A Regularized Nonsmooth Newton Method for Multi-class Support Vector Machines[1]

Ping Zhong[2] and Masao Fukushima[3]

March 3, 2005

Abstract Multi-class classification is an important and on-going research subject in machine learning. Recently, the ν-K-SVCR method was proposed by the authors for multi-class classification. Since many optimization problems have to be solved in multi-class classification, it is extremely important to develop an algorithm that can solve those optimization problems efficiently. In this paper, the optimization problem in the ν-K-SVCR method is reformulated as an affine box constrained variational inequality problem with a positive semidefinite matrix, and a regularized version of the nonsmooth Newton method that uses the D-gap function as a merit function is applied to solve the resulting problems. The proposed algorithm fully exploits the typical feature of the ν-K-SVCR method, which enables us to reduce the size of Newton equations significantly. This indicates that the algorithm can be implemented efficiently in practice. The preliminary numerical experiments on benchmark datasets show that the proposed method is considerably faster than the standard Matlab routine used in the original ν-K-SVCR method.

Keywords: Multi-class classification; Box constrained variational inequality problem; Nonsmooth Newton method; D-gap function; Regularization methods

AMS Subject classification: 65K05, 68T10, 68Q32

1 Introduction

The support vector machine for multi-class classification is an important and on-going issue in data mining and machine learning. It refers to the construction of an approximation of

[1] This work was supported in part by the Scientific Research Grant-in-Aid from Japan Society for the Promotion of Science and the National Science Foundation of China Grant No. 10371131.

[2] Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan./Faculty of Science, China Agricultural University, Beijing, 100083, China. Email: pingsunshine@hotmail.com

[3] Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. Email: fuku@amp.i.kyoto-u.ac.jp

an unknown function defined from an input space $\mathcal{X} \subset R^N$ onto an unordered set of classes $\mathcal{Y} = \{\Theta_1, \cdots, \Theta_K\}$ based on independently and identically distributed (i.i.d.) data

$$\mathcal{T} = \{(\mathbf{x}_p, \theta_p)\}_{p=1}^m \subset \mathcal{X} \times \mathcal{Y}. \tag{1}$$

Currently, there are two types of approaches for multi-class classification. One is the "decomposition-reconstruction" architecture approach which consists of the 'one-against-all' method [5, 20], the 'one-against-one' method [10, 12], the directed acyclic graph SVM method [16], the 'error-correcting output code' method [7, 1], and '1-versus-1-versus-rest' method [2, 23]. The other is the "all-together" approach [20, 21, 3, 13, 6]. In this paper, we focus on the "decomposition-reconstruction" architecture approach. The recently proposed K-SVCR method [2] and ν-K-SVCR method [23] improve the standard one-against-all and one-against-one structures by the 1-versus-1-versus-rest structure. Both of them use the mixed classification and regression SV (or ν-SV) machine formulations in the decomposing scheme. In particular, the ν-K-SVCR method has parameters that allow us to control the numbers of support vectors and margin errors effectively, which is helpful in improving the accuracy of each classifier. For K-class classification, they construct $K(K-1)/2$ classifiers, which means that we need to solve $K(K-1)/2$ quadratic programs to assign a new pattern to the proper class. So it is important to establish a fast algorithm for solving these programs. However, in [2] and [23], the resulting quadratic programs are solved only by the standard packages which cannot be used even for moderately large data sets.

The quadratic programs in [2] and [23] are convex quadratic programs. The first-order optimality conditions can be reformulated as an affine box constrained variational inequality problem (BVIP) with a positive semi-definite matrix. In this paper, we develop a Newton-type algorithm for solving the optimization problem derived from the ν-K-SVCR method. For this purpose, the nonsmooth Newton method proposed by Kanzow and Fukushima [11] is prescribed. This method is globally and fast locally convergent. Moreover, it has a finite termination property for the BVIP described by an affine function. However, the results for the affine BVIP assume that the problem involves a P-matrix, which is not the case for the quadratic program in the ν-K-SVCR method. Therefore we incorporate a regularization technique in the Kanzow-Fukushima method and use it to solve the support vector machine problem. The proposed algorithm fully exploits the typical feature of the ν-K-SVCR method, which enables us to reduce the size of Newton equations significantly. This indicates that the algorithm can be implemented efficiently in practice. Preliminary numerical experiments on benchmark datasets show that the proposed approach substantially improves the training speed.

