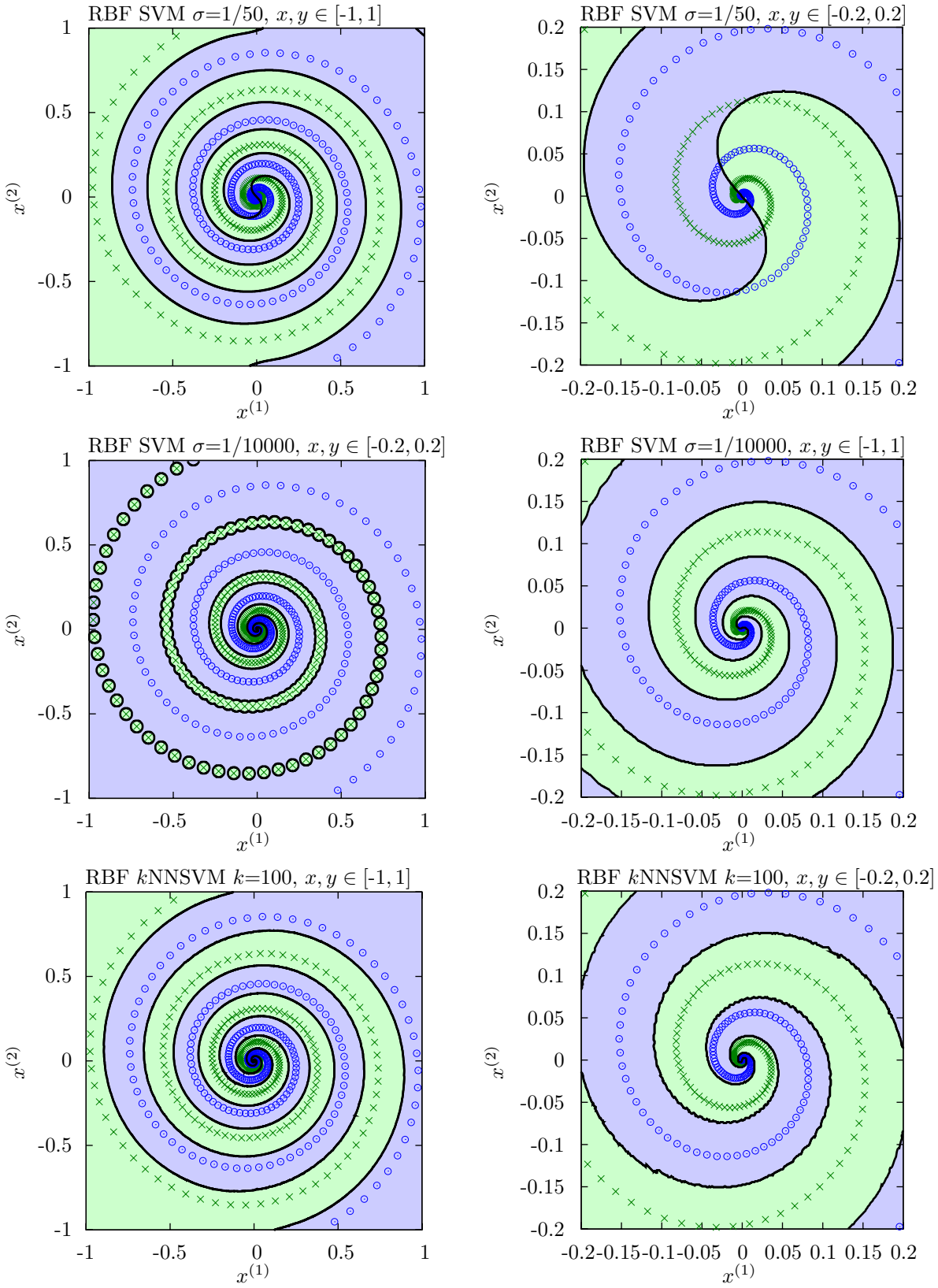

Figure 3: The SVM and kNNSVM with RBF kernel on the 2SPIRAL dataset. In the left columns, from the top, we have RBF SVM with $\sigma = 1/50$, RBF SVM with $\sigma = 1/10000$ and RBF kNNSVM with $k = 100$ and σ automatically set with the 0.1 percentile of the distribution of the distances between the samples. The automatic selection of σ for SVM gives $\sigma = 0.004$ which leads to a classification accuracy of the training set of 89% thus underfitting the data in the central region. The right column reports the same classifiers on the same dataset but reducing the resolution to the $[-0.2, 0.2]$ interval on both axes.

[4] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *Proc of the 23rd Int Conf on Machine learning*, pages 161–168, 2006.

[5] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] TR Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531, 1999.

[8] W. He and Z. Wang. Optimized local kernel machines for fast time series forecasting. *Third Int Conf on Natural Computation (ICNC 2007)*, 1, 2007.

[9] T.K. Ho and E.M. Kleinberg. Building projectable classifiers of arbitrary complexity. *Proc of the 13th Int Conf on Pattern Recognition (ICPR-96)*, 2:880, 1996.

[10] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Technical report, Dept. of Computer Science, Taiwan University, 2003.

[11] RD King, C. Feng, and A. Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Appl Artif Intell*, 9(3):289–333, 1995.

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

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

[14] H. Zhang, A.C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *Proc of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:2126–2136, 2006.