Empirical Assessment of Classification Accuracy of Local SVM

Nicola Segata
DISI, University of Trento, Italy
segata@disi.unitn.it

Enrico Blanzieri
DISI, University of Trento, Italy
blanzier@disi.unitn.it

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 novel promising classifier. Since no extensive validation was performed yet, we test here kNNSVM on 13 widely used datasets obtaining statistically significant better classification results with respect to SVM for linear and polynomial kernels. For RBF kernels the advantages seems not to be substantial, but we present two toy datasets in which kNNSVM performs much better than SVM with RBF kernel. The empirical results suggest to use kNNSVM for specific problems in which high classification accuracies are crucial and motivates further refinements of the approach.

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 by Blanzieri and Melgani (2008). 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. kNNSVM belongs to the class of *local learning algorithms* (Bottou and Vapnik, 1992) for which the locality parameter permits to find a lower minimum of the guaranteed risk as demonstrated in (Vapnik and Bottou, 1993; Vapnik, 2000). Zhang et al. (2006) 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. A method that includes locality in kernel machines has been presented also for regression (He and Wang, 2007).

Even if the kNNSVM has been successfully applied on two specific classification tasks (remote sensing by Blanzieri and Melgani (2006) and visual category recognition by Zhang et al. (2006)), 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 that the properties of local learning algorithms stated in (Bottou and Vapnik, 1992; Vapnik and Bottou, 1993) are effective in combination with a local maximal margin principle. Moreover, assessing the better classification accuracy of kNNSVM, would suggest to investigate more in depth the convergence between kNN and SVM and in particular some approximations of kNNSVM in order to make it scalable for large and very large datasets and applicable in online and/or active learning settings.

In this work, we empirically compare the classification performances of SVM and kNNSVM on 13 datasets taken from different application domains and with 4 kernel functions and further analyse the kNNSVM and SVM behaviours in combination with the RBF kernel. The paper is organized as follows. After preliminaries on kNN and SVM (Section 2) we describe the kNNSVM classifier (Section 3). Then we detail the comparison of kNNSVM and SVM on real datasets (Section 4) and for the RBF kernel by means of two toy datasets (Section 4). Finally, we draw some conclusions and discuss future works.

Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning, pages 47–55,
Tilburg, May 2009.

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 the function $r_{x'} : \{1, \ldots, N\} \to \{1, \ldots, N\}$ as:

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

In this way, $x_{r_{x'}(j)}$ is the point of the set X in the j-th position in terms of (Euclidean) 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 is defined by

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

Support vector machines. SVMs (Cortes and Vapnik, 1995) are classifiers with sound foundations in statistical learning theory (Vapnik, 2000). 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:

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

with $\xi_i \geq 0$, $i = 1, \ldots, N$ and where $y_i \in \{-1, +1\}$ is the class label of the i-th nearest training sample. The slack variables ξ_i's allow some misclassification on the training set and are set accordingly to the regularization parameter C. Reformulating such an optimization problem with Lagrange multipliers α_i $(i = 1, \ldots, N)$, and introducing a positive definite (PD) kernel 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 kernel trick avoids the explicit definition of the feature space $\mathcal{F}$ and of the mapping Φ (Schölkopf and Smola, 2002; Cristianini and Shawe-Taylor, 1999). Popular kernels are the linear kernel, the Gaussian radial basis function kernel (σ is the width), the homogeneous and inhomogeneous polynomial kernels (δ is the degree) defined as:

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

3 The kNNSVM classifier

The method (Blanzieri and Melgani, 2006; Blanzieri and Melgani, 2008) 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.

kNNSVM tackles the classification problem differently from traditional supervised learning and SVM in particular. In fact instead of estimating a global decision function with a low probability of errors on *all* possible unseen samples, kNNSVM tries to estimate a decision function with a low probability of error on labeling a *given* point. Notice that for kNN (the simplest local learning algorithm) this learning statement is crucial because the majority rule is effective only locally (globally it reduces to the class with the highest cardinality). With respect to global SVM, the possibility of estimating a different maximal margin hyperplane

[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.

for each test point can thus achieve a lower probability of misclassification on the whole test set. These considerations are formalized in the theory of local structural risk minimization for local learning algorithms (Vapnik and Bottou, 1993) which is a generalization of the structural risk minimization (Vapnik, 2000). The main idea is that, in addition to the complexity of the class of possible functions and of the function itself, the choice of the locality parameter (k for kNNSVM) can help to lower the guaranteed risk.

kNNSVM builds an SVM over the neighborhood of each test point x'. Accordingly, the constraints in (1) become:

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

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

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

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:

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

For the RBF kernel or any polynomial kernels with degree 1, the ordering function is equivalent to using the Euclidean metric. For some non-linear kernels (other than the RBF kernel) the ordering function can be different to that produced using the Euclidean metric.