The paper is organized as follows: In Section 2, we reformulate the optimization problem in the ν-K-SVCR method as an affine BVIP with a positive semi-definite matrix. In Section 3, we describe the nonsmooth Newton method proposed by Kanzow and Fukushima and regularize it for our purpose. In Section 4, we report numerical results to show the performance of the algorithm. Section 5 concludes the paper.

2 Reformulation

In this section, we describe the optimization problem in the ν-K-SVCR method [23] that leads to an affine BVIP with a positive semi-definite matrix. First, we recall the optimization problem in the ν-K-SVCR method.

Let the training set $\mathcal{T}$ be given by (1). For an arbitrary pair $(\Theta_j, \Theta_k) \in \mathcal{Y} \times \mathcal{Y}$ of classes, the ν-K-SVCR method constructs a decision function $f(\mathbf{x})$ which separates the two classes Θ_j and Θ_k as well as the remaining classes. Without loss of generality, let patterns $\mathbf{x}_i,\ i = 1, \cdots, m_1$, and $\mathbf{x}_i,\ i = m_1 + 1, \cdots, m_1 + m_2$ belong to the two classes Θ_j and Θ_k which will be labelled $+1$ and -1, respectively, and the remaining patterns belong to the other classes which will be labelled 0. Then $f(\mathbf{x})$ is supposed to satisfy

$$f(\mathbf{x}_i) = \begin{cases} +1, & i = 1, \cdots, m_1, \\ -1, & i = m_1 + 1, \cdots, m_1 + m_2, \\ 0, & i = m_1 + m_2 + 1, \cdots, m. \end{cases} \tag{2}$$

For simplicity, we denote $m_{12} = m_1 + m_2$ and $m_3 = m - m_{12}$ in the following. For $\nu_1, \nu_2 \in (0, 1]$ and $C, D > 0$ chosen a priori, the ν-K-SVCR method needs to solve the following optimization problem [23]:

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\Big(\frac{1}{m_{12}}\sum_{i=1}^{m_{12}} \xi_i - \nu_1 \rho\Big) + D\Big(\frac{1}{m_3}\sum_{i=m_{12}+1}^{m} (\varphi_i + \tilde{\varphi}_i) + \nu_2 \varepsilon\Big) & (3) \\
\text{s.t.} \quad & \mathrm{y}_i \cdot ((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b) \geq \rho - \xi_i,\ i = 1, \cdots, m_{12}, & (4) \\
& (\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b \leq \varepsilon + \varphi_i,\ i = m_{12} + 1, \cdots, m, & (5) \\
& (\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b \geq -\varepsilon - \tilde{\varphi}_i,\ i = m_{12} + 1, \cdots, m, & (6) \\
& \xi_i, \varphi_i, \tilde{\varphi}_i, \varepsilon \geq 0, & (7) \\
& \rho \geq \varepsilon, & (8)
\end{aligned}$$

where $\mathrm{y}_i = 1,\ i = 1, \cdots, m_1$ and $\mathrm{y}_i = -1, i = m_1 + 1, \cdots, m_{12}$. The dual of the problem (3)–(8) can be expressed as follows: For $\nu_1, \nu_2 \in (0, 1]$ and $C, D > 0$ chosen a priori,

$$\begin{aligned}
\min \quad & \frac{1}{2}\gamma^T Q \gamma & (9) \\
\text{s.t.} \quad & a_1^T \gamma - C\nu_1 \geq 0, & (10) \\
& -a_2^T \gamma + D\nu_2 \geq 0, & (11) \\
& h^T \gamma = 0, & (12) \\
& l \leq \gamma \leq u, & (13)
\end{aligned}$$

