Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 63–71
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A comparative analysis of structural risk minimization by support vector machines and nearest neighbor rule ☆

Bilge Karaçalı [a], Rajeev Ramanath [b,*], Wesley E. Snyder [b]

[a] Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, USA
[b] Department of Electrical and Computer Engineering, Box 7911, North Carolina State University, Raleigh, NC 27695-7911, USA

Received 23 October 2002; received in revised form 25 July 2003

Abstract

Support vector machines (SVMs) are by far the most sophisticated and powerful classifiers available today. However, this robustness and novelty in approach come at a large computational cost. On the other hand, nearest neighbor (NN) classifiers provide a simple yet robust approach that is guaranteed to converge to a result. In this paper, we present a technique that combines these two classifiers by adopting a NN rule-based structural risk minimization classifier. Using synthetic and real data, the classification technique is shown to be more robust to kernel conditions with a significantly lower computational cost than conventional SVMs. Consequently, the proposed method provides a powerful alternative to SVMs in applications where computation time and accuracy are of prime importance. Experimental results indicate that the NNSRM formulation is not only computationally less expensive, but also much more robust to varying data representations than SVMs.

Keywords: Nearest neighbor; Structural risk minimization; Support vector machines; Kernel operator; Prototype selection

1. Introduction

Support vector machines (SVMs) (Cortes and Vapnik, 1995; Vapnik et al., 1997; Osuna et al., 1997; Schölkopf et al., 1999) provide a novel means of classification using the principles of structural risk minimization (SRM) (Vapnik, 1998). It is one of the most sophisticated non-parametric supervised classifiers available today, with many different configurations depending upon the function (kernel) used for generating the transform space in which the decision surface is constructed.

The main objective of the SRM principle is to control the generalization ability of the decision function by limiting the flexibility of the set of candidate functions, measured by the VC dimension (Vapnik, 1998). Instead of constructing a decision function f by minimizing the training error on a representative data set, f is chosen in a way to minimize an upper bound on the test error given by

☆ RR and WES were supported by US Army Space and Missile Defense Command grant number DASG60-02-1-0005.
* Corresponding author.
E-mail address: rajeev.ramanath@ieee.org (R. Ramanath).

0167-8655/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2003.09.002

$$R(f) \leqslant R^{\text{emp}}(f) + \varepsilon(f, \mathsf{h}, v) \tag{1}$$

where $v \in [0,1]$, $R^{\text{emp}}(f)$ is the training error of f (or *empirical risk*), $\varepsilon(f, \mathsf{h}, v)$ is the confidence term, and h is the VC dimension of $\mathscr{F}$, the set of candidate decision functions. The seminal work by Vapnik (Vapnik, 1998) shows that maximizing the margin of a planar decision surface between two classes achieves the minimum of this upper bound. This strategy is then employed in a transform space, $\mathscr{Z}$, using the images of the training data points under a non-linear map $z : \mathscr{X} \to \mathscr{Z}$, where $\mathscr{X}$ is the space in which the original data resides. The new construction is formulated through a positive definite symmetric function K referred to as the *kernel* characterizing the inner product between the images of data points through

$$\langle z(\boldsymbol{x}_i), z(\boldsymbol{x}_j) \rangle \equiv K(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2}$$

where $\boldsymbol{x}_i \in \mathscr{X}$ and $y_i \in \{-1, 1\}$ for $i = 1, \ldots, \ell$ denote the training data. The classifier takes on the form

$$f(\boldsymbol{x}) = \operatorname{sgn}\left\{ \sum_{i=1}^{\ell} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right\} \tag{3}$$

where α_i's are obtained by maximizing

$$\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4}$$

in the positive quadrant subject to $\sum_i \alpha_i y_i = 0$. The planar decision boundary in the transform space then corresponds to a non-linear classifier in the original space. Given an application, the choice of an appropriate transform space, however, is far from obvious. The radial basis function (RBF) kernel given by