The decision rule becomes: $k\text{NNSVM}(x) =$

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

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.

Table 1: The 13 datasets used in the experiments. Number of classes, training set cardinality and number of features are reported.

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 this work we use a simple C++ implementation of kNNSVM using LibSVM (Chang and Lin, 2001) for training the local SVM models, and a *brute-force* implementation of the kNN procedure used to retrieve the neighborhoods, since in this work the focus is not on computational performances. A fast exact kNNSVM implementation, called FkNNSVM, is available in FaLKM-lib by Segata (2009), a library for fast local kernel machines, freely available for research and education purposes at `http://disi.unitn.it/~segata/FaLKM-lib`.

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 community and belong to different research fields and application domains; they are retrieved from the website of LibSVM (Chang and Lin, 2001) and their original references are: UCI (Asuncion and Newman, 2007), TG99 (Golub and others, 1999), Statlog (King et al., 1995), CWH03a (Hsu et al., 2003), TKH96a (Ho and Kleinberg, 1996). 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 evaluate the performances using the 10-fold cross validation (CV) classification accuracies considering the linear kernel (LIN), the

Table 2: 10 fold CV 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	√

Table 4: 10-fold CV 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	

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 (on each fold) was performed with 10-fold CV splitting randomly the data at each application. The regularization parameter C of SVM is chosen in {1, 5, 10, 25, 50, 75, 100, 150, 300, 500}, σ 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 in $\{1, 3, 5, 7, 9, 11, 15, 23, 39, 71, 135, 263, 519, |training\_set|\}$. In case of multi-class datasets we adopt the one-against-all strategy (both for SVM and kNNSVM) that does not require the generalization of the binary class case formalized in Section 2. To assess the statistical significance of the differences between SVM and kNNSVM we use the two-tailed paired t-test ($\alpha = 0.05$) on the two sets of fold accuracies. For SVM we used the LibSVM (Chang and Lin, 2001).

The 10-fold CV accuracy 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. 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).

kNNSVM performs substantially better than SVM in a considerable number of datasets without cases of significant accuracy losses. Considering all 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

Table 3: 10-fold CV 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	0.925	+0.217	√
LIVER	0.722	0.728	+0.006	
SVMGUIDE2	0.836	0.844	+0.008	
VEHICLE	0.849	0.840	−0.008	
VOWEL	0.992	0.998	+0.006	
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 5: 10 fold CV 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	

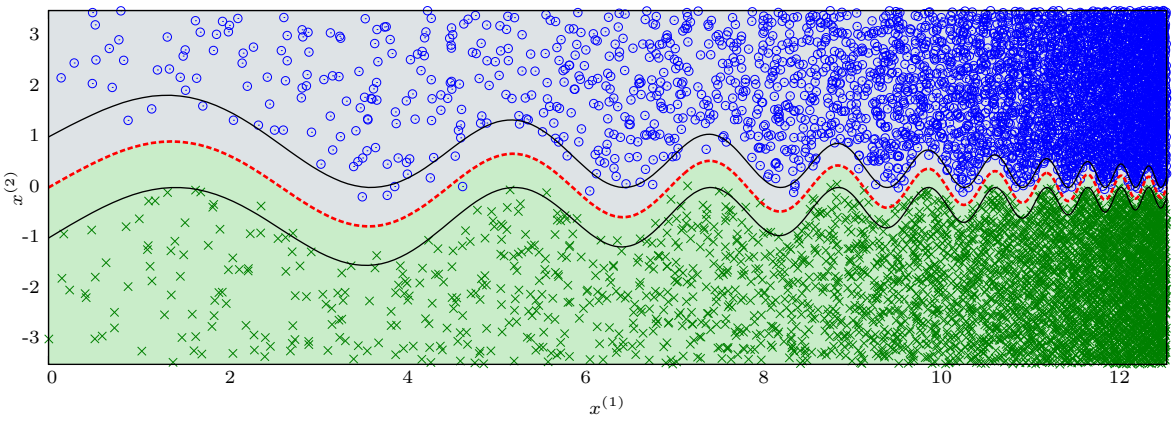

Figure 1: The DECSIN dataset. The black lines denote the limit of the points of the two classes without noise, the dotted line denotes the optimal separation of the two classes.

which it reduces the accuracies of SVM the differences are never significant. Considering all datasets at the best of the kernel choice, kNNSVM produces 8 times the best result against the 5 of SVM. For kNNSVM with the LIN kernel we have 9 datasets in which kNNSVM achieves 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; 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 SVM and kNNSVM with RBF kernel in the next section.

