Neural Computing and Applications (2021) 33:9975–9991
https://doi.org/10.1007/s00521-021-05765-6

ORIGINAL ARTICLE

One novel class of Bézier smooth semi-supervised support vector machines for classification

En Wang[1] · Zi-Yang Wang[2] · Qing Wu[3]

Received: 12 August 2020 / Accepted: 21 January 2021 / Published online: 1 March 2021

Abstract
The semi-supervised support vector machine (S^3VM) for classification is introduced for dealing with quantities of unlabeled data in the real world. Labeled data are utilized to train the algorithm and then were adapted to classify the unlabeled data. However, this algorithm has several drawbacks, such as the non-smooth term of semi-supervised objective function negatively affects the classification precision. Moreover, it is required to endure heavy burden in solving two quadratic programming problems with inversion matrix operation. To cope with this problem, this article puts forward a novel class of Bézier smooth semi-supervised support vector machines (BS^4VMs), based on the approximation property of Bézier function to the non-smooth term. Because of this approximation, a fast quasi-Newton method for solving BS^4VMs can be used to decrease the calculating time scale. This new kind of algorithm enhances the generalization and robustness of S^3VM for nonlinear case as well. Further, to show how the BS^4VMs can be practically implemented, experiments on synthetic, UCI dataset, USPS dataset, and large-scale NDC database are offered. The theoretical analysis and experiments comparisons clearly confirm the superiority of BS^4VMs in both classification accuracy and calculating time.

Keywords Machine learning · Semi-supervised classification · Support vector machine · Smooth technique · Bézier function

1 Introduction

In the information age, mass production of information has caused serious information overloaded. Facing this dilemma, support vector machine (SVM), one kind of fast information classification algorithm, becomes one effective solution. As one kind of full-supervised statistical machine learning, support vector machine (SVM) has get widely application for its good performance in information classification. However, in order to achieve satisfactory classification standard, it is necessary to train the SVM with quantities of labeled datasets. In fact, this condition cannot be fully achieved as the acquisition of the labeled data is usually difficult or the payment is much expensive. In contrast, unlabeled data are abundant and easy to collect. Furthermore, relatively few labeled datasets lead to a frequent drawback, that is the over fitting to the training data with a consequent loss of generality. Thus, to deal with this problem, the semi-supervised support vector machine (S^3VM) learning method is proposed [1–3].

The semi-supervised support vector machine is the method utilizing both the labeled and unlabeled data for learning. The main goal of the S^3VM is to employ the large collection of unlabeled data together with a limited labeled data to improve the classification accuracy. Because of its elegant properties with unique global optimal solution and avoiding the disaster of dimensionality, lots of scholars have marched for this area and applied the S^3VM to many fields, such as text classification [4], the multi-class human

✉ En Wang
enwang19@xsyu.edu.cn

✉ Zi-Yang Wang
wangziyang1@buaa.edu.cn

Qing Wu
wuqing@xupt.edu.cn

[1] School of Marxism, Xi'an Shiyou University, Xi'an 710065, China

[2] Research Institute of Frontier Science, Beihang University, Beijing 100191, China

[3] School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

action recognition [5, 6], biomedical science [7, 8], graph reduction [9], image and video classification [10], and applications in industry and business [11, 12].

However, the main drawback of S^3VM is that the objective function is usually non-smooth. It needs to endure heavy burden in solving two quadratic programming problems with inversion matrix operation. Also, fast algorithm cannot be used, increasing the computing complexity. Some researchers have proposed several advanced methods to smooth the objection function. In 2005, the replacement of the non-smooth term $\max\{0, 1-|x|\}$ with $\exp(-3x^2)$ is given and the low density separation LDS-S^3VM [3] was proposed by Chapelle and Zien. But the approximation accuracy is not so high. In 2009, Liu et al. showed the polynomial function [13]$P(x) = \frac{1-x^2}{2} + \frac{1}{8}(1-x^2)^2 + \frac{1}{16}(1-x^2)^3 + \frac{5}{128}(1-x^2)^4 + \frac{7}{256}(1-x^2)^5$, $x \in [-\frac{1}{k}, \frac{1}{k}]$. However, the 10-order polynomial function is too complex and has too many calculations. Later, Yang et. al offered one new smoothing strategy of approximate function $\rho_\varepsilon(x) = \sqrt{x^2+\varepsilon} \approx |x|$ [14] based on robust difference of convex functions in 2013. This new smooth method applied the DC optimization algorithms for solving the S^4VMs, and didn't add new variables and constraints to the corresponding S^3VMs. It is a promising direction to facilitate the research of S^4VMs. Zhang et al. introduced their cubic spline function [15] $s(x,k) = \frac{k^2|x|^3}{3} - kx^2 - \frac{1}{3k} + 1$, $(|x| \le \frac{1}{k})$, and quintic spline function [16] $s(x,k) = -\frac{k^4|x|^5}{5} + \frac{1}{2}k^3x^4 - kx^2 - \frac{3}{10k} + 1$, $(|x| \le \frac{1}{k})$ in 2015. However, the above smooth techniques are not so satisfied.

Motivated by the works of [3, 13–16], a new research question is gradually arisen, whether there is any other smooth technology, improving accuracy and decreasing calculation scale. In this paper, a new class of Bézier smooth functions is applied. Employing the smooth Bézier function $B_n(x)$ to approximate the non-smooth term $\max\{0, 1-|t|\}$, a novel class of Bézier smooth semi-supervised support vector machines (BS^4VMs) is derived. The new programming possesses the following attractive advantages: firstly, the fast gradient algorithm can be used to solve the BS^4VMs as the objective function becomes smooth and differentiable. Much calculation time can be saved. Secondly, a new class of smooth functions is proposed. The optimal smooth function can be selected for different scale datasets. Lastly and more importantly, convergence analysis and experimental comparisons verify that BS^4VMs are superior to the given models in classification capability and efficiency.

In order to make the expression more clear, the definition of each variable involved in equations is listed in Table 1. For example, all vectors are column vectors, and $\nabla f(t)$ represents the gradient of the function.

Table 1 List of symbols

y	The label of dataset
l	The number of labeled data
u	The number of unlabeled data
x^i	The input matrix of data
w	The weight vector
b	The bias term
ξ	The vector of slack variables
C	The penalty parameters for labeled data
C^*	The penalty parameters for unlabeled data
$L(\cdot)$	The hinge loss function
$J(\cdot)$	The quadratic unconstrained programming
$B_n(\cdot)$	The n-order of Bézier function
p_i	The Bézier interpolation point
$\phi(\cdot)$	The Gaussian kernel function
χ_F^2	The Fredman statistic
CD	The critical difference
∇	The gradient of the function
∂	The partial derivative of the function

The rest of this paper is organized as follows. The preliminary background knowledge of S^3VM will be introduced in Sect. 2. Section 3 shows how the BS^4VMs can be derived. A fast quasi-Newton algorithm for solving the programming will be followed in Sect. 4. Then the nonlinear BS^4VMs and the convergence analysis of the model are listed in Sects. 5 and 6. The comparisons of the proposed algorithm with other advanced methods based on four kinds of datasets will be analyzed in Sect. 7. The discussion and conclusion will be followed in the last section.

2 Preliminary of semi-supervised support vector machine

The purpose of S^3VM for binary classification is to maximize the margin by using the labeled and unlabeled data. Considering one programming, the training data contain the l labeled points $\{(x^i, y_i)\}_{i=1}^{l}, y_i = \pm 1$ and the u unlabeled dataset $\{x^i\}_{i=l+1}^{l+u}$, where $x^i = (x_1^i, x_2^i, ..., x_m^i) \in \mathbb{R}^m$. For the linearly separable data, one optimal separating hyperplane with the largest distance for the S^3VM classifier should be explored.

Let $y \triangleq (y^l, y^{l+u})$ be a column vector, where $y^l = (y_1, y_2, ..., y_l)^{\mathrm{T}}$ is the known label, and $y^{l+u} = (y_{l+1}, y_{l+2}, ..., y_{l+u})^{\mathrm{T}}$ is the unknown label. The

vector $y_n = [y_{l+1}, ..., y_{l+n}]$ based on the largest margin is the pursuing goal. For the linear condition, the S^3VM can be described as

$$\begin{aligned} J(w) = \min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{j=l+1}^{l+u}\xi_i \\ s.t.\, y_i(w^{\mathrm{T}}x^i + b) \geq 1 - \xi_i, i = 1, ..., l \\ \left|w^{\mathrm{T}}x^j + b\right| \geq 1 - \xi_i, j = l+1, ..., l+u, \\ \xi = \{\xi_1, \xi_2, ..., \xi_n\} \geq 0 \end{aligned} \tag{1}$$

where C and C^*, the penalty parameters for both labeled and unlabeled data, are greater than zero. The programming (1) can be changed into the unconstrained form of

$$J(w) = \min_{w,b\in R^{n+l}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} L^2(y_i(w^{\mathrm{T}}x_i + b)) + C^*\sum_{i=l+1}^{l+u} L(\left|w^{\mathrm{T}}x_i + b\right|) \tag{2}$$

in which $L(t)$ is the hinge loss function and $L(t) = \max(0, 1-t)$[3].

3 Bézier smooth semi-supervised support vector for classification

3.1 Background knowledge about the Bézier function

Bézier curves were invented in 1968 by the French engineer Pierre Bézier for the initial purpose of designing automobile bodies [18]. For one series of interpolation points $P_0, P_1, \cdots P_{n-1}, P_n$ that need to be fitted, the intermediate points $P_1, \cdots P_{n-1}$ are used to specify the endpoint tangent vectors. Hence the Bézier curve passes through P_0 and P_n and approximates the other controlpoints, just like Fig. 1. To accomplish this goal, some kinds of weighting functions representing the influence of the control points at a given point of the Bézier curve are required. Arbitrary function satisfying the requirements is allowed, but in most cases the Bernstein polynomial is employed. A Bézier curve of degree n can be expressed as $B(t) = \sum_{i=0}^{n} C_i^n(t)P_i$, where P_i is the control point or anchor point.