$$K^{\text{rbf}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{e}^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}}^2}{2\tau^2}} \tag{5}$$

where τ is a scale parameter has gained some popularity among researchers due to its controlled non-linearity through τ and its remarkable performance in various applications. The distance in $\mathscr{X}$ computed using the vector space norm denoted by $\|\cdot\|_{\mathscr{X}}$ can be of various forms including L_1 and L_2 distances. Even when a particular kernel is chosen, multiple experiments still need to be performed, usually as a line search, in order to determine the "best" scale parameter. This, in turn, is highly time consuming.

In a recent publication, a novel technique is proposed (Karaçalı and Krim, 2003) to implement the SRM principle using a nearest neighbor (NN) rule, referred to as NNSRM, as an alternative to computationally expensive conventional SVMs. Analysis has shown that this new method had a lower order of complexity as opposed to the quadratic optimization of Eq. (4) required to train an SVM classifier, without the ordeal of choosing a suitable transform space and with nearly comparable performance.

In this paper, we establish the connection of SVM classification with the NN rule, and combine the frameworks of NNSRM and kernel operators to construct a novel family of classifiers. We then use experiments on synthetic and real data to compare the performances of the generalized NNSRM and SVM classifiers in terms of computational cost (measured by the training time), and classification accuracy. The results obtained on varying kernel parameters indicate that the NNSRM formulation is more cost-effective and robust to changing data representations than SVMs. Furthermore, the structural analysis of the two classification strategies following non-linear transformations of the original data provides additional insight on the notion of separability of classes.

This paper is organized as follows. Section 2 is devoted to the analysis of SVM classification from a NN rule perspective. In Section 3, we briefly summarize the NNSRM method, and extend it further to develop SVM-like classifiers using kernel operators. In Section 4, we provide a comparative performance analysis between the newly constructed classification technique and conventional SVMs for simulated and real data.

2. Support vector machine classification and nearest neighbor rule

The NN rule relies on feature similarities among points of the same class in making a categorical decision. The similarity is measured as a quantity inversely related to the distance between

points, and an uncategorized point is assigned to the class with most points in its neighborhood. The 1-NN rule effectively partitions the observation space into a union of Voronoi cells, describing non-overlapping regions of respective classes, and the classification decision is made according to which class' cell the uncategorized point falls.

The SVM strategy, in essence, quantifies the similarity between points through their inner product, which also reflects on the classifier functions in Eq. (3), and therefore bares a close relationship to the NN rule. In case of RBF kernels, this relationship is more apparent. Consider a trained SVM classifier using a RBF kernel with a certain scale parameter τ. Note that

$$\begin{aligned}\|z(\boldsymbol{x}_i)-z(\boldsymbol{x}_j)\|_{\mathscr{Z}}^2 &= \langle z(\boldsymbol{x}_i)-z(\boldsymbol{x}_j), z(\boldsymbol{x}_i)-z(\boldsymbol{x}_j)\rangle \\ &= \langle z(\boldsymbol{x}_i), z(\boldsymbol{x}_i)\rangle + \langle z(\boldsymbol{x}_j), z(\boldsymbol{x}_j)\rangle \\ &\quad - 2\langle z(\boldsymbol{x}_i), z(\boldsymbol{x}_j)\rangle \\ &= K(\boldsymbol{x}_i,\boldsymbol{x}_i) + K(\boldsymbol{x}_j,\boldsymbol{x}_j) \\ &\quad - 2K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ &= 2 - 2K(\boldsymbol{x}_i,\boldsymbol{x}_j), \end{aligned} \tag{6}$$

or equivalently,

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = 1 - \frac{1}{2}\|z(\boldsymbol{x}_i)-z(\boldsymbol{x}_j)\|_{\mathscr{Z}}^2. \tag{7}$$