5 kNNSVM and the RBF kernel

In order to study the situations in which kNNSVM improves on SVM with RBF kernel we built two toy datasets. Our intention is to show that there are cases in which kNNSVM is able to build decision functions that are much closer to the optimal one with respect to SVM with RBF kernel. The following two toy datasets are thus used to graphically compare the decision function of the two methods without explicit classification accuracies. Moreover, we also highlight the ability of kNNSVM of setting locally the kernel parameters further increasing the local adaptation of the decision function. In particular, the local choice of σ is performed using the 0.1 percentile of the distribution of the distances between every pair of the k samples nearest to the testing point. The choice of k in this section is arbitrary since our purpose here is to show that kNNSVM on these two datasets has the potentialities of approaching the perfect decision function while no choices of SVM parameters permits a similar behaviour.

The 2SPIRAL dataset. The first toy dataset is based on the two spiral problem, a recurrent artificial benchmark problem in machine learning, see for example (Ridella et al., 1997; Suykens and Vandewalle, 1999):

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

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 every $t \cdot \pi/30$.

Figure 2 shows the application of SVM and kNNSVM with RBF kernel on the 2SPIRAL dataset. Although no noise is added to the data, RBF-SVM exhibits problems of under- and over-fitting whereas kNNSVM is able to find a separating function very close to the optimal one (details in the caption of Figure 2).

The DECSIN dataset. The second toy dataset (Figure 1) is a two feature dataset built starting from the models of Park et al. (2004) and Adibi and Safabakhsh (2007) and

