Egyptian Informatics Journal (2012) **13**, 177–183

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification

E.A. Zanaty

Mathematics Dept., Computer Science Section, Faculty of Science, Sohag University, Sohag, Egypt

Received 5 January 2012; revised 6 August 2012; accepted 15 August 2012
Available online 13 September 2012

KEYWORDS

Neural networks;
Support vector machine;
Kernel functions;
Quadratic Programming (QP)

Abstract In this paper, we introduce a new kernel function for improving the accuracy of the Support Vector Machines (SVMs) classification. The proposed kernel function is stated in general form and is called Gaussian Radial Basis Polynomials Function (GRPF) that combines both Gaussian Radial Basis Function (RBF) and Polynomial (POLY) kernels. We implement the proposed kernel with a number of parameters associated with the use of the SVM algorithm that can impact the results. A comparative analysis of SVMs versus the Multilayer Perception (MLP) for data classifications is also presented to verify the effectiveness of the proposed kernel function. We seek an answer to the question: "which kernel can achieve a high accuracy classification versus multi-layer neural networks". The support vector machines are evaluated in comparisons with different kernel functions and multi-layer neural networks by application to a variety of non-separable data sets with several attributes. It is shown that the proposed kernel gives good classification accuracy in nearly all the data sets, especially those of high dimensions. The use of the proposed kernel results in a better, performance than those with existing kernels.

1. Introduction

Learning with spatially localized basis functions has become a popular paradigm in machine learning community. In the context of radial basis function networks [1–5], it was demonstrated that these learning methods offer an alternative to learning with global basis functions, such as sigmoidal neural networks. Today, support vector machines and along with other learning based-kernel algorithms show better results than artificial neural networks and other intelligent or statistical models, on the most popular benchmark problems [6]. The scarcity of the model results from a sophisticated local learning that matches the model capacity to the data complexity ensuring a good performance on the future, previously unseen, data. They gave a single solution characterized by the global minimum of the optimized functional and not multiple solutions associated with the local minima as in the case of neural networks. Moreover, support vector machines do not rely so heavily on heuristics, i.e. an arbitrary choice of the model and have a more flexible structure [7].

Classification is one of the important machine learning operations. It is the operation that enables organizations to

E-mail address: zanaty22@yahoo.com
Peer review under responsibility of Faculty of Computers and Information, Cario University.

Production and hosting by Elsevier

1110-8665
http://dx.doi.org/10.1016/j.eij.2012.08.002

discover patterns in large or complex data sets. Classification and function approximation using SVMs are formulated as Quadratic Programming (QP) problems which can be solved efficiently by using many well-documented optimization algorithms [8–10]. The neural networks may be regarded as the universal classifications of the measured data in the multi-dimensional space. They realize two types of classification: the global and local one [11]. The most important example of global network is the Multi-Layer Perception (MLP), employing the sigmoid activation function of neurons. In MLP the neurons are arranged in layers, counting from the input layer (the set of input nodes), through the hidden layers, up to the output layer. The interconnections are allowed only between two neighboring layers. The network is feed forward, i.e., the processing signals propagate from input to the output side.

The most representative example of local neural network is the Support Vector Machine (SVM) of different kernels functions. Choosing different kernel functions will produce different SVMs and may result in different performances [12,13]. In the SVM literature, there exist polynomials SVMs, radial basis function of SVMs, two-layer Neural Network (NN) SVMs and so on [11]. They correspond to the kernel functions of polynomial, radial basis function and two-layer NN, respectively. Once the kernel is fixed, SVM classifiers have only a user-chosen parameter (the error penalty), but the kernel is a very big rug under which many parameters have to be determined. Some works have been done on limiting kernels using prior knowledge, but the best choice of a kernel for a given problem is still an open research issue [14–16].