Using Eq. (7) in Eq. (3), we obtain

$$\begin{aligned} f(\boldsymbol{x}) &= \operatorname{sgn}\left\{\sum_{i=1}^{\ell}\alpha_i y_i\left(1-\frac{1}{2}\|z(\boldsymbol{x})-z(\boldsymbol{x}_i)\|_{\mathscr{Z}}^2\right)+b\right\} \\ &= \operatorname{sgn}\left\{\sum_{i=1}^{\ell}\alpha_i y_i - \frac{1}{2}\sum_{i=1}^{\ell}\alpha_i y_i\|z(\boldsymbol{x})-z(\boldsymbol{x}_i)\|_{\mathscr{Z}}^2 + b\right\} \\ &= \operatorname{sgn}\left\{-\sum_{i=1}^{\ell}\alpha_i y_i\|z(\boldsymbol{x})-z(\boldsymbol{x}_i)\|_{\mathscr{Z}}^2 + 2b\right\}. \end{aligned} \tag{8}$$

According to Eq. (8), the initial SVM classifier essentially makes a categorical decision for a given point $\boldsymbol{x}$ based on the sign of the sum of its weighted distances to the support vectors biased by $2b$. A marginal data point belonging to class 1 residing in the region of support of class 2 is typically assigned a larger weight, which results in reduction of its region of support (in class 2 territory). Geometrically, this induces a shift on the decision boundary towards the point with the larger weight, also explaining the reason why the Lagrange multipliers in the SVM solution corresponding to the support vectors that reside nearest to the boundary turn out to be large. In this sense, in RBF space, the SVM classifier is one of many possible classifiers that can be constructed by assigning non-zero weights to the squared distances from data points in the region where the two classes meet.

3. NNSRM classification

The NN classifier makes a categorical decision for a given data point based on its distance to individual data points in its reference set, which usually is the whole training set. In its simplest form (1-NN), the NN classifier can be formulated as

$$f_S(\boldsymbol{x}) = y_{i'}$$

where

$$\|\boldsymbol{x}_{i'}-\boldsymbol{x}\|_{\mathscr{X}} = \min_{\boldsymbol{x}_i\in S}\|\boldsymbol{x}_i-\boldsymbol{x}\|_{\mathscr{X}} \tag{9}$$

where S is the reference set of sample data points from classes 1 and 2. The attractive aspects of NN classification are its universality, its versatility, and its rapid convergence properties to statistical optimum with increasing training samples populating the reference set (Cover and Hart, 1967; Devijver and Kittler, 1982).

The primary problem with the classical NN rule is the computational and data storage load. Given an uncategorized data point $\boldsymbol{x}$, computation of Eq. (9) entails calculating the distance from $\boldsymbol{x}$ to every other data point in the training set. Alleviating this load by means of reducing the number of data points in the reference set has been a major focus of attention in the pattern recognition literature of the past three decades, proliferating numerous methods and strategies on exact (Bhattacharya and Kaller, 1998) and inexact (but consistent) (Hart, 1968; Gates, 1972; Ritter et al., 1975; Tomek, 1976; Chidananda-Gowda and Krishna, 1979; Dasarathy, 1994; Kuncheva and Bezdek, 1998; Cerveron and Ferri, 2001) reference set thinning algorithms. A survey of these methods can be found in (Wilson and Martinez, 2000).

In spite of such an abundance, without the connection between the number of reference points and the generalization ability of the resulting classifier, these methods still remain improvised: they represent intuitive alternatives of reducing the computational cost of the NN classification rule without aiming at a particular theoretical criterion. The idea behind NNSRM classification is to combine the versatility of the NN rule with the generalizability of the SRM principle. The approach proposed in (Karaçalı and Krim, 2003) is to minimize the upper bound on the test error in Eq. (1) directly by minimizing the confidence term while fixing R^{emp} at zero. It is shown (Karaçalı and Krim, 2003) that the V–C dimension of a NN classifier is equal to the number of reference points in the training set. The monotonic relationship of the confidence term to the V–C dimension suggests seeking a consistent reference set with minimal number of data points in order to achieve a classification rule with controlled generalizability. A sub-optimal solution to this problem is provided by the following algorithm, making an analogy to the manner in which SVM decision boundary is created.