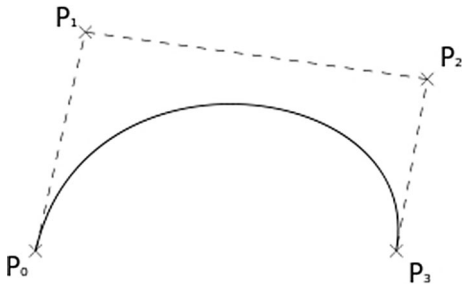

Fig. 1 Schematic diagram of the Bézier interpolation function

$C_i^n(t)$ means the Bernstein polynomial given by $C_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$, in which $i \in \{0, 1, ..., n\}$.

Many advantages for Bézier Curves have been noticed:

(1) They always passed through anchor points P_0 and P_n.
(2) They are always tangent to the lines of path $P_0 \rightarrow P_1$ and $P_{n-1} \rightarrow P_n$.
(3) They always lie within the convex hull consisting of the control points [19]. Owing to these good performances, the Bézier curves have been widely applied in computer graphic, such as technical illustration programs, CAD programs, trajectory guidance, and so forth [20–23].

For approximating the hinge loss function, the quadratic parameter Bézier function can be expressed as $\begin{cases} B_{2x}(t) = (2t-1)/k \\ B_{2y}(t) = (-2t^2 + 2t)/k \end{cases}$ in which $p_0 = (-\frac{1}{k}, 0), p_1 = (0, \frac{1}{k}), p_2 = (\frac{1}{k}, 0)$. Eliminating the parameter t, $y = B_2(x) = -\frac{1}{2k}(k^2x^2 - 1)$ will be given. Similarly, the cubic parameter Bézier function $\begin{cases} B_{3x}(t) = (2t^3 - 2t^2 + 2t - 1)/k \\ B_{3y}(t) = (-3t^2 + 3t)/k \end{cases}$ will be acquired by interpolating four points p_0, p_1, p_2, p_3, in which $p_0 = (-\frac{1}{k}, 0), p_1 = p_2 = (0, \frac{1}{k}), p_3 = (\frac{1}{k}, 0)$. From the general formula $B(t) = \sum_{i=0}^{n} C_i^n(t)P_i$. the n-order Bézier function $y = B_n(x)$ will be acquired.

Theorem 1 *Bézier curve $B_{n-1}(t)$ is $n-1$-order smooth at the points $x = \pm\frac{1}{k}$.*

Proof The proof is based on mathematical induction.

(i) $\forall x \in \Omega, B_2(x) = -\frac{1}{2}(k^2x^2 - 1)$ satisfies the following equalities at the points $x = \pm\frac{1}{k}$

$$\begin{cases} B_2(-\frac{1}{k}) = 0, & B_2(\frac{1}{k}) = 0, \\ B_2'(-\frac{1}{k}) = 1, & B_2'(\frac{1}{k}) = -1. \end{cases} \tag{3}$$

So, $B_2(x, k)$ is one-order smooth.

(ii) $B_3(x)$ satisfies the following equalities at the points $x = \pm\frac{1}{k}$,

$$\begin{cases} B_3(-\frac{1}{k}) = 0, & B_3(\frac{1}{k}) = 0, \\ B_3'(-\frac{1}{k}) = 1, & B_3'(\frac{1}{k}) = -1, \\ B_3''(-\frac{1}{k}) = 0, & B_3''(\frac{1}{k}) = 0. \end{cases} \tag{4}$$

Hence, $B_3(x)$ is twice-order smooth.

(iii) Let $B_{P_0P_1\ldots P_{n-1}}$ denote the Bézier curve determined by points $P_0, P_1, \ldots, P_{n-1}$. Based on

$$B(t) = B_{P_0\cdots P_{n-1}}(t) = (1-t)B_{P_0\cdots P_{n-2}}(t) + tB_{P_1\cdots P_{n-1}}(t), \tag{5}$$

according to the mathematical induction, $B_{n-1}(x)$ is $n-1$ order smooth can be proved.

3.2 Bézier smooth semi-supervised support vector for classification

From (2), the last term $C^*\sum_{i=l+1}^{l+u} L(|w^{\mathrm{T}}x_i + b|)$ is non-smooth and difficult to solve [4], making the formula (2) become a difficult-solving mixed-integer quadratic programming. Replacing this term with smooth function $y = B_n(x)$, a new class of Bézier smooth semi-supervised support vector machines (BS^{4}VMs) is derived, described in formula (6)

$$\begin{aligned}\min_{w,b}\varphi(w,b) = \min\frac{1}{2}w^2 + C\sum_{i=1}^{l}L^2(y_i(w^{\mathrm{T}}x_i+b)) \\ +C^*\sum_{i=l+1}^{l+u}B_n(w^{\mathrm{T}}x_i+b).\end{aligned} \tag{6}$$

In this paper, without loss of generality, 4-order Bézier interpolation function $y = B_4(x)$ is taken into consideration. The higher the order of Bézier function, the better the approximation. The approximation comparison of different smooth models can be seen in Fig. 2.

From Fig. 2, one can find that (1) the 4-order Bézier function performs best among 3-order Bézier function, exponent function, 10-order polynomial, the cubic spline function, and quintic spline function in approximating the hinge loss function. (2) 3-order Bézier function performs almost the same with 10-order polynomial, while the calculation complexity is much less than the latter.

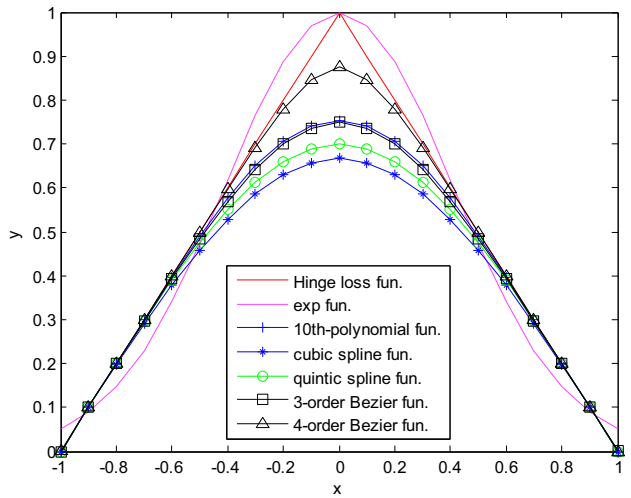

Fig. 2 Approximation comparison among the proposed models and the Bézier model with $k = 1$.

4 The nonlinear kernel for BS^{4}VM

For the nonlinear case, the kernel function $k(x^i, x^j) = \phi(x^i)^{\mathrm{T}}\phi(x^j)$ can be applied to map the original data into the high dimension Hilbert space. After this transforming, the linear program will be arrived. Let $\phi : R^m \to R^d (d > m)$ be the mapping function of the formula (1). The nonlinear kernel-based S^{3}VM can be shown as

$$\begin{aligned}J(w) = \min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{j=l+1}^{l+u}\xi_i \\ s.t.\, y_i(w^{\mathrm{T}}\phi(x^i)+b) \geq 1-\xi_i, i = 1,\ldots,l \\ \left|w^{\mathrm{T}}\phi(x^j)+b\right| \geq 1-\xi_i, j = l+1,\ldots,l+u. \\ \xi = \{\xi_1,\xi_2,\ldots,\xi_n\} \geq 0\end{aligned} \tag{7}$$

In this paper, the Gaussian kernel $k(x^i, x^j) = \exp(-\|x^i - x^j\|_2^2/2\sigma^2)$ is adopted and the kernel function $K = k(x^i, x^j)$ is positive semi-definite matrix [17]. For formula (2), the variable w will be replaced by $w = \sum_{i=1}^{m} u_i y_i x_i$, in which $u \in R^m$. The nonlinear S^{3}VM is achieved.

$$\begin{aligned}\min_{u,b}\varphi(u,b) = \min\frac{1}{2}u^2 + C\sum_{i=1}^{l}L^2(y_i(k(x_i,x_j)u_j+b)) \\ + C^*\sum_{i=l+1}^{l+u}L(\left|k(x_i,x_j)u_j+b\right|).\end{aligned} \tag{8}$$

Applying the n-order Bézier smooth function, the nonlinear BS^{4}VM model with kernel function is offered.

$$\begin{aligned}\min_{u,b}\varphi(u,b) = \min\frac{1}{2}u^2 + C\sum_{i=1}^{l}L^2(y_i(k(x_i,x_j)u_j+b)) \\ + C^*\sum_{i=l+1}^{l+u}B_n(k(x_i,x_j)u_j+b).\end{aligned} \tag{9}$$

The objective function (9) is $n-1$-order differentiable for any arbitrary kernel.

5 One fast quasi-Newton method for solving BS^{4}VM

In this section, the sub-LBFGS algorithm will be employed to solve semi-supervised problem (1) [2, 24, 25]. Differentiate (2) with the method of subgradient, the following will be given:

$$\partial J(w) = w + C\sum_{i=1}^{l}\beta_i y_i x_i + C^*\sum_{i=l+1}^{l+n}\beta_i y_i x_i, \tag{10}$$

where

$$\beta_i := \begin{cases} 1 \text{ if } i \in E, E := \{i : 1 - y_i w^{\mathrm{T}} x_i > 0\}, \\ \psi \text{ where} \psi \in (0,1), \text{ if } i \in M, M := \{i : 1 - y_i w^{\mathrm{T}} x_i = 0\}, \\ 0 \text{ if } i \in W, W := \{i : 1 - y_i w^{\mathrm{T}} x_i < 0\}, \end{cases}$$

E, M and W denote the sets of points which are in error, on the margin and well-classified, respectively. For a given direction p, it is required to find a subgradient g. Based on formula (10), Eq. (11) will be given:

$$\begin{aligned} \sup_{g\in\partial J(w_t)} g^{\mathrm{T}}p &= \sup_{\beta_i, i\in M_t}(w + C\sum_{i\in M_t}\beta_i y_i x_i + C^*\sum_{i\in M_t}\beta_i y_i x_i)^{\mathrm{T}}p \\ &= w^{\mathrm{T}}p + C\sum_{i\in M_t}\sup_{\beta_i\in[0,1]}\beta_i y_i x_i^{\mathrm{T}}p + C^*\sum_{i\in M_t}\sup_{\beta_i\in[0,1]}\beta_i y_i x_i^{\mathrm{T}}p. \end{aligned} \tag{11}$$

Now the S^{3}VM algorithm with sub-LBFGS optimization solving procedure can be offered (Algorithm 1).