One issue is identifying an appropriate kernel for the given data. Most algorithms rely on a priori knowledge to select the correct kernel. Tsang et al. [16] discussed a way to take advantage of the approximations inherent in kernel classifiers, by using a minimum enclosing ball algorithm as an alternative means of speeding up training. Training time had previously been reduced mostly by modifying the training set in some way. Their Core Vector Machine converged in linear time with space requirements independent of the number of data points. Also, Kwok and Tsang [17], they applied it to the kernel PCA problem successfully, but also found that it did not perform well when applied to the reduced set problem. It is interesting to note that the methods of [17,18] can both be applied to pre-image applications with a discrete input space, since they do not require the gradient of the objective function. Generally, in implementations of this method, the time and space complexities are very high because the core of the SVMs is based on approximate minimum enclosing ball algorithms which are computationally expensive.

Maji et al. [19] presented a technique for the exact evaluation of intersection kernel SVMs which is logarithmic in time. They have shown that the method is relatively simple and the classification accuracy is acceptable, but the runtimes are significantly increased compared with the established Radial Bases Function (RBF) and Polynomials (POLY) kernels due to large number of SV for each classifier [20,21].

Zanaty et al. [22,23] described the SVMs with combination the both RBF and POLY functions to take advantage of their respective strengths. They introduced kernel functions called Polynomial Radial Basis Function (PRBF).

Despite the maturity of classification, problems remain, especially in choosing the most appropriate kernel of SVMs for a particular application [24,25]. In other words, exploration of new techniques and systematic methodology to construct an efficient kernel function for designing SVMs in a specific application is an important research direction in SVMs. Several survey papers on comparing SVMs with Gaussian Kernels to Radial Basis Function Classifiers can be found in literature, but these focus on a sub-set of techniques and often only on performance accuracy [24–27].

In this study, a new kernel function called Gaussian Radial basis Polynomial Function (GRPF) is introduced that could improve the classification accuracy of Support Vector Machines (SVMs) for both linear and non-linear data sets. The aim is to train Support Vector Machines (SVMs) with different kernels compared with back-propagation learning algorithm in classification task. Moreover, we compare the proposed algorithm to algorithms based on both Gaussian and polynomial kernels by application to a variety of non-separable data sets with several attributes. It is shown that the proposed kernel gives good classification accuracy in nearly all the data sets, especially those of high dimensions.

The rest of this study is organized as follows: In Section 2, the SVM classifier is described. The multi-layer perception classifier is designed in Section 3. In Section 4, the SVMs are presented with a new kernel function. The kernel parameters are optimized in Section 5. Section 6 gives comparison results between support vector machines and multi-layer perception classifier. Our conclusion is presented in Section 7.

2. SVM classifiers

The details of SVM classification are described in [28–32]. We outline the basic equations and we follow the notations of Schölkopf et al. [29]. Let x_i for $1 \leqslant i \leqslant N_x$ be the input vectors in input space, with corresponding binary labels $y_i \in \{-1, 1\}$ (i.e. y_i takes 1 if x_i is in class 1 and takes -1 if x_i is in class 2). Let $\phi(x_i)$ be the corresponding vectors in feature space, where, $\phi(x_i)$ is the implicit kernel mapping and let $K(x_i, x_j) = \phi(x_i) \bullet \phi(x_j)$ be the kernel function, implying a dot product in the feature space. Popular kernel functions include classical kernels and recent kernels as shown in ([32]).

The optimization problem for a soft-margin SVM is:

$$\min_{w,b}\left\{\frac{1}{2}\|w\|^2 + C\sum_i \xi_i\right\} \tag{1}$$

Subject to the constraints $y_i(w.x + b) = 1 - \xi_i$ and $\xi_i \geqslant 0$ where w is the normal vector of the separating hyper plane in feature space and $C > 0$ is a regularization parameter controlling the penalty for misclassification. Eq. (1) is referred to as the primal equation. From the Lagrangian form of (1), we derive the dual problem:

$$w(\alpha) = \max_{\alpha}\left\{\sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(x_i,\ x_j)\right\} \tag{2}$$