- setup phase:
  - compute all pairwise distances $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}}$ for all $\boldsymbol{x}_i$ in class 1 and $\boldsymbol{x}_j$ in class 2
  - sort these distances in ascending order, $d^{(0)}, d^{(1)}, \ldots$
- training phase:
  - initialize $k = 1$, and $S = \{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ where $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}} = d^{(0)}$
  - while $R^{\text{emp}}(f_S) > 0$, do
    find $\boldsymbol{x}_i$ in class 1 and $\boldsymbol{x}_j$ in class 2 such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}} = d^{(k)}$
    update $S \leftarrow S \bigcup \{\boldsymbol{x}_i, \boldsymbol{x}_j\}$
    increment k

The algorithm is guaranteed to converge (Karaçalı and Krim, 2003) since, in the worst case, the reference set will consist of all the training samples, hence the classifier will have zero training error. Furthermore, since most classification errors are made in the region where the two classes "face" each other, points closest to the opposite class samples are going to be introduced into the reference set S prior to the remaining points. Ultimately, the reference set will include only the points characterizing the region of the space where two classes are at close proximity, sufficient to achieve zero classification error on the complete training set. Since the computational complexity of this algorithm is small, it provides a low cost, robust alternative to conventional SVMs.

In cases where the two classes overlap, the final decision boundary will partition the region of overlap typically into Voronoi cells according to the membership of respective points. Outliers will therefore create Voronoi holes in the opposite class' territory, so as to satisfy the zero training error criterion, unless they reside far from the boundary and do not affect the internal classification. This behavior, on the other hand, is consistent with the NN strategy and the particular approach to minimize the upper bound on the test risk in Eq. (1) undertaken by the algorithm.

In this paper, we extend the approach described above by formulating the NNSRM method in a transform space using kernel operators instead of the initial observation space. A major component of SVM classification is the construction of a linear decision boundary in the transform space where the separability of the two classes is arguably improved. This suggests that constructing the NNSRM classifier in a similar transform space would also improve the performance. A similar example of using NN classification in a non-Euclidean space is proposed in (Farach-Colton and Indyk, 1999) using the Hausdorff metric, and in (Yu et al., 2002) using the kernels generating Hilbert spaces. With the availability of kernels, the development in a transform space $\mathscr{Z}$ is done using the same algorithm above with the distance in $\mathscr{X}$ replaced by the distance in $\mathscr{Z}$ which can be computed using

$$\begin{aligned}\|z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j)\|^2_{\mathscr{Z}} &= \langle z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j), z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j)\rangle_{\mathscr{Z}} \\ &= K(\boldsymbol{x}_i, \boldsymbol{x}_i) + K(\boldsymbol{x}_j, \boldsymbol{x}_j) - 2K(\boldsymbol{x}_i, \boldsymbol{x}_j)\end{aligned} \tag{10}$$

where K is the kernel characterizing the inner product in $\mathscr{Z}$ (Schölkopf, 2000; Yu et al., 2002).

An interesting observation is that the NNSRM algorithm applied using the RBF expansion will

return the same set of reference points (and thus the classifier in the transform space $\mathscr{Z}$) as it does in the initial observation space $\mathscr{X}$. Note that in RBF space, $\|z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j)\|_{\mathscr{Z}}$ is given by

$$\|z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j)\|_{\mathscr{Z}}^2 = 2 - 2\mathrm{e}^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}}^2}{2\tau^2}} \tag{11}$$

This shows the monotonic increasing property of $\|z(\boldsymbol{x}_i) - z(\boldsymbol{x}_j)\|_{\mathscr{Z}}$ as a function of $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathscr{X}}$ for a RBF kernel. The implication of this observation is even more interesting. The rationale behind employing non-linear transformations on the data is to achieve better separation of the two classes. The fact that the NNSRM returns the same classification rule for all RBF transform spaces as it does in the observation space suggests that the NNSRM method achieves the separability of an arbitrary RBF space in the observation space.