In step 3 of Algorithm 1, a classifier is obtained by firstly running BS^{4}VM on the labeled examples alone. Steps 5–17 show the loop iteration process when solving the objective programming. Step 9 identifies pairs of unlabeled examples with temporary positive and negative labels such that switching these labels would decrease the value of the objective function.

6 Convergence analysis of the Bézier function and BS^{4}VM

This section will show the approximation precision of Bézier function to hinge loss function and the convergence of BS^{4}VM. In addition, the convergence condition holds in the nonlinear BS^{4}VM.

6.1 Approximation accuracy analysis of Bézier function

Theorem 2 *Let $x \in R, k > 0, L(x)$ stand for the hinge loss function, and $B_4(x,k)$ be the Bézier function with five interpolation points. There will be such results*

$$\begin{aligned} &0 \le B_4(x,k) \le L(|x|,k) \\ &0 \le L^2(|x|,k) - B_4^2(x,k) \le \frac{15}{64k^2} \end{aligned} \tag{12}$$

Proof (i) It is obvious that $L(|x|,k) - B_4(x,k)=0$ holds with $|x| > \frac{1}{k}$. For $x \in [-\frac{1}{k}, 0)$, $L(|x|,k)$ and $B_4(x,k)$ are monotonically increasing, and $L(|x|,k) - B_4(x,k) \ge L(|\frac{-1}{k}|,k) - B_4(\frac{-1}{k},k)=0$ is easy to obtain. For $x \in [0, \frac{1}{k}]$, $L(|x|,k)$ and $B_4(x,k)$ are monotonically decreasing, and there will be $L(|x|,k) - B_4(x,k) \ge L(|\frac{1}{k}|,k) - B_4(\frac{1}{k},k)=0$. So $0 \le B_4(x,k) \le L(|x|,k)$ is achieved.

(ii) $L^2(|x|,k) - B_4^2(x,k)=0$ holds with $|x| > \frac{1}{k}$. For $x \in [-\frac{1}{k}, 0)$, from (i), one can find $L(|x|,k)$ and $B_4(x,k)$ are monotonically increasing; therefore, $0 \le L(|x|,k) - B_4(x,k) \le L(0,k) - B_4(0,k) = \frac{1}{8k}$ is established. As is known, $L(|x|,k) + B_4(x,k) \le L(0,k) + B_4(0,k) = \frac{15}{8k}$, $L^2(|x|,k) - B_4^2(x,k) \le \frac{1}{8k} \cdot \frac{15}{8k} = \frac{15}{64k^2}$ will be derived. In short, $0 \le L^2(|x|,k) - B_4^2(x,k) \le \frac{15}{64k^2}$ is proved.

6.2 Convergence analysis of the BS^{4}VM

Theorem 3 *Let $A \in R^{m\times n}, b \in R^{m\times 1}$, and define two real functions $g(x)$ and $f(x,k)$ as follows*:

$$\begin{aligned} g(x) &= \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|L(|Ax+b|)\|_2^2 + \frac{1}{2}\|L(|Ax+b|)\|, \\ f(x,k) &= \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|B_4(Ax+b,k)\|_2^2 + \frac{1}{2}\|B_4(Ax+b,k)\|. \end{aligned} \tag{13}$$

The following results can be achieved:

(1) $\forall k > 0$, there will be $\|x_k^* - x^*\| \le \frac{15}{128k^2}$

(2) $\lim_{k\to\infty}\|x_k^* - x^*\| = 0$

Proof (i) Applying the first-order optimization condition and convex property of $g(x)$ and $f(x,k)$, formula (14) is attained,

$$\begin{aligned} g(x_k^*) - g(x^*) &\ge \nabla g(x^*)(x_k^* - x^*) + \frac{1}{2}\|x_k^* - x^*\|_2^2 \\ &= \frac{1}{2}\|x_k^* - x^*\|_2^2, \\ f(x^*,k) - f(x_k^*,k) &\ge \nabla f(x^*)(x^* - x_k^*) + \frac{1}{2}\|x_k^* - x^*\|_2^2 \\ &= \frac{1}{2}\|x_k^* - x^*\|_2^2. \end{aligned} \tag{14}$$

Based on the formula (13) and the property of $B_4(x,k) \le h(x)$, formula (14) is acquired,

$$\begin{aligned}\left\|x_k^* - x^*\right\| &\le g(x_k^*) - g(x^*) + f(x^*,k) - f(x_k^*,k)\\ &= (f(x^*,k) - g(x^*)) - (f(x_k^*,k) - g(x_k^*))\\ &\le g(x^*) - f(x^*,k)\\ &= \frac{1}{2}\|L(Ax+b)\|_2^2 - \frac{1}{2}\|B_4(Ax+b,k)\|_2^2.\end{aligned} \tag{15}$$

According to Theorem 2, for $x \in [-\frac{1}{k}, \frac{1}{k}]$, $L^2(|x|,k) - B_4^2(|x|,k) \le L^2(0,k) - B_4^2(0,k) = \frac{15}{64k^2}$. So $\left\|x_k^* - x^*\right\| = \frac{1}{2}[L^2(|x|,k) - B_4^2(x;k)] \le \frac{15}{128k^2}$ holds.

(ii) As $\left\|x_k^* - x^*\right\| \le \frac{15}{128k^2}$, it is easy to draw the conclusion of $\lim_{k\to\infty}\left\|x_k^* - x^*\right\| = 0$. Theorem 3 is proved.

7 The experiments and comparisons

This section will evaluate the performance, effectiveness and complexity of the proposed BS^{4}VMs. It will be surveyed from two dimensions. The longitudinal dimension means the comparison BS^{4}VMs with other three smooth models, LDS^{4}VM (S^{3}VM with low density separation) [3], CS^{4}VM (S^{3}VM with cubic spline function) [15], and the QS^{4}VM (S^{3}VM with quintic spline function) [16]. The horizontal dimension stands for the comparison of BS^4-VMs within different orders. This part lists three kinds of BS^{4}VMs, BS^{4}VM-I (S^{3}VM with 2-order Bézier function), BS^{4}VM-II (S^{3}VM with 3-order Bézier function), and BS^{4}VM-III (S^{3}VM with 4-order Bézier function). Experiments are carried on four kinds of datasets, the artificial datasets, UCI dataset,[1] USPS dataset, and large-scale NDC dataset. These four kinds of datasets are of significant difference. Subsection 7.1 shows the experiment on small-size artificial dataset named "checkboard." It is produced generated by two dimensions of uniformly distributing the regions to points. The "checkboard" belongs to one kind of data with nonlinear separable. In subsection 7.2, UCI datasets are the real-world datasets. They are generated from some statistical departments, electronic sensors, and reports. Some datasets are multi-classes and irregular data. Preprocessing is required. They have different data size. In subsection 7.3, handwritten symbol consists of 16*16 grayscale pixels of handwritten digits from '0' to '9'. These data are from the USPS Company and belong to the digital pattern recognition of real world. The last kind of dataset is NDC, namely, normally distributed clusters, generated by the NDC algorithm. The algorithm generates a series of random centers for multivariate normal distributions. Randomly generate a fraction for this center and a separating plane. Based on the plane, some classes for centers will be chosen. Then the points are randomly generated from the distributions. The size can be changed according to the experimenter. For the test of large-scale dataset, the NDC is a good choice.

Because of the too high complexity of the 10-order polynomial function in [13], the calculation time exceeds the acceptable range. Thus, this section ignores algorithm [13] in comparison. As the parameters C and C^* are not sensitive to the accuracy of classification, $C = C^*$ is set varying from 10^{-2} to 10^2. All classifiers are implemented on PC of Windows 10 with 64 bit operation system, Intel I7 processor (1.6 GHZ) and 16 GB RAM. The codes of models are written in MATLAB R2009a.

Experiments are set up according to the following rules: the ratio of the labeled points m varies from 5 to 65%, and the rest is the unlabeled points, similar to the unlabeled data ratio evolving from 20 to 80% in [26]. The labeled ratio is set according to the missing label scenarios in real world. 5% labeled ratio means the majority of data labels are missing. This is a picky condition to detect a good classifier. On the other hand, if the labeled ratio is more than 70%, too many labels means the gap between semi-supervised SVM and full-supervised SVM is quite small. Therefore, the labeled ratio is set from 5 to 65% with the interval of 20%. The labeled data are used for training the LDS^{4}VM, CS^{4}VM, QS^{4}VM, BS^{4}VM, and then predicting the unlabeled points. Before simulation, all databases are normalized and the two classes of label are divided into classes of -1 and $+1$. Each experiment is carried on with tenfold cross-validation.

7.1 Experiment based on artificial dataset

The first experiment is designed to demonstrate the effectiveness of BS^{4}VM through the artificial nonlinear "tried and true" checkboard dataset [27]. The checkboard dataset is generated by two dimensions of uniformly distributing the regions to points and labeling two classes "White" and "Black." Each dimension has 100 points, and thus the checkboard dataset has 10,000 samples to train and test the algorithms, just as Fig. 3 shows. The comparison result can be seen in Table 2.

Table 2 demonstrates that (1) with the increase in the labeled ratio, the classification accuracy climbs on the whole. (2) The higher the order of the spline polynomial, the better the classification accuracy. (3) The checkboard dataset is not suitable for too few labeled samples, as the experiment result is not so satisfactory with labeled ratio equal to 5%. Lastly, the comparison in Table 2 shows the BS^{4}VM performs the best classification accuracy.

[1] The UCI dataset can be available at https://archive.ics.uci.edu/ml/datasets.php and https://cs.nyu.edu/~roweis/data.html.