Subject to $0 \leqslant \alpha_i \leqslant C$. This is a quadratic optimization problem that can be solved efficiently using algorithms such as Sequential Minimal Optimization [32]. Typically, many α_i go to zero during optimization and the remaining x_i corresponding to those $\alpha_i > 0$ are called support vectors. To simplify

notation, from here on we assume that all non-support vectors have been removed, so that N_x is now the number of support vectors and $\alpha_i > 0$ for all i. With this formulation, the normal vector of the separating plane w is calculated as:

$$w = \sum_{i=1}^{N_x} \alpha_i y_i x_i \tag{3}$$

Note that because $\phi(x_i)$ is defined implicitly, w exists only in feature space and cannot be computed directly. Instead, the classification $f(q)$ of a new query vector q can only be determined by computing the kernel function of q with every support vector:

$$f(q) = \text{sign}\left(\sum_{i=1}^{N_x} \alpha_i y_i . k(q,\ x_i) + b\right) \tag{4}$$

where the bias term b is the offset of the hyperplane along its normal vector, determined during SVM training [33].

3. Multi-level perception

The Multilayer Perception (MLP) is perhaps the most popular network architecture in use today both for classification and regression. MLPs are feed forward neural networks which are typically composed of several layers of nodes with unidirectional connections, often trained by back propagation [34,35]. The learning process of MLP network is based on the data samples composed of the N-dimensional input vector x and the M-dimensional desired output vector d, called destination. By processing the input vector x, the MLP produces the output signal vector $y(x, w)$ where w is the vector of adapted weights. The error signal produced actuates a control mechanism of the learning algorithm. The corrective adjustments are designed to make the output signal $y_k (k = 1, 2, \ldots, M)$ to the desired response d_k in a step by step manner.

The learning algorithm of MLP is based on the minimization of the error function defined on the learning set (x_i, d_i) for $i = 1, 2, \ldots, N$ using the Euclidean norm:

$$E(w) = \frac{1}{2}\sum_{i=1}^{N} \|y(x_i,\ w) - d_i\|^2 \tag{5}$$

The minimization of this error leads to the optimal values of weights. The most effective methods of minimization are the gradient algorithms, from which the most effective is the Levenberg–Marquard algorithm for medium size networks and conjugate gradient for large size networks. Generally in all gradient algorithms the adaptation of weights is performed step by step according to the following scheme:

$$w(k+1) = w(k) + \gamma p(k) \tag{6}$$

In this relation $p(k)$ is the direction of minimization in kth step, γ is the learning coefficient, and w is the adaptation coefficient. Various learning methods differ in the way the value of $p(k)$ is generated.

In Levenberg–Marquardt approach the least square formulation of learning problem is exploited:

$$E(w) = \frac{1}{2}\sum_{i=1}^{M} (y_i(w) - d_i)^2$$

and solved by using second order method of Newton type:

$$p(k) = -G(k)^{-1} g(k) \tag{7}$$

where $g(k) = \frac{\partial E}{\partial w(k)}$ is the gradient of error function Eq. (5) and $G(k)$ is the Hessian approximation ([34,35]), determined by applying the Jacobian matrix $J(k)$:

$$G(k) = J(k)^T J(k) + v. \tag{8}$$

In this equation the Jacobian matrix J is equal $J = \frac{\partial e}{\partial w}$ and $e = [y_i(w) - d_i, \ldots, y_M(w) - d_M]^T$.

The variable is the Levenberg–Marquard parameter adjusted step by step in a way to provide the positive definiteness of Hessian G (the value of is eventually reduced to zero).

In conjugate gradient approach, most effective for large networks, the direction $p(k)$ is evaluated according to the formula:

$$p(k) = -g(k) + \beta p(k-1) \tag{9}$$

where the conjugate coefficient β is usually determined according to the Polak–Ribiere rule:

$$p(k) = \frac{g(k)^T (g(k) - g(k-1))}{g(k-1)^T g(k-1)}$$

In the weight update Eq. (6) the learning coefficient γ should be adjusted by the user. It is done usually by applying so called adaptive way [32], taking into account the actual progress of minimization of the error function.