where

$$
\begin{aligned}
&a_1 = (\mathrm{y}_1, \cdots, \mathrm{y}_{m_1}, \mathrm{y}_{m_1+1}, \cdots, \mathrm{y}_{m_{12}}, 0, \cdots, 0, 0, \cdots, 0)^T \in R^{m_1+m_2+m_3+m_3},\\
&a_2 = (0, \cdots, 0, 0, \cdots, 0, 1, \cdots, 1, 1, \cdots, 1)^T \in R^{m_1+m_2+m_3+m_3},\\
&h = (1, \cdots, 1, 1, \cdots, 1, -1, \cdots, -1, 1, \cdots, 1)^T \in R^{m_1+m_2+m_3+m_3},\\
&l = (0, \cdots, 0, -\frac{C}{m_{12}}, \cdots, -\frac{C}{m_{12}}, 0, \cdots, 0, 0, \cdots, 0)^T \in R^{m_1+m_2+m_3+m_3},\\
&u = (\frac{C}{m_{12}}, \cdots, \frac{C}{m_{12}}, 0, \cdots, 0, \frac{D}{m_3}, \cdots, \frac{D}{m_3}, \frac{D}{m_3}, \cdots, \frac{D}{m_3})^T \in R^{m_1+m_2+m_3+m_3},
\end{aligned}
$$

and

$$
Q = \begin{pmatrix} (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \\ -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) \\ (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \end{pmatrix} \in R^{(m_{12}+m_3+m_3)\times(m_{12}+m_3+m_3)}
$$

is a symmetric positive semi-definite matrix with $\mathbf{x}_i = \mathbf{x}_{i-m_3},\ i = m+1, \cdots, m+m_3$, and $k(\cdot, \cdot)$ being a kernel function. The typical choices of kernel functions include polynomial kernels $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$, $d \in N$, and Gaussian kernels $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\kappa)$, $\kappa > 0$.

The ν-K-SVCR method has two adjustable parameters ν_1 and ν_2. It has been proved in [23] that $\frac{\nu_1 m_{12}+\nu_2 m_3}{m}$ provides both an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. Moreover, when the pattern size goes to infinitely, both fractions converge almost surely to $\frac{\nu_1 m_{12}+\nu_2 m_3}{m}$ under general assumptions on the learning problems and the kernels used ([23], Theorem 3.1). Therefore, we can use the parameters ν_1 and ν_2 to control the number of margin errors, which is helpful in improving the accuracy of each classifier. In addition, each classification hyperplane generated by the ν-K-SVCR method has the 'outlier' resistance property, that is, if the classification hyperplane normal vector $\mathbf{w}$ can be expressed in terms of the support vectors that are not margin errors, then a sufficiently small perturbation of any margin error along the direction $\mathbf{w}$ in feature space does not change the hyperplane ([23], Theorem 3.2). This shows that the classifiers are robust.

In the following, we occasionally denote $\tilde{m} = m + m_3 = m_1 + m_2 + m_3 + m_3$. Define $L : R^{\tilde{m}+2+1+\tilde{m}+\tilde{m}} \to R^{\tilde{m}}$ by

$$
L(\gamma, \lambda, \mu, \upsilon, \omega) = Q\gamma - \lambda_1 a_1 + \lambda_2 a_2 + \mu h - \upsilon + \omega.
$$

Then the KKT system for the quadratic program (9)–(13) can be written as

$$
\begin{cases}
L(\gamma, \lambda, \mu, \upsilon, \omega) = 0,\\
\lambda_1 \geq 0,\ a_1^T\gamma - C\nu_1 \geq 0,\ \lambda_1^T(a_1^T\gamma - C\nu_1) = 0,\\
\lambda_2 \geq 0,\ -a_2^T\gamma + D\nu_2 \geq 0,\ \lambda_2^T(-a_2^T\gamma + D\nu_2) = 0,\\
\mu \text{ free},\ -h^T\gamma = 0,\\
\upsilon \geq 0,\ \gamma - l \geq 0,\ \upsilon^T(\gamma - l) = 0,\\
\omega \geq 0, -\gamma + u \geq 0,\ \omega^T(-\gamma + u) = 0.
\end{cases}
\tag{14}
$$