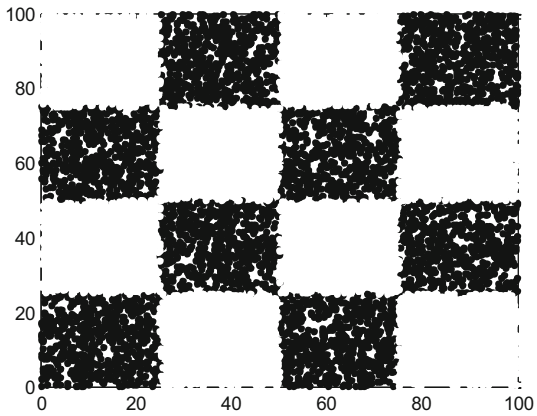

Fig. 3 Figure of the checkerboard dataset

7.2 Results on UCI datasets

In this subsection, eight real-world UCI datasets[2] are chosen to test the four classification algorithms. This collection of databases was created in 1987 and has been widely used by the machine learning community for the empirical analysis. It provides various datasets from many areas in reality life, such as disease diagnosis, manufacturing, business, and so on. The calculating results are given in Table 3.

Table 3 illustrates the detailed comparisons of the proposed model with other three models in eight different datasets. From Table 3, one can find that with the increase in the labeled ratio, all the algrorithms show better classification accuracy. For Clean dataset with labeled ratio varying from 25 to 65%, the experimental result by BS^{4}VM (accuracy 68.35%, 71.28%, 75.45%) outperforms other three algorithms, LDS^{4}VM (66.95%, 65.94%, 72.90%), CS^{4}VM (66.11%, 66.13%, 70.66%), and QS^{4}VM (66.53%, 67.94%, 71.41%). This conclusion holds for datasets Lympho, Bupa, Tumor, WDBC, and Adult as well in most scenarios. For datasets Balance, German, the advantages of classification accuracy go up and down, and BS^{4}VM performs a litter better than other three methods.

For the purpose of describing the dynamic process of test accuracy for each dataset with various labeled ratios, Fig. 4 is given. It presents the overall trend of these algorithms. It claims all the lines have the trend of climbing with the labeled ratio increasing. Taking Data (4) for example, the red line stands for the proposed BS^{4}VM method, the blue and black lines mean QS^{4}VM, CS^{4}VM, and the purple line denotes LDS^{4}VM. For the labeled ratio 5%, the accuracy of CS^{4}VM is better than BS^{4}VM. But with the ascension of ratio, the red line is always above the other three lines, claiming the BS^{4}VM performs the best with high labeled ratio.

[2] http://archive.ics.uci.edu/ml/index.php.

To further analyze the statistical accuracy more clearly, the average ranks of all the classifiers are computed and listed in Table 4 and Fig. 5. Table 4 indicates the average ranks of eight datasets. This rank order is calculated by average value of each algorithm with different labeled ratios. The smaller the number of rank, the higher the simulation accuracy. From the last row of Table 4, one can notice that BS^{4}VM ranks in the first place for eight datasets, whereas the others stand on second, third and fourth places.

In order to verify the advantage of proposed algorithm BS^{4}VM, the Friedman statistical method is employed. Fredman statistic is distributed based on χ_F^2 with $k-1$ degree of freedom, where k means the counts of algorithms and N stands for the counts of datasets.

For the above experiment on UCI datasets, under the null hypothesis that all the algorithms are equivalent, Fredman statistic can be calculated as [28]

$$\begin{aligned}\chi_F^2 &= \frac{12N}{k(k+1)}\left[\sum_{i=1}^{4} R_i^2 - \frac{k(k+1)^2}{4}\right] \\ &= \frac{12\times 8}{4\times 5}\left[3.2188^2 + 2.7031^2 + 2.8281^2 + 1.25^2 \right. \\ &\quad \left. - \frac{4\times 5^2}{4}\right] = 10.6954\end{aligned}$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} = \frac{7\times 10.6954}{8\times 3 - 10.6954} = 5.6264$$

For four algorithms and eight datasets, F_F is distributed with $(k-1)=3$ and $(k-1)(N-1)=21$ degrees of freedom. The critical or threshold value of $F(3,21)$ for significance level $\alpha = 0.05$ is 3.072. Obviously, $F_F = 5.6264 > F(3,21) = 3.072$, thus the null hypothesis will be rejected, and these four algorithms having significant differences can be surely verified.

After the null hypothesis is rejected, the Nemenyi test can be proceed when all classifiers are compared to each other [28]. The performance of two classifiers is of significant difference if the corresponding average ranks differ by at least the critical difference $CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}}$. For the UCI experiment, $CD = 2.291\sqrt{\frac{4\times 5}{6\times 8}} = 1.4788$ at $\alpha = 0.1$. As the average rank difference between LDS^{4}VM and BS^{4}VM (3.2188–1.25 = 1.9688) is bigger than critical difference 1.4788, the performance of BS^{4}VM is significantly better than that of LDS^{4}VM. Similarly, the performance of BS^{4}VM is quite superior than that of QS^{4}VM ($2.8281 - 1.25 = 1.5781 > 1.4788$). Due to $2.7031 - 1.25 = 1.4531 < 1.4788$, this Nemenyi test cannot detect the significant difference between CS^{4}VM and BS^{4}VM.

Figure 5 visually presents the accuracy ranks of experiment results with different labeled ratios. One can find that

Table 2 Test accuracy on checkboard dataset with different labeled ratio (the bold part is the best result)

Dataset	Labeled ratio(%)	LDS^{4}VM Corr.(%)	CS^{4}VM Corr.(%)	QS^{4}VM Corr.(%)	BS^{4}VM Corr.(%)
Check_board	$m = 5$	50.72	48.69	55.67	**57.16**
	$m = 25$	56.29	53.86	57.43	**59.47**
	$m = 45$	86.18	86.00	86.73	**87.45**
	$m = 65$	86.00	86.57	86.57	**87.71**

Table 3 Tenfold cross-validation results of the average correction with different ratios of labeled points on eight public datasets for the four algorithms (the bold part is the best result)

Dataset	Labeled ratio(%)	LDS^{4}VM Corr.(%)	CS^{4}VM Corr.(%)	QS^{4}VM Corr.(%)	BS^{4}VM Corr.(%)
Lympho	$m = 5$	78.19	82.62	81.91	**85.82**
(148*18)	$m = 25$	81.31	79.95	82.66	**87.39**
	$m = 45$	84.45	88.72	86.59	**89.02**
	$m = 65$	91.83	90.38	90.87	**92.31**
Bupa	$m = 5$	94.74	95.79	95.35	**96.14**
(330*13)	$m = 25$	94.67	95.67	94.56	**96.00**
	$m = 45$	95.15	95.61	95.30	**96.36**
	$m = 65$	96.19	97.62	97.38	**99.05**
Tumor	$m = 5$	**59.56**	57.43	59.23	59.48
(339*17)	$m = 25$	62.45	62.25	64.31	**65.10**
	$m = 45$	65.24	66.31	66.04	**69.92**
	$m = 65$	65.55	67.44	66.81	**75.63**
Clean	$m = 5$	58.11	61.75	56.79	**62.91**
(476*166)	$m = 25$	66.95	66.11	66.53	**68.35**
	$m = 45$	65.94	66.13	67.94	**71.28**
	$m = 65$	72.90	70.66	71.41	**75.45**
Balance	$m = 5$	79.46	88.68	**91.50**	90.57
(625*4)	$m = 25$	91.90	**94.46**	92.86	93.60
	$m = 45$	89.53	92.81	91.50	**94.99**
	$m = 65$	91.21	95.32	94.41	**96.69**
German	$m = 5$	68.95	68.95	**69.05**	68.95
(1000*24)	$m = 25$	66.80	**68.27**	68.00	66.93
	$m = 45$	66.91	66.91	65.27	**67.09**
	$m = 65$	70.00	72.00	70.00	**72.57**
WDBC	$m = 5$	88.68	88.54	83.50	**90.39**
(569*30)	$m = 25$	93.33	92.92	93.03	**93.91**
	$m = 45$	94.00	94.49	94.33	**94.57**
	$m = 65$	92.88	92.38	92.50	**94.00**
Adult	$m = 5$	73.79	77.89	75.95	**78.42**
(1000*14)	$m = 25$	79.12	79.57	79.73	**79.93**
	$m = 45$	79.68	80.68	78.73	79.64
	$m = 65$	79.43	79.43	79.43	**80.29**

the advantage of BS^{4}VM varies. But from a statistical point of view, the BS^{4}VM performs best, just as Table 4 shows. The proposed algorithm shows satisfactory performance from Fig. 5b–d for most cases. This reminds us that, for different machine learning algorithms, the statistical results of quantities of datasets are more precision and credible, rather not one specific calculation.