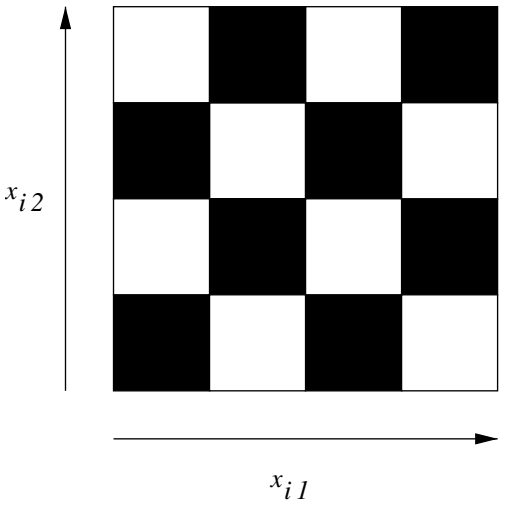

Fig. 1. Chessboard pattern used to generate synthetic data. Black squares correspond to class label −1 and white squares correspond to label +1.

4. Experiments and results

We have implemented the NNSRM method described above using a polynomial kernel and an RBF kernel. The polynomial kernel is given by

$$K^{\mathrm{poly}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + 1)^d \tag{12}$$

where d is the order of the polynomial space, which is treated as a parameter. The RBF kernel is given in Eq. (5). We performed comparison tests with synthetic and real data. The simulations consist of five independent runs of NNSRM and SVM classifiers in polynomial spaces of varying orders. The synthetic classification data is a set of 2-dimensional vectors embedded in a 4 by 4 checkerboard configuration. Fig. 1 shows such a configuration—an ordered pair of uniformly distributed random variables $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ is generated. If $\boldsymbol{x}_i$ lies in a black square, it is assigned to class label −1 and if in a white square, is assigned to class label +1. The data is split in the ratio of 60 to 40 into a training set (600 samples) and a test set (400 samples). SVM implementations were adapted from the SVM$^{\mathrm{light}}$ (Joachims, 1999), using $C = 100$ (tradeoff between training error and the margin). This particular choice for C in this experiment as well as the values chosen for the following experiments reflect C values for which the classifier performed best among other several possibilities.

Table 1 tabulates the results obtained by using NNSRM and SVM using the polynomial kernel for varying polynomial orders. The time taken to perform the training is also included for comparative purposes. All experiments were conducted on a Pentium II 450 MHz dual processor computer running Windows 2000. The superior performance of the NNSRM classifier over a variety of parameter values is evident.

Real data has been obtained from the UCI machine learning repository (Blake and Merz, 1998). One of the chosen databases is the Wisconsin Breast Cancer Database. The data consists of 683 samples with nine features each. The original data consists of more than 683 samples—the samples with incomplete feature vectors have been pruned. The dataset is divided into 409 training samples 274 test samples. Classifier performance for the polynomial kernel obtained as averages of five independent runs is tabulated in Table 2. For the SVM implementation, C was set at 10.