Let $[l,u]$ denote the set $\{\gamma \in R^{\tilde{m}} |\ l_i \leq \gamma_i \leq u_i,\ i = 1, \cdots, \tilde{m}\}$. Define $\tilde{L} : R^{\tilde{m}+2+1} \to R^{\tilde{m}}$ by

$$\widetilde{L}(\gamma, \lambda, \mu) = Q\gamma - \lambda_1 a_1 + \lambda_2 a_2 + \mu h$$

and $r : R^{\tilde{m}+2+1} \to R^{\tilde{m}+2+1}$ by

$$r(\gamma, \lambda, \mu) = \begin{pmatrix} \gamma - \Pi_{[l,\,u]}\left[\gamma - \widetilde{L}(\gamma, \lambda, \mu)\right] \\ \lambda_1 - \Pi_{R_+}\left[\lambda_1 - (a_1^T\gamma - C\nu_1)\right] \\ \lambda_2 - \Pi_{R_+}\left[\lambda_2 - (-a_2^T\gamma + D\nu_2)\right] \\ -h^T\gamma \end{pmatrix},$$

where $\Pi_X(x)$ denotes the Euclidean projection of a vector x on the set X. If $(\gamma^*, \lambda^*, \mu^*, \upsilon^*, \omega^*)$ is a solution of the KKT system (14), then $(\gamma^*, \lambda^*, \mu^*)$ satisfies $r(\gamma^*, \lambda^*, \mu^*) = 0$. Conversely, if $(\gamma^*, \lambda^*, \mu^*)$ is a solution of $r(\gamma, \lambda, \mu) = 0$, then $(\gamma^*, \lambda^*, \mu^*, \upsilon^*, \omega^*)$ is a solution of the KKT system (14) with υ^* and ω^* defined by

$$\upsilon^* = \Pi_{R_+^{\tilde{m}}}(\widetilde{L}(\gamma^*, \lambda^*, \mu^*)), \quad \omega^* = \Pi_{R_+^{\tilde{m}}}(-\widetilde{L}(\gamma^*, \lambda^*, \mu^*)).$$

So finding a solution of the KKT system (14) is equivalent to solving $r(\gamma, \lambda, \mu) = 0$. Let $z = (\gamma, \lambda, \mu)$, $X = [l, u] \times R_+^2 \times R$, and

$$F(z) = \begin{pmatrix} \widetilde{L}(z) \\ a_1^T\gamma - C\nu_1 \\ -a_2^T\gamma + D\nu_2 \\ -h^T\gamma \end{pmatrix}. \tag{15}$$

Then $r(\gamma, \lambda, \mu) = 0$ can be written as

$$r(z) = z - \Pi_X(z - F(z)) = 0. \tag{16}$$

The function $r(z)$ is the so-called natural residual of the following box constrained variational inequality problem (BVIP): Find $z^* \in X$ such that

$$F(z^*)^T(z - z^*) \geq 0 \quad \text{for all} \quad z \in X. \tag{17}$$

We denote the problem (17) as BVI(F, X). Solving BVI(F, X) is equivalent to solving (16). The function $F(z)$ defined by (15) can be written as

$$F(z) = \begin{pmatrix} Q & -a_1 & a_2 & h \\ a_1^T & 0 & 0 & 0 \\ -a_2^T & 0 & 0 & 0 \\ -h^T & 0 & 0 & 0 \end{pmatrix} z + \begin{pmatrix} 0 \\ -C\nu_1 \\ D\nu_2 \\ 0 \end{pmatrix}.$$