4. Proposed kernel functions

A critical step in support vector machine classification is choosing a suitable kernel of SVMs for a particular application, i.e. various applications need different kernels to get reliable classification results. It is well known that the two typical kernel functions often used in SVMs are the radial basis function kernel and polynomial kernel. More recent kernels are presented in [28–32] to handle high dimension data sets and are computationally efficient when handling non-separable data with multi attributes. However, it is difficult to find kernels that are able to achieve high classification accuracy for a diversity of data sets. In order to construct kernel functions from existing ones or by using some other simpler kernel functions as building blocks, the closure properties of kernel functions are essential [36,37].

For given non-separable data, in order to be linearly separable, a suitable kernel has to be chosen. Classical kernels, such as Gauss RBF and POLY functions, can be used to transfer non-separable data to separable, but their performance in terms of accuracy is dependent on the given data sets. The following POLY function performs well with nearly all data sets, except high dimension ones [36]:

$$\text{POLY}(x,\ z) = (x^T z + 1)^d \tag{10}$$

where d is the polynomial degree.

The same performance is obtained with the Gauss RBF of the following form [3]:

$$\text{RBF}(x,\ z) = \exp(-\gamma \|x - z\|^2) \tag{11}$$

where γ is appositive parameter controlling the radius. Zanaty et al. [23] in presented the Polynomial Radial basis Function (PRBF) as:

$$\text{PRBF} = ((1 + \exp(\omega))/V)^d \tag{12}$$

where $\omega = |x - z|$, $V = p * d$ and p is a prescribed parameter. Completely achieving a SVM with high accuracy classification therefore, requires specifying high quality kernel function.

Here, we combine POLY, RBF, and PRBF into one kernel to become:

$$GRPF(x,\ z) = \left(\frac{d + r \cdot \exp(-\|X - z\|^r/(r \cdot \sigma^2))}{r + d}\right)^{d+1} \tag{13}$$

where σ is a statistic distribution of the probability density function of the input data; and the values of $r(r > 1)$ and d can be obtained by optimizing the parameters using the training data. The proposed kernel has the advantages of generality. However, The existing kernels such as PRBF and proposed in Zanaty et al. [22], Gaussian and polynomials kernel function by setting d and r in different values. For example if $d = 0$, we get Exponential Radial when $r = 1$ and Gaussian Radial for $r = 2$ and so on. Moreover various kernels can be obtained by optimizing the parameters using the training data.

5. Optimizing the kernel parameters

Let's go back to the SVM algorithm. We assume that the kernel GRPF depends on two parameters d and r, encoded into a vector $\theta = (d, r)$. We thus consider a class of decision functions parameterized by α, b, and θ:

$$f_{\alpha,\ b,\ \theta}(x) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y_i \text{GRPF}_\theta(x,\ z) + b\right) \tag{14}$$

We want to choose the values of the parameters α and θ such that w (in Eq. (2)) is maximized (maximum margin algorithm) and T, the model selection criterion, is minimized (best kernel parameters). More precisely, for θ fixed, we want to have $\alpha^0 = \arg\max w(\alpha)$ and choose θ^0 such that:

$$\theta^0 = \arg \min_\theta T(\alpha^0, \theta). \tag{15}$$

When, θ is a one dimensional parameter, one typically tries a finite number of values and picks the one which gives the lowest value of the criterion T. When both T and the SVM solution are continuous with respect to θ a better approach has been proposed by Cristianini et al. [38]. They used an incremental optimization algorithm, one can train an SVM with little effort when θ is changed by a small amount. However, as soon as θ has more than one component computing $T(\alpha, \theta)$ for every possible value of θ becomes intractable, and one rather looks for a way to optimize T along a trajectory in the kernel parameter space.

In this work, we use the gradient of a model selection criterion to optimize the model parameters (see [39,40] for more discussions). This can be achieved by the following iterative procedure:

1. Initialize θ to some value.
2. Using a standard SVM algorithm, find the maximum of the quadratic form $w\ \alpha^0 = \arg\max w(\alpha)$.
3. Update the parameters θ such that T is minimized. This is typically achieved by a gradient step (Chapelle et al. [39] for these calculations).
4. Go to step 2 or stop when the minimum of T is reached.