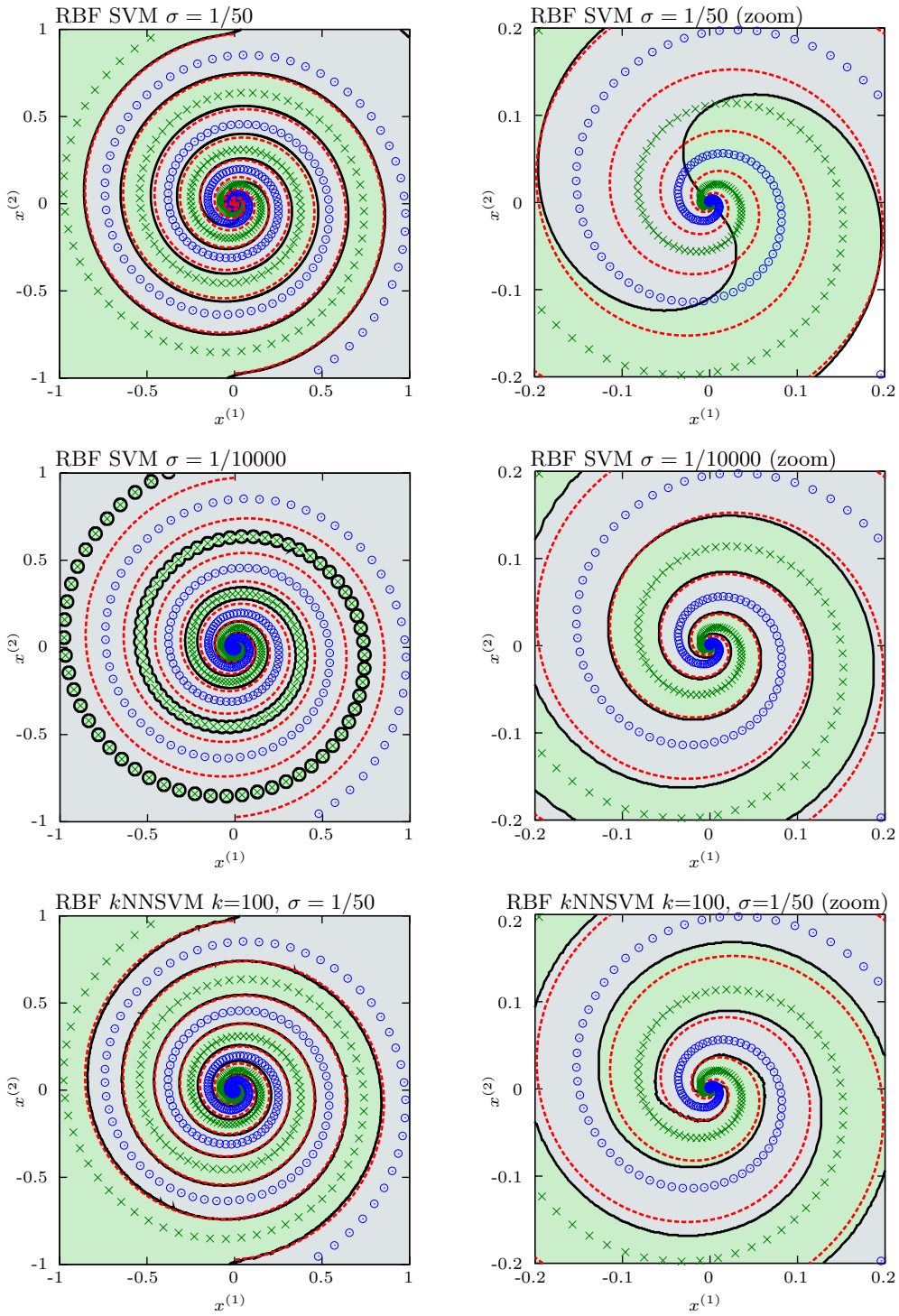

Figure 2: The decision function of SVM and kNNSVM with RBF kernel on the 2SPIRAL dataset (the dotted line denotes the optimal separation). In the first row we have RBF SVM with $\sigma = 1/50$, in the second RBF SVM with $\sigma = 1/10000$, in the third row RBF kNNSVM with $\sigma = 1/50$. The right columns report the same classifiers on the same dataset but reducing the resolution to the [-0.2, 0.2] interval on both axes. The underfitting problems of large σ values and overfitting problems of low σ values for SVM are evident. Intermediate values of σ are not resolutive because, even if it is not clear from the picture, also $\sigma = 1/10000$ gives underfitting problems in the central region in fact the perfect training set separation is achievable only with $\sigma < 1/77750$. On the contrary, kNNSVM (last row) does not show evident over or under-fitting problems with the same σ that causes underfitting of the central region with SVM.