For simplicity, we denote

$$\hat{Q} = \begin{pmatrix} Q & -a_1 & a_2 & h \\ a_1^T & 0 & 0 & 0 \\ -a_2^T & 0 & 0 & 0 \\ -h^T & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad q = \begin{pmatrix} 0 \\ -C\nu_1 \\ D\nu_2 \\ 0 \end{pmatrix}.$$

Since Q is symmetric positive semi-definite, $\hat{Q}$ is positive semi-definite. In this paper, we use a regularized version of the nonsmooth Newton method proposed by Kanzow and Fukushima [11] to solve BVI(F, X). This nonsmooth Newton method is globalized by means of the D-gap function [15, 22] defined by

$$g_{\alpha\beta}(z) = g_\alpha(z) - g_\beta(z), \tag{18}$$

where $0 < \alpha < \beta$, and $g_\alpha(z)$ is the regularized gap function [8] defined by

$$g_\alpha(z) = F(z)^T(z - y_\alpha(z)) - \frac{\alpha}{2}\|z - y_\alpha(z)\|^2 \tag{19}$$

with $y_\alpha(z) = \Pi_X\left(z - \alpha^{-1}F(z)\right)$, and likewise $g_\beta(z)$.

3 Algorithm

Kanzow and Fukushima [11] propose a nonsmooth Newton method based on the D-gap function as a merit function to solve BVIP. This method is globally and fast locally convergent. Moreover, it has a finite termination property for BVIP involving an affine function. However, we note that this method is not well defined for the BVI(F, X) with the function F given by (15) since in this case the underlying Jacobian may be singular. In the following, we give a regularized nonsmooth Newton method for finding a solution of BVI(F, X).

Instead of solving the original BVI(F, X) directly, the regularization method solves a sequence of regularized problems obtained by replacing F by F_ϵ, where ϵ is a positive parameter converging to 0. If we adopt the Tikhonov regularization, this scheme consists of solving a sequence of BVI(F_ϵ, X) with

$$F_\epsilon(z) = (\hat{Q} + \epsilon I)z + q.$$

Recall that $\hat{Q}$ is positive semi-definite. Therefore, $\hat{Q} + \epsilon I$ is positive definite, and hence BVI(F_ϵ, X) has the unique solution $z(\epsilon)$ for any $\epsilon > 0$. For a fixed $\epsilon > 0$, let $r_\epsilon(z)$ denote the natural residual of BVI(F_ϵ, X), i,e.,

$$r_\epsilon(z) = z - \Pi_X(z - F_\epsilon(z)).$$

It is known that $r_\epsilon(z)$ is semismooth. The B-subdifferential of $r_\epsilon(z)$ at z is defined by

$$\partial_B r_\epsilon(z) = \left\{ H \in R^{(\tilde{m}+3)\times(\tilde{m}+3)} \mid H = \lim_{k\to\infty} F'_\epsilon(z^k), z = \lim_{k\to\infty} z^k, F'_\epsilon(z^k) \text{ exists for all } k \right\},$$

where $F'_\epsilon(z^k)$ is the Jacobian of F_ϵ at $z^k \in R^{\tilde{m}+3}$, and the Clarke generalized Jacobian of $r_\epsilon(z)$ at z is defined by

$$\partial r_\epsilon(z) = \text{conv}\, \partial_B r_\epsilon(z),$$

where conv denotes the convex hull. Denote $X = [l, u] \times R^2_+ \times R = \prod_{i=1}^{\tilde{m}+3} [\underline{X}_i, \bar{X}_i]$, and

$$\varrho_1 = \varrho_1(z) = \{i | z_i - F_{\epsilon,i}(z) \in (\underline{X}_i, \bar{X}_i)\},$$

$$\varrho_2 = \varrho_2(z) = \{i | z_i - F_{\epsilon,i}(z) \in \{\underline{X}_i, \bar{X}_i\}\},$$

$$\varrho_3 = \varrho_3(z) = \{i | z_i - F_{\epsilon,i}(z) \notin [\underline{X}_i, \bar{X}_i]\},$$