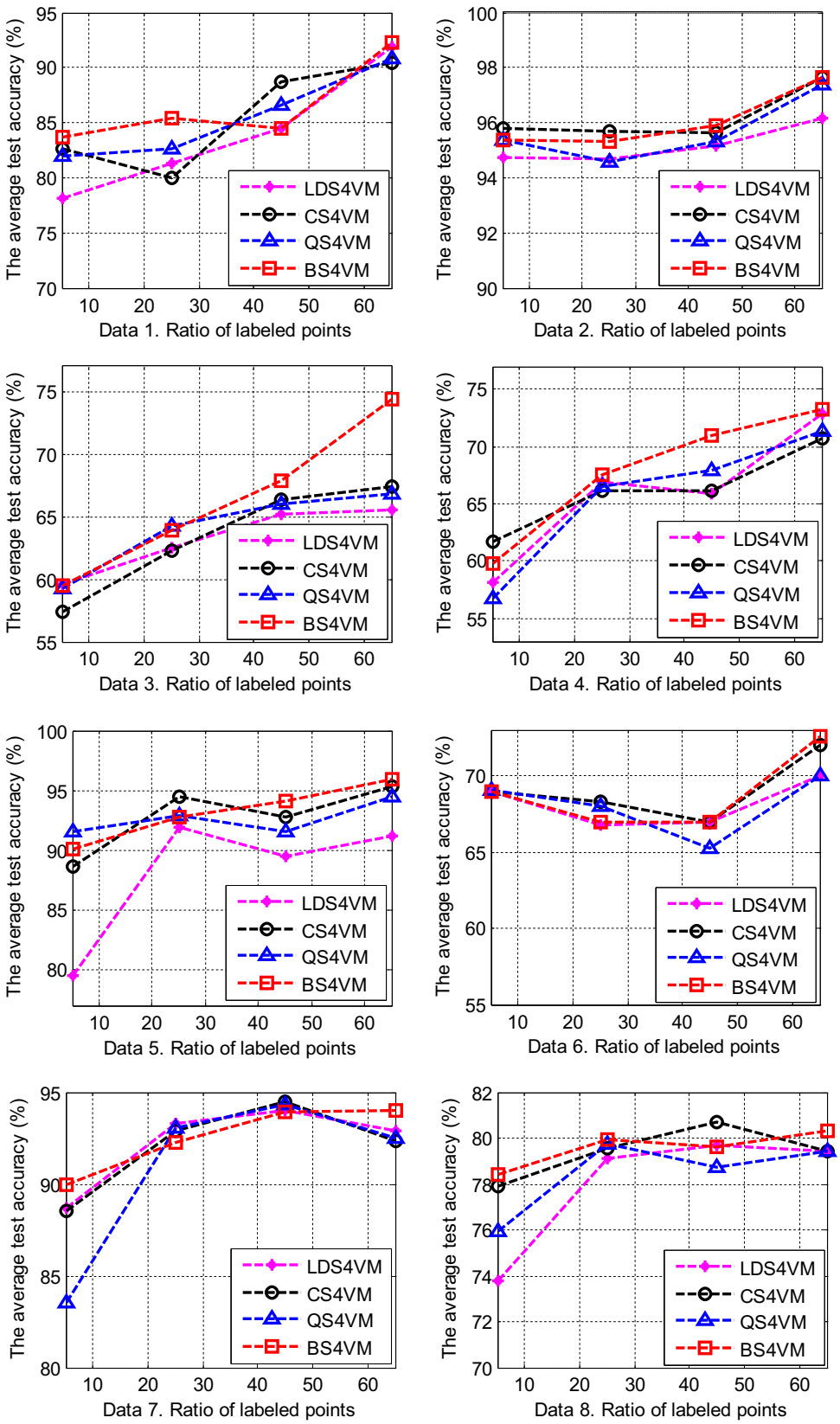

Fig. 4 The accuracy comparison of the LDS^{4}VM, CS^{4}VM, QS^{4}VM, BS^{4}VM on eight publicly available datasets, with 5%, 25%, 45%, and 65% as labeled data: (1) Lympho, (2) Bupa, (3) Tumor, (4) Clean, (5) Balance, (6) German, (7) WDBC, (8) Adult

Table 4 Accuracy average ranks of LDS^{4}VM,CS^{4}VM, QS^{4}VM, BS^{4}VM with linear kernel

Dataset	LDS^{4}VM	CS^{4}VM	QS^{4}VM	BS^{4}VM
Lympho	3.25	3.25	2.5	1
Bupa	3.75	2	3.25	1
Tumor	3	3	2.75	1.25
Clean	2.75	3.25	3	1
Balance	4	2	2.5	1.5
German	3.25	2.625	2.625	1.5
WDBC	2.5	3.25	3.25	1
Adult	3.25	2.25	2.75	1.75
Average rank	3.2188	2.7031	2.8281	1.25

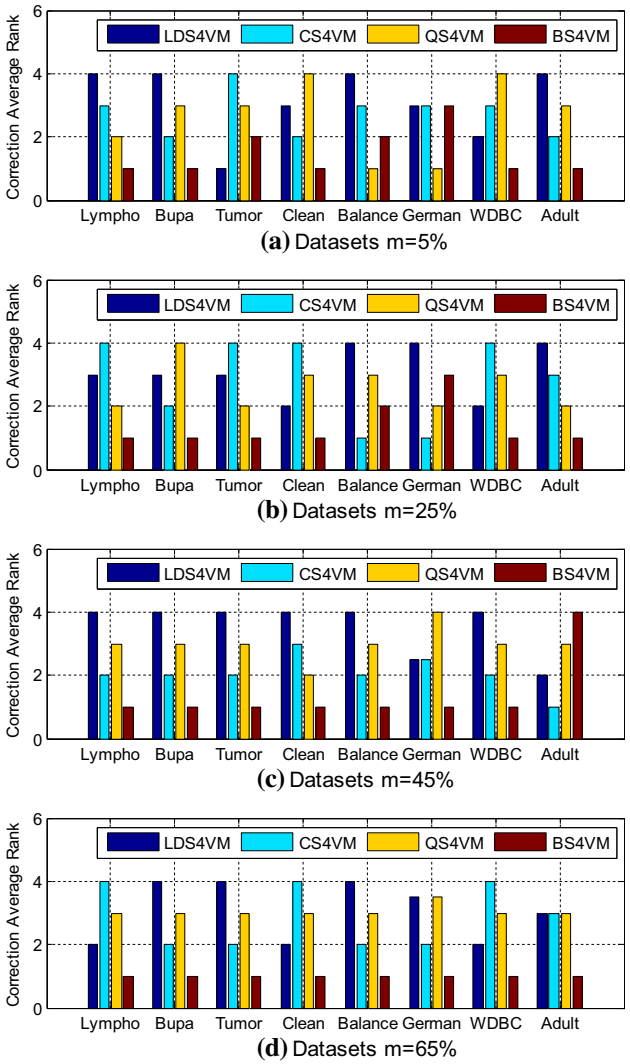

Fig. 5 Correction average ranks of LDS^{4}VM, CS^{4}VM, QS^{4}VM, BS^{4}VM in each dataset with different labeled ratios

Fig. 6 Ten number symbols of the USPS database

7.3 Results on handwritten symbol recognition

In this section, USPS handwritten datasets[3] will be investigated to show the impact of the number of labeled data on the classification accuracy. The handwritten database consists of grayscale images of handwritten digits from '0' to '9', as shown in Fig. 6.

The comparison of four pairwise digits '0' versus '8', '2' versus '4', '1' versus '7', and '3' versus '6' is given, respectively. The calculation results of accuracy and dynamic process can be seen in Table 5 and Fig. 7. From Table 5 and Fig. 7, the classification accuracies of two pairs '0′ versus '8′ and '1′ versus '7′ arrive at more than 80%, even almost 99%, while the classification accuracies of pairs '2' versus '4' and '3' versus '6' are less than 80%, even smaller than 52%. Thus, the generalization ability of S^{3}VM varies. The suitable dataset should be considered if one plans to carry out the identification process.

Table 6 and Fig. 8 express the accuracy ranks of each dataset with various labeled percentages. Table 6 proves that the BS^{4}VM ranks in the first place; meanwhile, the other three algorithms perform similarly. Figure 8 shows the accuracy rank of each calculation. Taking Fig. 8d for example, when the labeled data are more than 50%, the proposed learning algorithm gets well trained and shows satisfactory precision.

The Friedman statistical method can also be applied on USPS dataset to compare these algorithms from a quantitative perspective. For the four algorithms and four datasets,

$$\begin{aligned}\chi_F^2 &= \frac{12N}{k(k+1)}\Big[\sum_{i=1}^{4} R_i^2 - \frac{k(k+1)^2}{4}\Big] \\ &= \frac{12\times 4}{4\times 5}\big[2.65625^2 + 2.8125^2 + 2.75^2 + 1.78125^2 \\ &\quad - \frac{4\times 5^2}{4}\big] = 10.1203\end{aligned}$$

[3] The USPS datasets are available at http://www.cs.nyu.edu/*roweis/data.html.

Table 5 Tenfold cross-validation results of the average correction and the number of labeled points on USPS database for four algorithms (the bold part is the best result)

Dataset	Label ratio(%)	LDS^{4}VM Corr.(%)	CS^{4}VM Corr.(%)	QS^{4}VM Corr.(%)	BS^{4}VM Corr.(%)
'0' versus'8'	$m = 5$	86.45	**87.85**	85.96	80.78
	$m = 25$	88.32	88.46	**89.55**	88.80
	$m = 45$	**89.74**	89.37	89.27	87.69
	$m = 65$	92.69	93.42	93.27	**94.15**
'2' versus '4'	$m = 5$	55.10	58.25	56.09	**61.78**
	$m = 25$	**69.54**	67.22	67.05	**69.54**
	$m = 45$	69.71	67.00	68.69	**70.72**
	$m = 65$	75.18	75.18	75.89	**77.30**
'1' versus '7'	$m = 5$	**97.88**	**97.88**	**97.88**	**97.88**
	$m = 25$	97.91	97.84	97.69	**98.21**
	$m = 45$	98.27	98.07	**98.37**	**98.37**
	$m = 65$	98.57	98.09	98.25	**98.73**
'3' versus'6'	$m = 5$	51.04	53.58	53.13	**54.03**
	$m = 25$	**61.36**	58.05	59.75	56.44
	$m = 45$	53.74	53.99	54.64	**58.25**
	$m = 65$	62.50	63.51	63.31	**64.52**

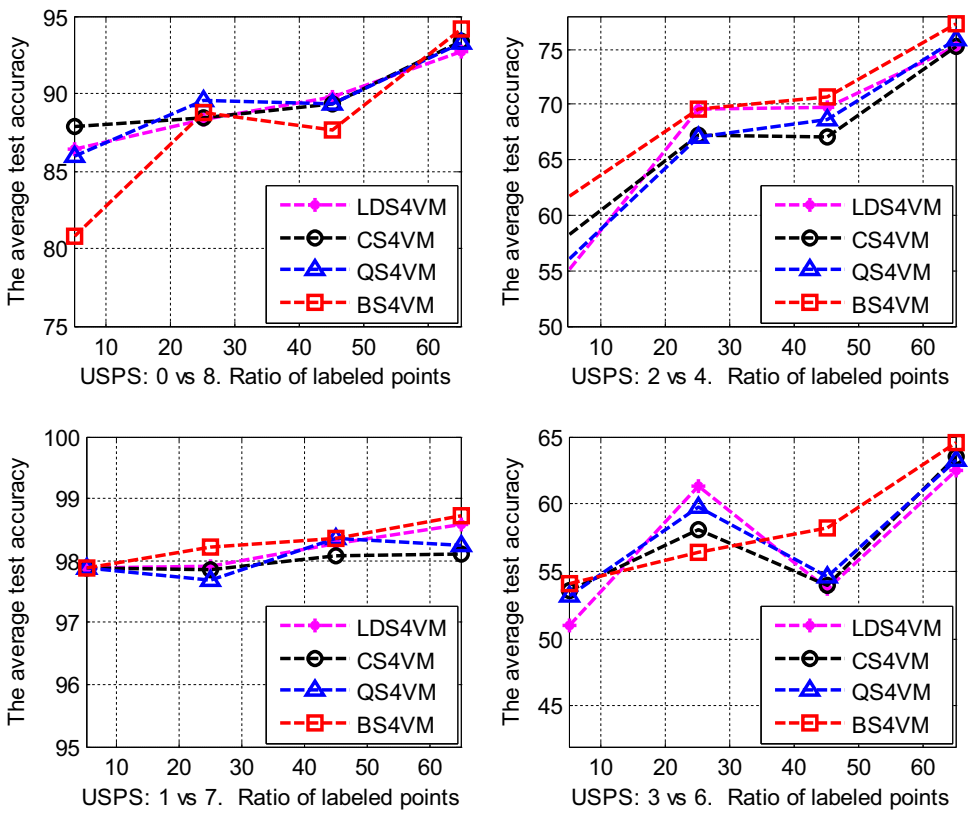

Fig. 7 The average test accuracy of LDS^{4}VM, CS^{4}VM, QS^{4}VM, BS^{4}VM on USPS dataset with various labeled ratios