Another real dataset obtained from the UCI machine learning repository (Blake and Merz, 1998) is one that is popularly referred to as the Monks Problem dataset that consisted of 432

Table 1
Performance results for the polynomial kernel applied to synthetic data

d	NNSRM			SVM		
	Train (s)	# SV	% Error	Train (s)	# SV	% Error
0.1	0.4758	423	5.50	0.2113	555	54.25
0.25	0.4700	417	5.00	0.2078	555	54.25
0.5	0.4584	409	5.00	0.1914	555	54.25
1	0.5470	411	5.50	0.4614	558	54.25
2	0.7705	428	5.25	1.0802	533	43.75
3	1.0733	446	5.50	1.5668	530	44.75
4	1.2699	465	6.25	1.1493	525	44.00
5	1.4023	479	6.50	1.1191	517	44.50
7	1.5262	502	6.75	1.8856	471	40.25
8	1.4758	506	7.25	2.5546	458	36.00
9	0.9206	453	7.50	3.7756	447	33.25
10	0.8954	456	7.50	4.4891	427	28.25
11	0.8962	461	7.50	5.0679	406	24.00
13	0.8119	467	7.25	5.5696	356	24.00

Note: Time in seconds, obtained using Win32 performance counters, truncated to four decimal places.

Table 2
Performance results for the polynomial kernel applied to the Wisconsin Breast Cancer data

d	NNSRM			SVM		
	Train (s)	# SV	% Error	Train (s)	# SV	% Error
0.1	0.1603	168	4.01	0.1591	301	32.48
0.25	0.3032	225	4.38	0.1389	301	32.48
0.5	0.3749	277	3.65	0.1433	301	32.48
1	0.3358	310	4.01	0.0845	29	32.48
2	0.3823	328	4.01	0.2157	33	4.38
3	0.3479	327	4.01	0.4163	36	6.57
4	0.3440	327	3.65	0.4710	34	5.84
5	0.4029	327	3.65	0.3979	33	6.57
7	0.5184	327	3.28	0.3200	35	5.84
8	0.3601	327	3.28	0.3480	36	6.20
9	0.3571	326	3.65	0.3593	36	6.57
10	0.3582	326	3.65	0.3216	33	6.57
11	0.3358	326	3.65	0.3707	34	6.57
13	0.3284	326	4.01	0.4238	30	7.66

Note: Time in seconds, obtained using Win32 performance counters, truncated to four decimal places.

samples of six features each. The dataset has the training and test data in separate files–hence the random attribute of selecting training (128 of the original 432 samples) and test sets is eliminated. Classifier performance for the polynomial kernel obtained as averages of five independent runs is tabulated in Table 3. For the SVM implementation, C was again set at 10. It needs to be noted that with the SVM implementation, the results were highly sensitive to the choice of C.

A direct comparison between SVM and NNSRM based solely upon the number of support vectors is difficult because of the varying complexity of the spaces involved; but is presented here for reference purposes.

From the tabulated results, it may be seen that the number of support vectors in the synthetic data are of a comparable order between NNSRM and SVM, suggesting that the NNSRM approximation is close to the SVM output. However, the classi-

Table 3
Performance results for the polynomial kernel applied to the Monks data

d	NNSRM			SVM		
	Train (s)	# SV	% Error	Train (s)	# SV	% Error
0.1	0.0178	98	14.12	0.0744	124	50.00
0.25	0.0198	91	14.81	0.0548	124	50.00
0.5	0.0163	96	15.28	0.0744	124	50.00
1	0.0123	77	16.90	0.2064	84	50.00
2	0.0159	105	16.44	16.4271	35	14.81
3	0.0171	113	16.20	4.0678	33	8.56
4	0.0175	112	17.36	0.8824	33	8.56
5	0.0176	110	17.82	1.2798	35	9.26
7	0.0194	117	21.30	0.5892	33	10.88
8	0.0203	118	21.99	0.5301	32	11.57
9	0.0200	119	21.76	0.4109	30	10.88
10	0.1992	118	22.22	0.4883	37	10.88
11	0.0203	119	22.45	0.3884	46	11.11
13	0.0205	119	22.22	0.1765	61	10.42

Note: Time in seconds, obtained using Win32 performance counters, truncated to four decimal places.

fication accuracies are much higher for the NNSRM solution. The same argument does not however hold true for the Wisconsin Breast Cancer and the Monks databases.