where $F_{\epsilon,i}(z)$ is the ith component function of the vector-valued function F_ϵ. Then any $H \in \partial_B r_\epsilon(z)$ can be expressed as

$$H = \begin{pmatrix} [F'_\epsilon(z)]_{\varrho_1\cup\delta,\, \varrho_1\cup\delta} & [F'_\epsilon(z)]_{\varrho_1\cup\delta,\, \bar{\delta}\cup\varrho_3} \\ 0_{\bar{\delta}\cup\varrho_3,\, \varrho_1\cup\delta} & I_{\bar{\delta}\cup\varrho_3,\, \bar{\delta}\cup\varrho_3} \end{pmatrix} \tag{20}$$

for some index set δ with $\delta \subseteq \varrho_2$, and $\bar{\delta} = \varrho_2 \setminus \delta$ denotes the complement of δ in ϱ_2. A solution $z(\epsilon)$ of BVI(F_ϵ, X) is said to be *b-regular* if the submatrices

$$[F'_\epsilon(z(\epsilon))]_{\varrho_1\cup\delta,\, \varrho_1\cup\delta}$$

are nonsingular for all index sets δ such that $\delta \subseteq \varrho_2$, where $\varrho_1 = \varrho_1(z(\epsilon))$ and $\varrho_2 = \varrho_2(z(\epsilon))$. Since $\hat{Q} + \epsilon I$ is positive definite, it is easy to see that the unique solution $z(\epsilon)$ of BVI(F_ϵ, X) is b-regular. The nonsmooth Newton method for BVI(F_ϵ, X) is globalized by the D-gap function

$$g_{\alpha\beta\epsilon}(z) = g_{\alpha\epsilon}(z) - g_{\beta\epsilon}(z),$$

where $0 < \alpha < \beta$, and

$$g_{\alpha\epsilon}(z) = F_\epsilon(z)^T (z - y_{\alpha\epsilon}(z)) - \frac{\alpha}{2} \|z - y_{\alpha\epsilon}(z)\|^2$$

with $y_{\alpha\epsilon}(z) = \Pi_X\left(z - \alpha^{-1} F_\epsilon(z)\right)$, and likewise $g_{\beta\epsilon}(z)$.

Algorithm 3.1 (The nonsmooth Newton method for BVI(F_ϵ, X))

(S.0) Choose $z^0 \in R^{\tilde{m}+3}$, $0 < \alpha < 1 < \beta$, $\zeta > 0$, $\tau \in (0, 1)$, $\sigma \in (0, 1/2)$, $p > 1$, $\eta \in (0, 1)$, $\theta \geq 0$, and set $k = 0$.

(S.1) If $\|\nabla g_{\alpha\beta\epsilon}(z^k)\| \leq \theta$, stop.

(S.2) Select an arbitrary element $H_k \in \partial_B r_\epsilon(z^k)$.

(a) Find a solution $d^k \in R^{\tilde{m}+3}$ of the Newton equation

$$H_k d = -r_\epsilon(z^k). \tag{21}$$

If it is not solvable, then set $d^k = -\nabla g_{\alpha\beta\epsilon}(z^k)$ and go to (S.3), else go to (b).

(b) If

$$g_{\alpha\beta\epsilon}(z^k + d^k) \leq \eta g_{\alpha\beta\epsilon}(z^k),$$

then set $t_k = 1$ and go to (S.4). Else, go to (c).

(c) If d^k does not satisfy the decent condition

$$\nabla g_{\alpha\beta\epsilon}(z^k)^T d^k \leq -\zeta \|d^k\|^p,$$

then set $d^k = -\nabla g_{\alpha\beta\epsilon}(z^k)$. Go to (S.3).

(S.3) Let t_k be the largest element in the set $\{1, \tau, \tau^2, \cdots, \}$ such that

$$g_{\alpha\beta\epsilon}(z^k + t_k d^k) \leq g_{\alpha\beta\epsilon}(z^k) + \sigma t_k \nabla g_{\alpha\beta\epsilon}(z^k)^T d^k.$$

(S.4) Set $z^{k+1} = z^k + t_k d^k$, $k = k + 1$, and go to (S.1).