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} = \frac{3 \times 10.1203}{4 \times 3 - 10.1203} = 16.1521$$

F_F is distributed with $(k-1) = 3$ and $(k-1)(N-1) = 9$ degrees of freedom. The threshold value of $F(3,9)$ for significance level $\alpha = 0.05$ is 3.863. Obviously, $F_F = 16.1521 > F(3,9) = 3.863$, thus the null hypothesis

Table 6 Average ranks of five algorithms with linear kernel on USPS accuracy values

Dataset	LDS^{4}VM	CS^{4}VM	QS^{4}VM	BS^{4}VM
0 versus 8	2.75	2	2.5	2.75
2 versus 4	2.75	3.125	3	1.125
1 versus 7	2.375	3.375	2.75	1.5
3 versus 6	2.75	2.75	2.75	1.75
Average rank	2.65625	2.8125	2.75	1.78125

will be rejected, and the hypothesis that four algorithms are of significant difference is proved. This means the generalization and robustness of BS^{4}VM are promising.

7.4 Results on large-scale NDC dataset for nonlinear Gaussian kernel

In the last subsection, further to verify which algorithm performs best on both accuracy and calculating time among BS^{4}VMs, experiments based on the NDC dataset[4] for nonlinear Gaussian kernel are carried out. The NDC dataset is designed with large-scale attributes or with large samples to test the robustness of the new algorithms. The NDC dataset is a temporal higher-order network dataset, which means a sequence of time-stamped simplices where each simplex is a set of nodes. As in the real world, large-scale datasets are more commonly classified, the test accuracy and calculation time should be considered as well.

Table 7 and Fig. 9 show the performances of three kinds of BS^{4}VMs, namely BS^{4}VM-I, BS^{4}VM-II, and BS^{4}VM-III with different orders of Bézier function. One can notice that (1) the BS^{4}VMs classify the NDC datasets very well, and most of the results are more than 96%. (2) With the climbs of labeled ratio and attributions of NDC1 $\sim$ NDC5, the computing time increases quickly. However, with the rise of samples of NDC6 $\sim$ NDC10, the calculating time doesn't go up dramatically. (3) Because these three algorithms are belong to the same kind of smooth technique, the accuracy differences are quite small. But the accuracy of BS^{4}VM-III stands the first place for most cases. Meanwhile, the computing time of BS^{4}VM-I and BS^{4}VM-II line up top two on account of the higher complexity of BS^{4}VM-III.

In order to clarify the comparison results, Table 8 lists the average ranks of BS^{4}VM-I, BS^{4}VM-II and BS^{4}VM-III with Gaussian kernel on accuracy and calculation time for NDC. From the statistics, the accuracy average rank of BS^{4}VM-III is 1.8875, smaller than other two number, indicating this method is more superior. The consuming

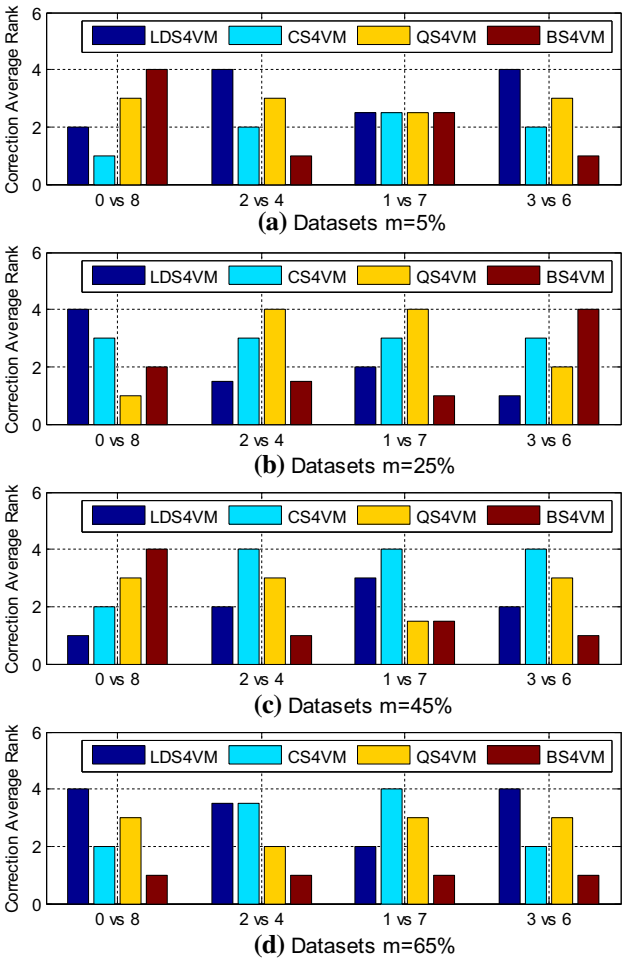

Fig. 8 Correction average ranks of LDS^{4}VM, CS^{4}VM, QS^{4}VM, BS^{4}VM in each dataset with different labeled ratios

time ranks of BS^{4}VM-I and BS^{4}VM-II are equal, revealing the computing complexity are the same even though BS^4-VM-II has higher order of Bézier function.

For the purpose of verifying whether the performances of the three algorithms have significant difference, the Friedman statistical method is utilized. For this experiment with three methods and ten datasets, statistical results χ_F^2 and F_F will be

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{i=1}^{5} R_i^2 - \frac{k(k+1)^2}{4}\right] = \frac{12\times 10}{3\times 4}\left[2.175^2 + 1.95^2 + 1.8875^2 - \frac{3\times 4^2}{4}\right] = 0.958$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} = \frac{9\times 0.958}{10\times 3 - 0.958} = 0.4528$$

The critical value of $F(2,18)$ for significance level $\alpha = 0.05$ is 3.55. Visibly, $F_F = 0.4528 < F(2,18) = 3.555$, and thus these three algorithms have no significant differences from the quantification method is verified. It is suggested

[4] http://www.cs.cornell.edu/$\sim$arb/data/NDC-classes/.

Table 7 The test correction and calculation time comparisons for Gaussian kernel (the bold part is the best result)

Dataset	Label ratio(%)	BS^{4}VM-I		BS^{4}VM-II		BS^{4}VM-III	
		Corr.(%)	Time(s)	Corr. %)	Time(s)	Corr.(%)	Time(s)
NDC1	$m = 5$	96.21	0.26	96.36	0.26	**96.38**	**0.23**
5,000*500	$m = 25$	98.27	0.83	**98.32**	**0.78**	**98.32**	0.83
	$m = 45$	98.76	1.67	**98.80**	1.64	**98.80**	**1.61**
	$m = 65$	**98.11**	2.50	**98.11**	**2.31**	**98.11**	2.39
NDC2	$m = 5$	**96.69**	**0.90**	96.65	1.05	96.67	1.14
5,000*1,000	$m = 25$	98.24	2.49	**98.27**	2.48	98.24	**2.74**
	$m = 45$	**98.84**	4.28	**98.84**	3.54	**98.84**	**4.11**
	$m = 65$	98.80	5.42	**98.86**	**5.21**	98.80	5.61
NDC3	$m = 5$	78.17	13.57	78.15	**11.84**	**78.19**	12.28
5,000*3,000	$m = 25$	94.05	24.40	**94.08**	**21.82**	94.00	22.34
	$m = 45$	96.80	36.96	**96.84**	**31.75**	**96.84**	33.55
	$m = 65$	96.51	45.40	**96.63**	**42.12**	96.57	45.74
NDC4	$m = 5$	85.98	31.02	**86.00**	27.22	85.96	**25.80**
5,000*4,000	$m = 25$	**96.11**	50.25	96.08	44.41	96.05	**42.23**
	$m = 45$	96.95	67.66	**97.05**	**61.22**	**97.05**	61.38
	$m = 65$	97.77	92.35	**97.83**	**79.95**	**97.83**	81.55
NDC5	$m = 5$	81.16	50.36	81.14	**44.14**	**81.18**	45.82
5,000*5,000	$m = 25$	**93.84**	87.22	93.79	77.62	93.81	**74.46**
	$m = 45$	97.02	116.47	97.05	102.58	**97.09**	**100.43**
	$m = 65$	97.54	154.77	97.43	130.92	**97.60**	**129.85**
NDC6	$m = 5$	98.44	**0.03**	**98.49**	0.04	98.48	0.04
10,000*100	$m = 25$	99.01	**0.13**	**99.01**	0.18	**99.01**	0.20
	$m = 45$	99.07	**0.26**	**99.09**	0.31	**99.09**	0.37
	$m = 65$	**99.06**	**0.36**	**99.06**	0.44	99.03	0.51
NDC7	$m = 5$	**99.01**	**0.08**	**99.01**	0.14	99.00	0.12
30,000*100	$m = 25$	**99.28**	**0.43**	99.27	0.51	**99.28**	0.61
	$m = 45$	99.34	**0.79**	99.34	1.09	**99.35**	1.08
	$m = 65$	99.49	**0.95**	99.49	1.15	**99.50**	1.27
NDC8	$m = 5$	**99.37**	**0.06**	99.35	0.10	99.36	0.09
40,000*100	$m = 25$	**99.55**	**0.35**	99.53	0.42	99.53	0.72
	$m = 45$	**99.59**	0.72	**99.59**	0.74	99.57	**1.17**
	$m = 65$	99.54	**1.06**	**99.55**	1.49	99.54	1.41
NDC9	$m = 5$	**99.36**	**0.11**	99.32	0.14	99.29	0.14
50,000*100	$m = 25$	99.51	**0.80**	**99.52**	0.83	**99.52**	1.13
	$m = 45$	99.57	1.35	99.57	1.64	**99.58**	**1.33**
	$m = 65$	99.59	**1.80**	99.59	2.13	**99.61**	2.04
NDC10	$m = 5$	**99.38**	**0.21**	99.34	0.27	99.33	0.28
100,000*100	$m = 25$	99.54	**1.09**	99.54	1.20	**99.55**	1.33
	$m = 45$	**99.53**	**2.08**	**99.53**	2.39	**99.53**	2.53
	$m = 65$	99.57	**3.12**	99.58	3.28	**99.59**	3.50

that if high accuracy is considered, higher order of BS^4-VMs should have the priority. However, if calculating time weighs a lot, the lower order of BS^{4}VMs should be chosen.