Solving step 3 requires estimating how T varies with θ where GRPF_θ can be differentiated with respect to θ, more discussions can be shown in [39].

6. Experimental results

6.1. Data sets and parameter optimization

In this section, we perform classification experiments on a number of toy and real-world data sets 2: Letter, Pendigits, Waveform, Satimage, DNA, Segment, ABE, and Zoo (Table 1). In Table 1, we describe these data sets, but for more details can be seen in [37]. The eight data have also been used in [26] and (Zanaty et al. [22,23]) for the single-kernel SVM. Each data set has different training/test splits to give use the reliable behavior of the proposed kernel. We can divide these data sets according to the training set size into two types, large data sets $(1 \rightarrow 4)$ and small ones $(5 \rightarrow 8)$. In the proposed algorithm, we used the optimization toolbox of Matlab to perform it. It includes second order updates to improve convergence speed. Although we used a loose stopping criterion, good value of the parameters minimizing the functional is obtained. Experimental results in Table 1 demonstrated that values of the parameter r are increased according to increasing attribute values while the values of d are decreased in the case of largest class values. In a way this renders the technique even more applicable since this estimate is very simple. The proposed algorithm used loose stopping criterion for avoiding holding out some data for validation and thus makes full use of the training set for the optimization of parameters, contrary to cross-validation methods.

6.2. Comparative results

In order to evaluate the performance of the support vector machine with different kernels, we carried out some experiments with different data sets from machine learning benchmarks domains. We design a multi-class support vector machine classifier based on one versus one algorithm using the voting strategy [41]. For C classes, C(C-1)/2 binary classifiers are constructed. The performance evaluation of the each support vector machine using different kernel and the multilayer neural networks done using the following equation:

$$Accuracy = (n/N) \cdot 100 \tag{16}$$

where n is the number of correct classified examples and N is the total number of the test examples.

We have experiment MLP network trained by using Levenberg–Marquardt algorithm and SVM with POLY, RBF, PRBF and GRPF functions trained by applying method [35,16]. The training time of MLP was approximately 10 times longer than SVM. These data sets have given to the algorithm with different sizes (classes and attributes). Through the

Table 1 Data sets and optimized parameters.

Data set	Classes	Attributes	Train	Test	d	r
Letter	26	16	15,000	5000	3.876	2.043
Pendigits	9	16	7435	3448	2.709	2.765
Waveform	3	21	4700	300	2.987	3.871
Satimage	6	36	4435	2000	2.765	3.892
DNA	3	180	2686	500	1.098	5.987
Segment	7	18	1810	500	1.704	3.863
ABE	3	16	1763	560	2.098	2.861

experiments, we set parameter $\sigma = 0.5$ (Gaussian RBF kernel width), and $d = 2$ for RBF, POLY, and GRPF.

Table 2 presents the classification accuracy for each data set using support vector machine with different kernel versus the multilayer neural networks.

The proposed method gives the best accuracy in most data sets. Table 3 presents the mean accuracy obtained from all kernels; it is obvious that the GRPF kernel obtains the best mean accuracy (95.91) compared to the classical RBF, POLY and PRBF. Moreover, Table 3 shows the comparison between the SVMs and MLPs classifiers, it is clear that the SVMs with the kernel (GRPF) achieve higher accuracy than MLP classifier and other kernels. Fig. 1 describes the mean accuracy of SVMs and MLP classifiers applied to all data sets. It is clear that the proposed kernel gives the best mean accuracy compared to the POLY, RBF and PRBF functions.

For future research, we focused on building a new hybrid classifier of SVMs and MLP to overcome the disadvantages of both algorithms.

6.3. Discussions

In this part of study, we attempt to investigate the best choice among SVM kernels for data classification task. Table 2 presented the performance of SVMs trained with POLY, RBF, PRBF and GRPF kernels on the different data sets.