Our experiments with RBF kernels affirm an observation noted in Section 3—the NNSRM technique returns (as expected) the same support vectors independent of the RBF parameters chosen, giving us the same classification results for the all values of τ; while SVMs performed better for increasing values of τ (as it begins to lose generalizability). Table 4 shows the classification performance for RBF kernels on the Wisconsin Breast cancer data. Note that the performance of the NNSRM classifier is "guaranteed" for the RBF kernel while that for the SVM classifier is parameter dependent. In other words, the performance of the NNSRM algorithm using the RBF kernel does not change with the scale parameter τ while the SVM classifier has classification accuracies varying over a large range.

Table 4
Performance results for the RBF kernel applied to the Wisconsin Breast Cancer data

τ	NNSRM			SVM		
	Train (s)	# SV	% Error	Train (s)	# SV	% Error
0.1	0.1924	363	2.92	0.0911	35	3.65
0.25	0.1900	363	2.92	0.0853	35	3.65
0.5	0.1910	363	2.92	0.1115	37	3.28
1	0.1971	363	2.92	0.0927	53	4.01
2	0.1921	363	2.92	0.1101	70	5.11
3	0.1981	363	2.92	0.1053	92	4.38
4	0.1988	363	2.92	0.1094	116	3.65
8	0.1893	363	2.92	0.1346	170	3.28
16	0.1871	363	2.92	0.1382	204	4.01
32	0.1805	363	2.92	0.1552	234	6.57
64	0.1916	363	2.92	0.1515	265	9.12
96	0.1900	363	2.92	0.1566	288	9.49
128	0.1897	363	2.92	0.1618	291	10.95

Note: Time in seconds, obtained using Win32 performance counters, truncated to four decimal places.

The number of support vectors retained by the SVM and NNSRM classifiers unveils an interesting aspect of kernel transformations on the geometry of the original classification problem. Conceptually, the non-linear mappings are employed in order to represent the data in a much higher dimensional space where the two classes could exhibit better (linear) separability, providing more accurate classification. We see, however, that the RBF kernel transformation does not affect the original order of pairwise distances at all, resulting in the same NNSRM classifier as in the original space. The maximum margin linear classifier in the RBF space, on the other hand, evolves with varying scale parameter, and structures different classifications. In the case of a polynomial kernel, both classifiers adapt to the changing data structure for varying order parameter, albeit with generally more reference points for the NNSRM classification than the SVM. These observations suggest that the concept of separability of two classes should be considered irrespective of the spaces in which they are embedded. Furthermore, without any structural a priori information on the data, constructing good classifiers amounts to finding the most suitable representation of the data conditional on the classifier model of choice, which still remains a guessing game for SVM classifiers to a large extent. Experimental results presented above indicate that the NNSRM formulation is not only computationally less expensive, but also much more robust to varying data representations than SVMs.

5. Conclusion

With higher orders of the polynomial we increase the dimensionality of the transform space, hence losing generalizability. This is clearly evident for both classifiers. However, the SVM classifier performs comparatively better (in the case of the synthetic data) with high polynomial orders (higher dimensionality) while the NNSRM classifier loses its accuracy (albeit a small amount) because of the approximate nature of the NNSRM solution. This performance however begins to drop with very large orders. The SVM classifier implementations suffer from the problem of choosing the right parameters to guarantee convergence. The NNSRM method however, in providing an approximate solution, guarantees convergence for any choice of kernel parameters in a reasonable length of time, without having to set a suitable C value to control the tradeoff between training error and the margin.

In its current form, the proposed classifier does not employ special treatment for outliers: keeping the training error at zero while minimizing the cardinality of the reference set is an approximate way of minimizing the upper bound on the test risk. The effects of disregarding possible outliers on the performance are then taken at face value in the simulation results. It is conceivable that if a limited number of training errors are acceptable, one may achieve a lower upper bound. Such a strategy, on the other hand, is significantly harder to follow because it entails deciding on a compromise between the number of reference points and the number of allowable training errors in outliers.