For the goal of visual expression, the diversities of classification correction and calculation time of each dataset with the variety of labeled ratio, the histogram Figs. 10 and 11 are given.

From Fig. 10c and d, the classification precision of BS^{4}VM-III lies in the forefront, when the threshold of labeled proportion is above 45%. However, this superior

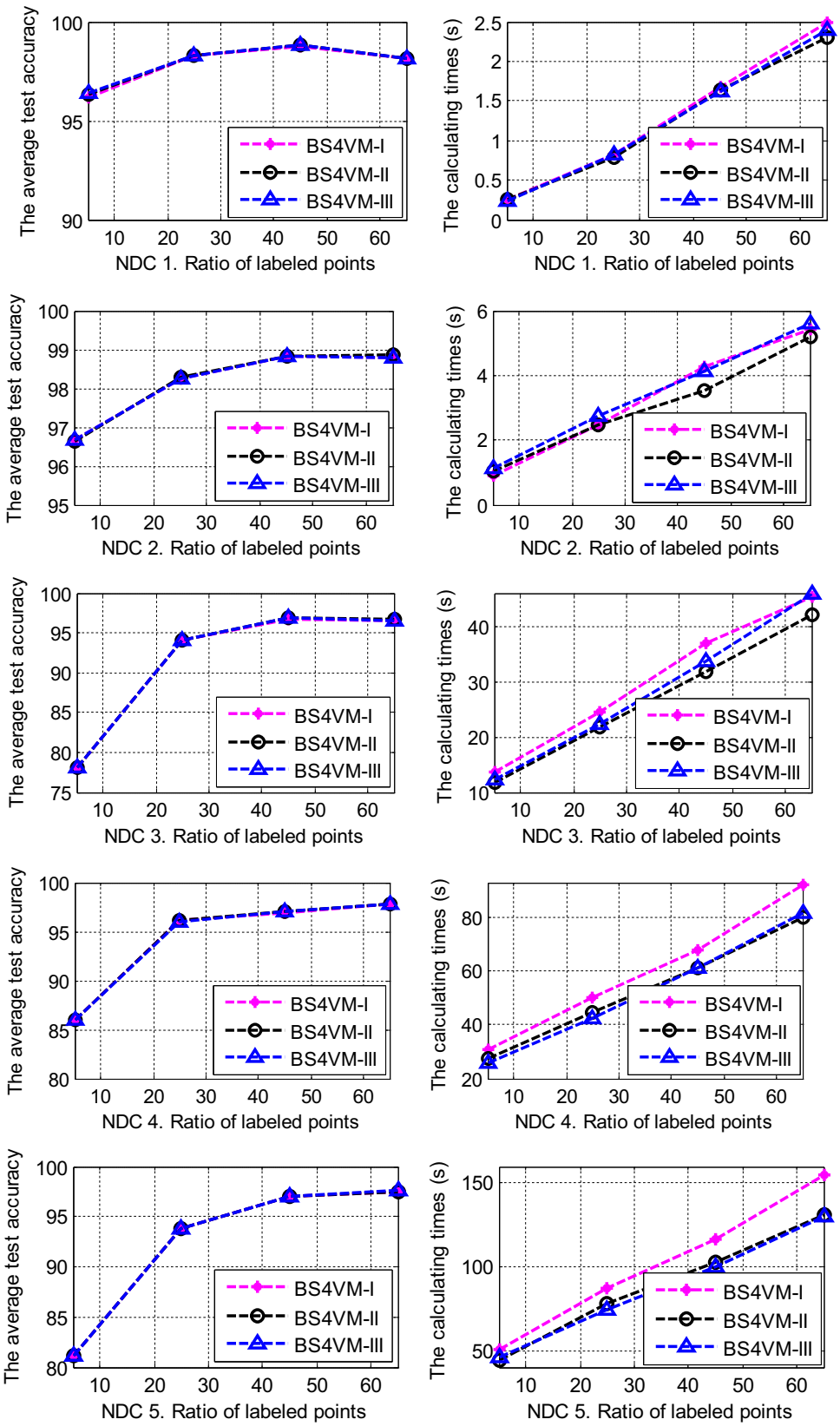

Fig. 9 The average test accuracy and calculation time of BS^{4}VM-I, BS4VM-II, and BS4VM-III on ten NDC dataset with various labeled ratio

Fig. 9 continued

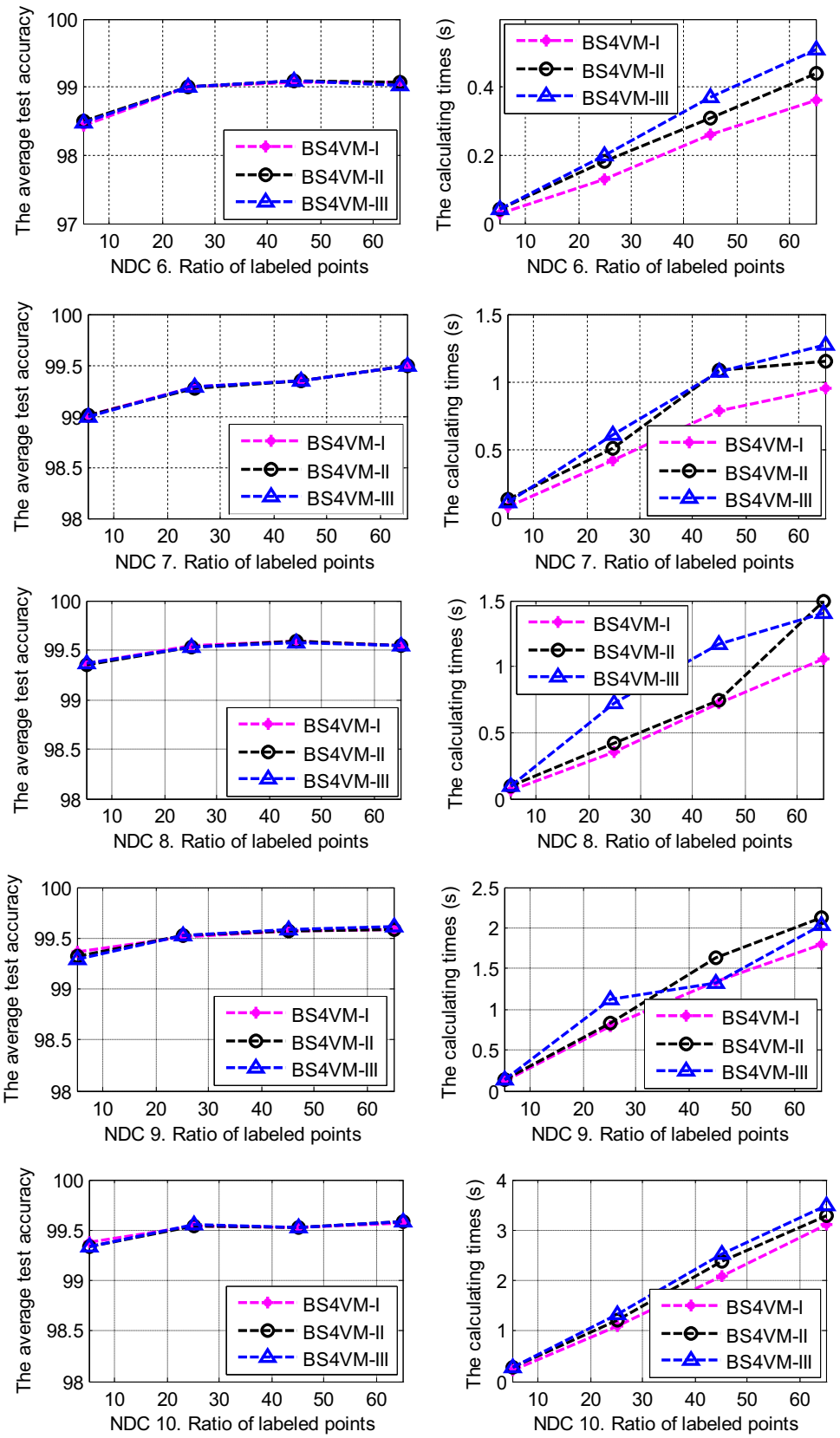

Table 8 Average ranks of BS^{4}VM-I, BS4VM-II and BS4VM-III with Gaussian kernel on NDC correction and time values

Dataset	BS^{4}VM-I		BS^{4}VM-II		BS^{4}VM-III	
	Corr.	Time	Corr.	Time	Corr.	Time
NDC1	2.75	2.75	1.75	1.625	1.5	1.625
NDC2	2	2.25	1.75	1.75	2.25	2
NDC3	2.5	2.75	1.625	1	1.875	2.25
NDC4	2.25	3	1.5	1.5	2.25	1.5
NDC5	2	3	2.75	1.75	1.25	1.25
NDC6	2.375	1	1.5	2.125	2.125	2.875
NDC7	2	1.125	2.375	2.25	1.75	2.25
NDC8	1.5	1.25	2	2.75	2.5	2
NDC9	2.25	1.25	2.125	2.625	1.625	2.125
NDC10	2.125	1	2.125	2	1.75	3
Average Rank	2.175	1.9375	1.95	1.9375	1.8875	2.0875