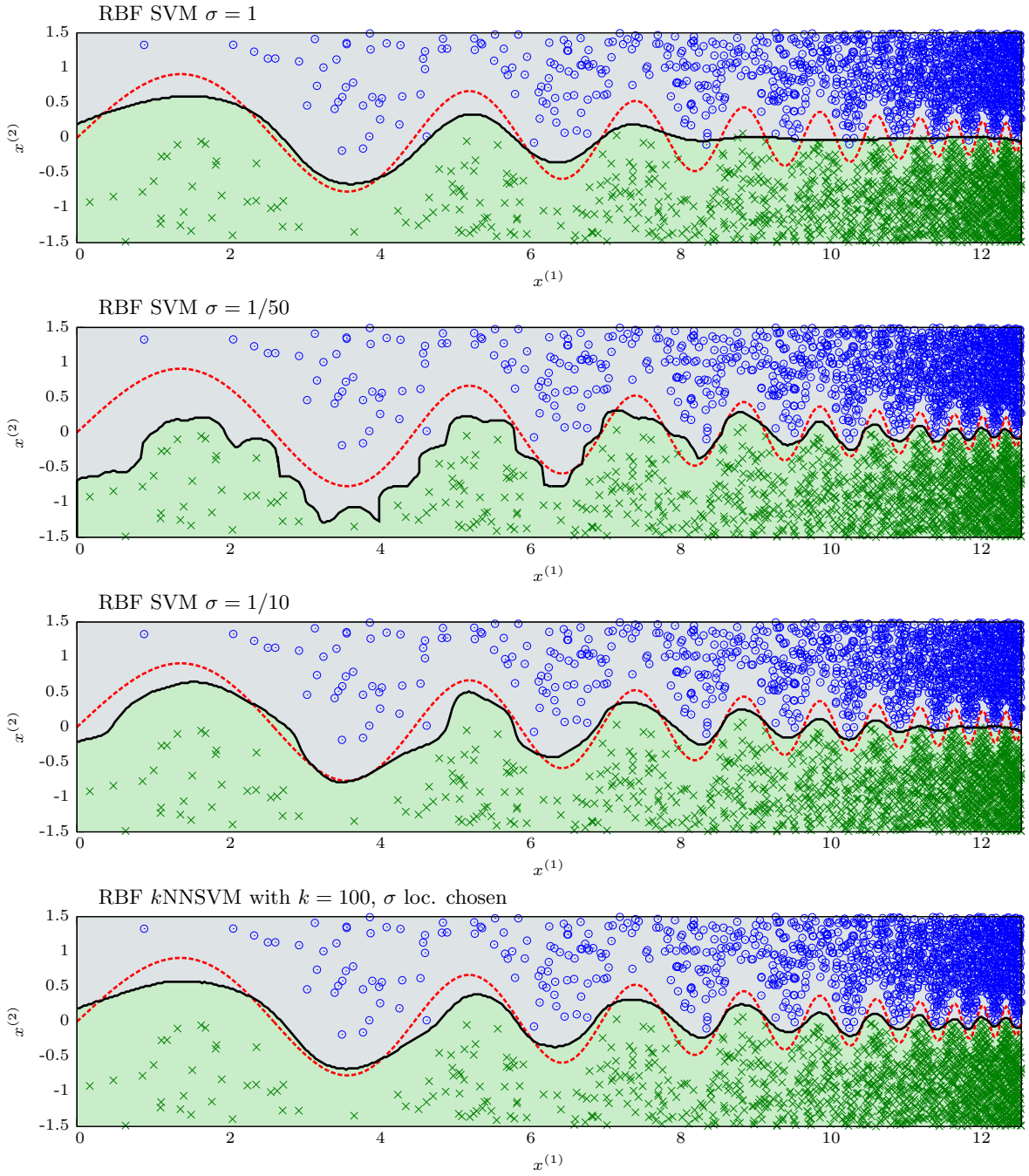

Figure 3: The behaviour of SVM and kNNSVM with RBF kernel on the DECSIN dataset (reported here on the $[-1.5, 1.5]$ interval on the y axis). We can notice that SVM has problems of under- or over-fitting depending on the σ parameter. In fact, if the σ parameter is too high ($\sigma = 1$, first row) the 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 ($\sigma = 1/50$, second row) there are problems of overfitting in the leftmost region. An intermediate value of the width parameter ($\sigma = 1/10$, third row) 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. kNNSVM with the local choice of σ and setting C=1 and k=100 (last row) has instead a decision function close to the optimal separation in every region of the dataset.

with 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 points are defined with a minimum distance of $\frac{1}{1+c \cdot t}$ from $v(t)$, increase the resolution as $\frac{1}{1+c \cdot t}$ on both axes and are modified by a Gaussian noise with zero mean and variance of $\frac{0.25}{1+c \cdot t}$.

The application of SVM and kNNSVM with the RBF kernel using the local choice of σ on the DECSIN dataset is shown in Figure 3. Also in this case, RBF-SVM has serious problem of under- and over-fitting depending on the value of σ while the separation produced by kNNSVM is close to the optimal separation in every region of the dataset without (details in the caption of Figure 3).

So, even if the classification performances of kNNSVM with RBF kernel was not particularly positive for the benchmark datasets of Section 4, we showed here that there are cased in which it can have substantial advantages with respect to SVM with RBF kernel.

6 Conclusions and future works

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. We found that, in comparison with 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 reach the theoretical advantage of a lower minimum of the guaranteed risk for local learning algorithms. For the RBF kernel the improvements are less marked, but we presented two toy datasets in which kNNSVM with RBF kernel behaves substantially better than SVM with the same kernel. So kNNSVM has the possibility to sensibly improve the classification accuracies of a wide range of classification problems.

Apart the suggestion to apply kNNSVM on specific problems in various domains, this empirical study motivates us to further develop the approach. Numerous research directions can in fact be identified. First, multiple local SVMs (for example varying k) can be used to predict the label of a test point, in a local ensemble fashion. Second, similarly to the kNN, automatic and robust approaches to set the locality parameter k are desirable. Third, specific data-structures to speed up the k-NN operations needed to retrieve the neighborhood of test points (like Cover Trees by Beygelzimer et al. (2006) that can be applied in general metric spaces thus also in Hilbert features spaces) can dramatically increase the computational performances of kNNSVM. Forth, it would be interesting to asses the level of influence of the curse of dimensionality on kNNSVM (it is well know that kNN, but not SVM, suffers in high-dimensional spaces). Fifth, in order to unburden the prediction phase, a set of local SVMs can be pre-computed in the training set (it is not necessary to build a local SVM for each training point because a single local SVM can be built for small clusters of close points) thus reducing the prediction step to select the nearest local SVM and use it to predict the label of the test point.