We presented an algorithm of small polynomial complexity that uses the SRM principle using the NN approach. We also compared performance of NNSRM and SVM classifiers using timing and accuracy as performance measures. In terms of providing a consistent classifier accuracy for various choices of kernel parameters, guaranteed convergence and fast computation, the NNSRM method clearly outperforms the SVM classifier. The proposed technique achieves robustness to kernel parameters and significantly lower computational cost at the expense of a small decrease in the classification performance, when compared to conventional SVMs. Consequently, in applications where the expense of computations prohibits conventional SVMs, the NNSRM method provides a viable and powerful alternative.

References

Bhattacharya, B., Kaller, D., 1998. Reference set thinning for the k-nearest neighbor decision rule. In: Proc. IEEE Internat. Conf. on Pattern Recognition, vol. 1, pp. 238–242.

Blake, C., Merz, C., 1998. UCI repository of machine learning databases. Available from <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Cerveron, V., Ferri, F., 2001. Another move towards the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule. IEEE Trans. Systems, Man Cybernet. B 31 (3), 408–413.

Chidananda-Gowda, K., Krishna, G., 1979. The condensed nearest neighbor rule using the concept of mutual nearest neighbourhood. IEEE Trans. Inform. Theory 25 (4), 488–490.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learn. 20, 273–297.

Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

Dasarathy, B., 1994. Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Trans. Systems Man Cybernet. 24 (3), 511–517.

Devijver, P.A., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice Hall, London.

Farach-Colton, M., Indyk, P., 1999. Approximate nearest neighbor algorithms for Hausdorff metrics via embeddings. In: 40th Annual Symposium on Foundations of Computer Science.

Gates, G.W., 1972. The reduced nearest neighbor rule. IEEE Trans. Inform. Theory 18 (3), 431–433.

Hart, P.E., 1968. The condensed nearest neighbor rule. IEEE Trans. Inform. Theory 14 (3), 515–516.

Joachims, T., 1999. Making large-scale svm learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods—Support Vector Learning. MIT Press.

Karaçalı, B., Krim, A., 2003. Fast minimization of the structural risk by nearest neighbor rule. IEEE Trans. Neural Networks 14 (1), 127–137.

Kuncheva, L., Bezdek, J., 1998. Nearest prototype classification: Clustering, genetic algorithms, or random search? IEEE Trans. Systems Man Cybernet. 28 (1), 160–164.

Osuna, E., Freund, R., Girosi, F., 1997. Training support vector machines: An application to face detection. In: Proc. Computer Vision and Pattern Recognition.

Ritter, G., Woodruff, H., Lowry, S., Isenhour, T., 1975. An algorithm for a selective nearest neighbor decision rule. IEEE Trans. Inform. Theory 21, 665–669.

Schölkopf, B., 2000. Kernel trick for distances. In: Leen, T.K., Dietterich, T.G., Tresp, V. (Eds.), Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA. MIT Press, pp. 301–307.

Schölkopf, B., Burges, C., Smola, A. (Eds.), 1999. Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge.

Tomek, I., 1976. An experiment with the edited nearest neighbor rule. IEEE Trans. Systems Man Cybernet. 6 (6), 448–452.

Vapnik, V., 1998. Statistical learning theory. In: Haykin, S. (Ed.), Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons.

Vapnik, V., Golowich, S., Smola, A., 1997. Support vector method for function approximation, regression estimation, and signal processing. In: Advances in Neural Information Systems, vol. 9. MIT Press, Cambridge.

Wilson, D.R., Martinez, T.R., 2000. Reduction techniques for instance-based learning algorithms. Machine Learn. 38 (3), 257–286.

Yu, K., Ji, L., Zhang, X., 2002. Kernel nearest-neighbor algorithm. Neural Process. Lett. 15 (2), 147–156.