From Table 2, the RPF function, with $p = 3$, $d = 2$ gives better accuracy with small data sets ($5 \rightarrow 8$) than the Polynomial function. The Polynomial kernel with two dimension data $d = 2$, gives better results than RBF for large sets ($1 \rightarrow 4$). Moreover, we noted that MLP achieved better results than POLY and RBF for these large sets ($1 \rightarrow 4$). The proposed kernel, GRPF, gives the best accuracy in nearly all the data sets. For data set 7 (ABE), although the size of data set is small and one feature is enough to distinguish a pair of ABE classes, the proposed kernel still achieves good results (the accuracy is of GRPF 97.98%) as shown in Fig. 2 shows the relation between the accuracy of RBF, POLY, PRBF, GRPF and the class number. As in Fig. 2, we noted that the accuracy of GRPF, RBPF and MLP are better than RBF and POLY kernels even in case of largest number of classes. For small class and attribute (data set 7), we noted that MLP achieves the worst accuracy.

Interestingly, as the number of attributes increases, the improvement in accuracy of the proposed method compared with POLY, RBF and PRBF kernels increases as shown in Fig. 3 (data set 5 which has the largest number of attributes). This improved performance is due to the fact that the proposed function is more complex and combines the performance of both its parents, RBF and Polynomial functions.

For largest attribute (data set 5), the proposed GRPF gives the best accuracy (99.88) at all and the MLP obtains the worst accuracy (68.4) at all.

Table 2 SVMs and MLP.

MLP	POLY	RBF	PRBF	GRPF
93.16	89.54	88.87	95.9	96.98
85.09	77.98	75.98	85	90.8
90.67	80.76	81.67	91.84	95.98
88.65	80.66	80.34	91.57	97.43
68.4	88.54	90.34	96.32	99.88
72.1	91.64	93.65	95.65	96.43
92.82	93.12	97.97	95.72	97.98
88.32	72.84	77.34	87.09	90.87

Table 3 The mean accuracy SVMs and MLP accuracy.

	Support vector machine	Multilayer neural networks
Poly	84.38	
RBF	85.77	
PRBF	92.38	84.90
GRPF	95.79	

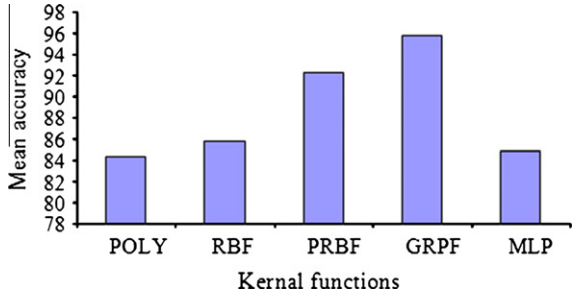

Figure 1. The relation between mean accuracy of SVMs kernels and MLP.

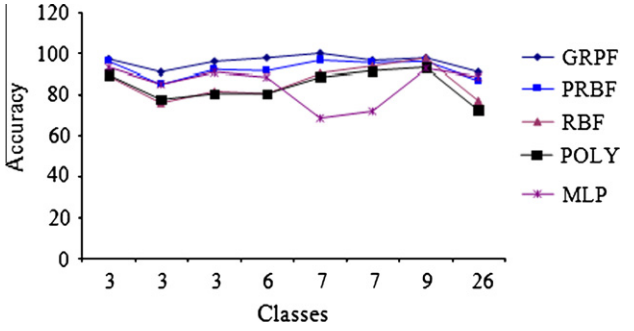

Figure 2. The relation between accuracy of SVMs and MLP classifiers; and class number.

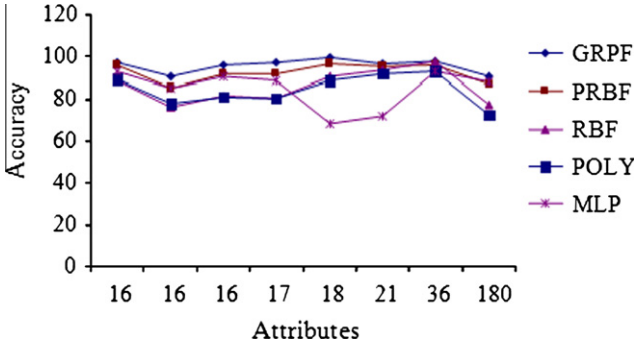

Figure 3. The relation between accuracy of SVMs and MLP classifiers; and attributes number.