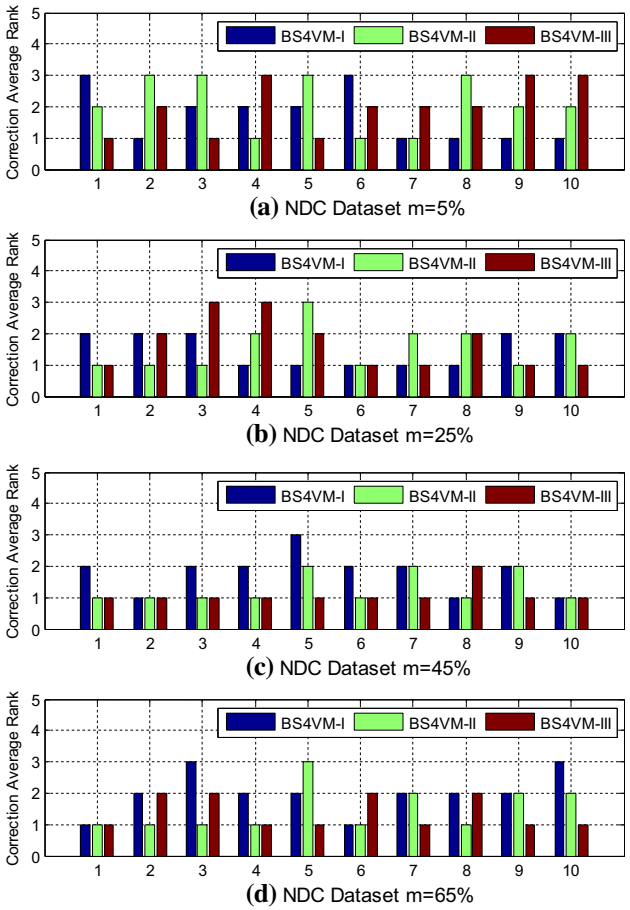

Fig. 10 Correction average ranks of BS^{4}VM-I, BS4VM-II and BS4VM-III in each dataset with different labeled ratios

performance is at the cost of complex calculation, just as Fig. 11c and d shows. From the ranks of calculation time,

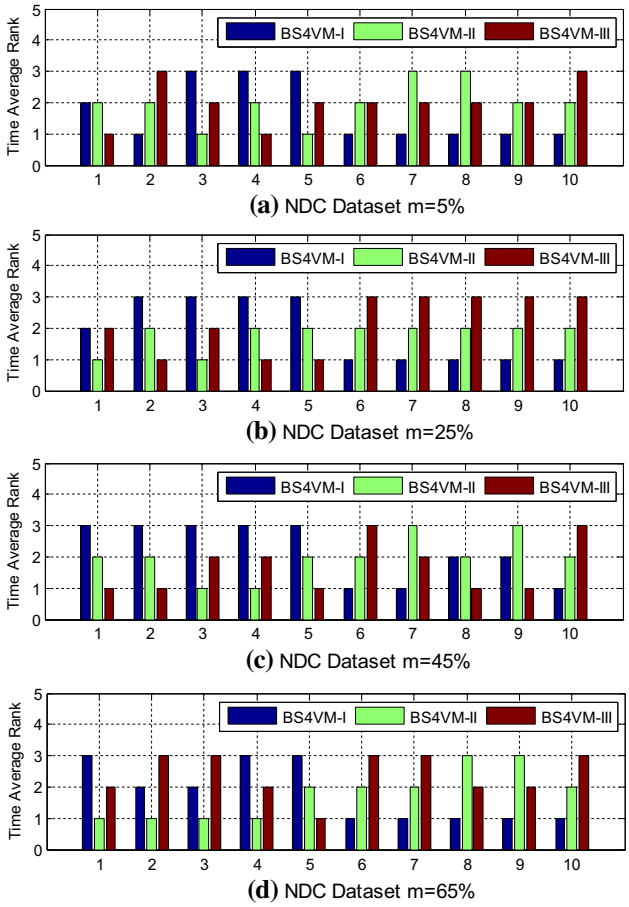

Fig. 11 Time average ranks of BS^{4}VM-I, BS4VM-II and BS4VM-III in each dataset with different labeled ratios

BS^{4}VM-I shows the perfect performance in Fig. 11a, b and d, as the lower order of Bézier function, the less computational complexity.

8 Conclusion

Considering the non-smooth term of semi-supervised support vector machines blocking the improvement in classification accuracy, a new class of Bézier functions is ultilized to approximate the hinge loss function, and a novel kind of Bézier smooth semi-supervised support vector machines (BS^{4}VMs) is constructed. The convergence proves the proposed model can draw close to the non-smooth objective function theoretically. As n-order Bézier function is $n-1$-order smooth and differentiable, the fast algorithm can be used to solve the programming. In contrast to the LDS^{4}VM, CS^{4}VM, and QS^{4}VM, experiments on artificial data, UCI data, USPS handwritten database, and NDC datasets clearly show that the BS^{4}VMs have the best performance and efficiency among exponential function, cubic spline function, and quintic spline

function. Moreover, the proposed algorithms show good performance for large-scale datasets. Due to the advantage of different order of BS^4VMs varying, when applying BS^4VMs, performance or efficiency priority should be paid attention. For further research, the feature selection and fuzzy membership should be good ways to improve the accuracy for different kinds of datasets. Bézier function for semi-supervised SVM on regression and its generalization performances will be explored as well.

Acknowledgements This work was supported by the Social Science Foundation of China under Grant (18ZDA027).

Compliance with ethical standards

Conflict of interest The author declares that they have no conflict of interest.

References

1. Bennett KP, Demiriz A (1999) Semi-supervised support vector machines. In: Kearns Michael S, Solla Sara A, Cohn David A (eds) Advances in neural information processing systems. MIT Press, London, pp 368–374
2. Reddy IS, Shevade S, Murty MN et al (2011) A fast quasi-Newton method for semi-supervised SVM. Pattern Recogn 44(10):2305–2313
3. Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. In: AISTATS 2005—Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, pp 57–64
4. Lanquillon C (2000) Learning from labeled and unlabeled documents: a comparative study on semi-supervised text classification. In: Zighed Djamel A, Komorowski Jan, Żytkow Jan (eds) Lecture notes in computer science. Springer, Berlin, pp 490–497
5. Liu CY, Jiang ZS, Su XX (2019) Detection of human fall using floor vibration and multi-features semi-supervised SVM. Sensors 19(17):3720
6. Kumar MP, Rajagopal MK (2019) Detecting facial emotions using normalized minimal feature vectors and semi-supervised twin support vector machines classifier. Appl Intell 49:4150–4174
7. Lang RL, Lu RB, Zhao CQ (2020) Graph-based semi-supervised one class support vector machine for detecting abnormal lung sounds. Appl Math Comput 364:124487
8. Ju Z, Gu H (2016) Predicting pupylation sites in prokaryotic proteins using semi-supervised self-training support vector machine algorithm. Anal Biochem 507:1–6
9. Xie XJ (2020) Multi-view semi-supervised least squares twin support vector machines with manifold-preserving graph reduction. Int J Mach Learn Cybern 11(11):2489–2499
10. Mygdalis V, Iosifidis A, Tefas A et al (2018) Semi-supervised subclass support vector data description for image and video classification. Neurocomputing 278:51–61
11. Liu CY, Gryllias K (2020) A semi-supervised support vector data description-based fault detection method for rolling element bearings based on cyclic spectral analysis. Mech Syst Signal Process 140:106682
12. Li Z, Tian Y, Li K et al (2017) Reject inference in credit scoring using semi-supervised support vector machines. Expert Syst Appl 74:105–114
13. Liu YQ, Liu SY, Gu MT (2009) Polynomial smooth classification algorithm of vector machines. Comput Sci (in Chinese) 36(7):179–181
14. Yang L, Wang L (2013) A class of smooth semi-supervised SVM by difference of convex functions programming and algorithm. Knowl-Based Syst 41:1–7
15. Zhang XD, Ma JG (2015) A general cubic spline smooth semi-supervised support vector machine. Chin J Eng 37:385–389
16. Zhang XD, Ma JG, Li AH et al (2015) Quintic spline smooth semi-supervised support vector classification machine. J Syst Eng Electron 26:626–632
17. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. Chapman and Hall/CRC, London
18. Bézier P (1968) Renault uses numerical control for car body design and tooling[C]//Paper Sae 680010, Society of Automotive Engineers Congress
19. Choi JW, Elkaim GH (2008) Bézier curve for trajectory guidance. World Congr Eng Comput Sci WCECS 2173(1):22–24
20. Mandad M, Campen M (2020) Bézier guarding: precise higher-order meshing of curved 2D domains. ACM Trans Graph 39(4):103–118
21. Raja SP (2020) Bézier and B-spline curves - a study and its application in wavelet decomposition. Int J Wavelets Multiresolut Inf Process 18(4):2050030
22. Zhu YF, Xu G, Ling CN (2019) Construction of energy-minimizing Bézier surfaces interpolating given diagonal curves. J Image Graph 24(11):1998–2008
23. Wu Q, Wang E (2015) Bézier function smooth support vector regression. ICIC express letters. Part B Appl Int J Res Surv 6:1773–1779
24. Nocedal J, Wright SJ (1999) Numerical optimization. Springer, New York
25. Yu J, Vishwanathau SVN, Gunter S et al (2010) A Quasi-Newton approach to nonsmooth convex optimization problems in machine learning. J Mach Learn Res 11:1145–1200
26. Chen W, Shao Y, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. Int J Mach Learn Cybern 5:459–468
27. Ho TK, and Kleinberg EM (1996) "Checkerboard dataset", http://www.cs.wisc.edu/∼musicant/data/ndc/ , accessed on July 20 2020
28. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30