Very recently, some of these aspects have been implemented obtaining a fast exact version of kNNSVM, called FkNNSVM, the so-called FaLK-SVM classifier (Segata and Blanzieri, 2009) which is based on the last point discussed in the previous paragraph and it is much faster from a computational viewpoint of both kNNSVM and SVM (more than one order of magnitude faster than SVM on the very large CoverType dataset) maintaining a generalization ability very similar to kNNSVM, and two algorithms for performing noise reduction for data cleansing and competence enhancing of case-based reasoning approaches called LSVM noise reduction (Segata et al., 2008) and FaLKNR (Segata et al., 2009). The source code of all these implementations can be found in FaLKM-lib by Segata (2009), a library for fast local kernel machines, freely available for research purposes at `http://disi.unitn.it/~segata/FaLKM-lib`.

References

P. Adibi and R. Safabakhsh. 2007. Joint Entropy Maximization in the Kernel-Based Linear Manifold Topographic Map. In *Int Joint Conf. on Neural Networks, 2007. IJCNN 2007.*, pages 1133–1138.

A. Asuncion and D.J. Newman, 2007. *UCI Machine Learning Repository*. University of California, Irvine.

A. Beygelzimer, S. Kakade, and J. Langford. 2006. Cover Trees for Nearest Neighbor. In *ICML-06*, pages 97–104. ACM Press New York, NY, USA.

E. Blanzieri and F. Melgani. 2006. An adaptive SVM nearest neighbor classifier for remotely sensed imagery. In *IGARSS 2006*.

E. Blanzieri and F. Melgani. 2008. Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Trans Geosci Remote Sens*, 46(6).

L. Bottou and V. Vapnik. 1992. Local learning algorithms. *Neural Comput*, 4(6).

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

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

N. Cristianini and J. Shawe-Taylor. 1999. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press New York, NY, USA.

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

W. He and Z. Wang. 2007. Optimized local kernel machines for fast time series forecasting. *Proc. of ICNC-2007*.

T.K. Ho and E.M. Kleinberg. 1996. Building projectable classifiers of arbitrary complexity. *Proc of ICPR-96*, 2:880.

C.W. Hsu, C.C. Chang, and C.J. Lin. 2003. A practical guide to support vector classification. Technical report, Taiwan University.

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

J.H. Park, K.H. Im, C.K. Shin, and S.C. Park. 2004. MBNR: Case-Based Reasoning with Local Feature Weighting by Neural Network. *Applied Intelligence*, 21(3):265–276.

S. Ridella, S. Rovetta, and R. Zunino. 1997. Circular backpropagation networks for classification. *Neural Networks, IEEE Transactions on*, 8(1):84–97.

B. Schölkopf and A.J. Smola. 2002. *Learning with kernels*. MIT Press.

N. Segata and E. Blanzieri. 2009. Fast local support vector machines for large datasets. In *Int Conf on Machine Learning and Data Mining MLDM 2009*. Accepted for Pubblication.

N. Segata, E. Blanzieri, S.J. Delany, and P. Cunningham. 2008. Noise reduction for instance-based learning with a local maximal margin approach. Technical Report DISI-08-056, DISI, University of Trento, Italy.

N. Segata, E. Blanzieri, and P. Cunningham. 2009. A scalable noise reduction technique for large case-based systems. In *Int Conf on Case-Based Reasoning (ICCBR 09)*. Accepted for Pubblication.

Nicola Segata, 2009. *FaLKM-lib: a Library for Fast Local Kernel Machines*. Freely available for research and education purposes at `http://disi.unitn.it/~segata/FaLKM-lib`.

JAK Suykens and J. Vandewalle. 1999. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9(3):293–300.

V. N. Vapnik and L. Bottou. 1993. Local algorithms for pattern recognition and dependencies estimation. *Neural Comput*, 5(6).

V.N. Vapnik. 2000. *The Nature of Statistical Learning Theory*. Springer.

H. Zhang, A.C. Berg, M. Maire, and J. Malik. 2006. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *Proc of the CVPR-06*, 2:2126–2136.