7. Conclusion

In this study, we have constructed SVMs and computed its accuracy. We held a comparison between the SVMs and MLP classifier. We have used different sizes of data sets with different attributes. Then, we have compared the results of the SVM algorithm to MLP classifier. The proposed GRPF kernel has achieved the best accuracy, especially with the data sets with many attributes.

Comparing the classification accuracy of the support vector machine to the multilayer neural networks learning algorithms, it is obvious from that support vector machines with the proposed new kernel function (GRPF) accomplishes better accuracy than multilayer networks, especially in high dimension data sets.

In MLPs classifiers, the tested data sets need more hidden units and the complexity is controlled by keeping the number of these units small, whereas the SVMs complexity does not depend on the dimension of the data sets. SVMs based on the minimization of the structural risk, whereas MLP classifiers implement empirical risk minimization. So, SVMs are efficient and generate near the best classification as they obtain the optimum separating surface which has good performance on previously unseen data points.

However, the main difference is in the complexity of the networks. The MLP network implementing the global approximation strategy usually employs very small number of hidden neurons. On the other side the SVM is based on the local approximation strategy and uses large number of hidden units. The great advantage of SVM approach is the formulation of its learning problem, leading to the quadratic optimization task. It greatly reduces the number of operations in the learning mode. It is well seen for large data sets, where SVM algorithm is usually much quicker.

References

[1] Duda R, Hart P, Stork G. Pattern classification. 2nd ed. UK: Wiley Interscience; 2001.
[2] Boser B, Guyon I, Vapnik V, A training algorithm for optimal margin classifiers. In: Proceeding COLT '92 proceedings of the fifth annual workshop on computational learning, Pittsburgh, PA, USA; 1992. p. 144–52.
[3] Taylor SJ, Cristianini N. Kernel methods for pattern analysis. Cambridge University; 2004.
[4] Osuna E, Freund R, Girosi F. Support vector machines, training and applications. Artificial Intelligence Laboratory, MIT Press, Series, Report, AIM-1602, CBCL-144, 1997.
[5] Boudat G, Anour F. Kernel-based methods and function approximation. In: International Joint Conference on Neural Networks (IJCNN). Washington: IEEE Press; 2001. p. 1244–9.
[6] Meyer D, Leisch F, Hornik K. The support vector machines under test. Neurocomputing 2003;55:169–86.
[7] Huang TM, Kecman V, Kopriva I. Kernel based algorithms for mining huge data sets, supervised, semi-supervised and unsupervised learning. Berlin, Heidelberg: Springer-Verlag; 2006.
[8] Vapnik V. The nature of statistical learning theory. New York, NY, USA: Springer New York Inc.; 1995.
[9] Cortes C, Vapnik V. Support vector networks. Mach Learn 1995;20:273–97.
[10] Schölkopf B. Support vector learning, PhD dissertation, Technical University, Berlin, Germany; 1997.
[11] Mohammad Muamer N, Norrozila Sulaiman, Emad Khalaf T. A novel local network intrusion detection system based on support vector machine. J Comput Sci 2011;7:1560–4.
[12] Vapnik V, Golowich S, Smola A. Support vector method for function approximation. Regression estimation and signal processing. In: Advances in neural information processing systems, vol. 9, no. 9, 1997. p. 281–7.
[13] Williamson RC, Smola A, Schölkopf B. Entropy numbers operators and support vector kernels. Cambridge: MA: MIT Press; 1999.
[14] Chapelle O, Schölkopf B. Incorporating invariances in non-linear support vector machines. In: Proc. NIPS; 2001. p. 609–16.
[15] Lei H, Govindaraju V. Speeding up multi-class SVM evaluation by PCA and feature selection. Center for Unified Biometrics and Sensors (CUBS); 2005.
[16] Tsang IW, Kwok JT, Cheung PM. Core vector machines: fast SVMs training on very large data sets. J Mach Learn Res 2005;6:271–363.
[17] Kwok JT, Tsang IW. The pre-image problem in kernel method. IEEE Trans Neural Networks 2004;15:1517–25.
[18] Bakir GH, Weston J, Schölkopf B. Learning to find pre-images. Adv Neural Inf Proc Syst 2004;16:449–56.
[19] Maji S, Berg AC, Malik J. Classification using intersection kernel support vector machines is efficient. In: Conference on computer vision and pattern recognition, CVPR IEEE; 2008. p. 1–8.
[20] Haasdonk B. Feature space interpretation of SVMs with indefinite kernels. IEEE Trans Pattern Anal Mach Intell 2005;27(4):482–92.
[21] Hastie T, Hsu CW, Lin CJ. A comparison of methods for multiclass support vector machines. IEEE Trans Neural Networks 2005;13:415–25.
[22] Zanaty EA, Afifi A, Khateeb RE. Improving the accuracy of support vector machines via a new kernel functions. Int J Intell Comput Sci 2009;1:55–67.
[23] Zanaty EA, Aljahdali S, Cripps RJ. Accurate support vector machines for data classification. Int J Rapid Manuf 2009;1(2).
[24] Ektefa Mohammadreza, Sidi Fatimah, Ibrahim Hamidah, Jabar Marzanah, Memar Sara. A comparative study in classification techniques for unsupervised record linkage model. J Comput Sci 2011;7:341–7.
[25] Sonkamble Balwant A, Doye D. A novel linear-polynomial kernel to construct support vector machines for speech recognition. J Comput Sci 2011;7:991–6.
[26] Zanaty EA, Afifi A. Support vector machine (SVMs) with universal kernel. Appl Artif. Intell. 2011;25(7):575–89.
[27] DeCoste D, Mazzoni D. Fast query optimized kernel machine classification via incremental approximate nearest support vectors. In: Proceedings of the 20th international conference on machine learning, Washington, ICML 2003; 2003. p. 115–22.
[28] Schölkopf B, Smola AJ. Learning with kernels. Cambridge, MA: The MIT Press; 2002.
[29] Schölkopf BS, Mika CJ, Burges P, Knirsch RR, Muller KG. Input space versus feature space in kernel-based methods. IEEE Trans Neural Network 1999;10(5):1000–17.
[30] Burges CJ, Schölkopf B. Improving the accuracy and speed of support vector machines. Neural Inform Proc Syst 1997;9:375–81.
[31] Burges CJ. A tutorial on support vector machines for pattern recognition. Data Min Knowledge Discovery 1998;2(2):121–67.
[32] Schölkopf B, Burges C, Smola A. Advances in kernel methods support vector learning. MIT Press; 1998.
[33] Taylor SJ, Bartlett PL, Willianmson RC, Anthony M. Structural risk minimization over data-dependent hierarchies. IEEE Trans Inf Theory 1998;5:1926–40.
[34] Making JT. Large-scale SVM learning practical. In: Advances in kernel methods-support vector learning. Cambridge: MIT Press; 1999. p. 42–56.
[35] Osowski S, Siwek K, Markiewicz T. MLP and SVM networks: a comparative study. In: Proceedings of the 6th nordic signal

processing symposium NORSIG, Espoo, Finland; June 2004. p. 9–11.
[36] Cristianini N, Taylor SJ. An introduction to support vector machines. Cambridge, MA: Cambridge University Press; 2000.
[37] http://archive.ics.uci.edu/ml/.
[38] Cristianini N, Campbell C, Taylor SJ. Dynamically adapting kernels in support vector machines. In: Proceedings of the 1998 conference on advances in neural information processing systems II; 1999. p. 204–10.
[39] Chapelle O, Vapnik V, Bousquet O, Mukherjee S. Choosing multiple parameters for support vector machines. Mach Learn 2002;46:131–59.
[40] Bengio Y. Gradient-based optimization of hyper-parameters. Neural Comput 2000;12(8):1889–900.
[41] Rifin R, Klautau A. In defense of one vs. all classification. J Mach Learn Res 2004;5:101–41.