Since F_ϵ is affine and $z(\epsilon)$ is a b-regular solution of BVI(F_ϵ, X) as mentioned above, Algorithm 3.1 can find $z(\epsilon)$ in a finite number of iterations [11].

Usually the training set $\mathcal{T}$ contains a large number of patterns. The typical feature of the dual problem (9)–(13) is that the number of variables is large but there are only three constraints except for the simple bound constraints on the variables. Moreover, by the nature of the ν-K-SCVR method, a large portion of the dual variables γ are expected to satisfy one of these bound constraints as an equality at the optimal solution. This means that the index sets $\varrho_1(z)$ and $\varrho_2(z)$ are usually very small compared with $\varrho_3(z)$. By exploiting the special structure of H defined by (20), the Newton equations (21) solved in Step (S.2) of Algorithm 3.1 can be written as

$$\begin{aligned} &[F'_\epsilon(z^k)]_{\varrho_1\cup\delta,\, \varrho_1\cup\delta}\, d_{\varrho_1\cup\delta} = -[r_\epsilon(z^k)]_{\varrho_1\cup\delta} + [F'_\epsilon(z^k)]_{\varrho_1\cup\delta,\, \bar{\delta}\cup\varrho_3}[r_\epsilon(z^k)]_{\bar{\delta}\cup\varrho_3}, \\ &d_{\bar{\delta}\cup\varrho_3} = -[r_\epsilon(z^k)]_{\bar{\delta}\cup\varrho_3}, \end{aligned}$$

with $\varrho_1 = \varrho_1(z^k)$, $\varrho_2 = \varrho_2(z^k)$ and $\varrho_3 = \varrho_3(z^k)$. Since the coefficient matrix $[F'_\epsilon(z^k)]_{\varrho_1\cup\delta,\, \varrho_1\cup\delta}$ of the first equation is small as mentioned above, the Newton equations (21) are reduced significantly.

Let S denote the solution set of BVI(F, X). Recall that BVI(F, X) is equivalent to the convex quadratic problem (9)–(13). Since the latter problem has a nonempty solution set, S is nonempty. The following lemma shows the boundedness of $\{z(\epsilon_k) : \epsilon_k > 0\}$, where $z(\epsilon_k)$ is the solution of BVI(F_{ϵ_k}, X). The result is similar to the ones for the monotone complementarity problems [18, 19]. For the sake of completeness, we give a proof.

Lemma 3.2 For $\epsilon_k > 0$, let $z(\epsilon_k)$ denote the solution of BVI(F_{ϵ_k}, X). Then $\{z(\epsilon_k) : \epsilon_k > 0\}$ is bounded.

Proof. Denote $v(\epsilon_k) = (\hat{Q} + \epsilon_k I)z(\epsilon_k) + q$ and $u = \hat{Q}\bar{z} + q$ where $\bar{z} \in S$. Then

$$v(\epsilon_k)^T(\bar{z} - z(\epsilon_k)) \geq 0 \quad \text{and} \quad u^T(z(\epsilon_k) - \bar{z}) \geq 0.$$

Hence

$$0 \leq (v(\epsilon_k) - u)^T(\bar{z} - z(\epsilon_k)) = \epsilon_k z(\epsilon_k)^T(\bar{z} - z(\epsilon_k)) - \left(\hat{Q}(z(\epsilon_k) - \bar{z})\right)^T (z(\epsilon_k) - \bar{z}). \tag{22}$$

Since $\hat{Q}$ is positive semi-definite,

$$\left(\hat{Q}(z(\epsilon_k) - \bar{z})\right)^T (z(\epsilon_k) - \bar{z}) \geq 0.$$

Hence by (22) and $\epsilon_k > 0$ we have

$$z(\epsilon_k)^T(\bar{z} - z(\epsilon_k)) \geq 0. \tag{23}$$

By (23) and Cauchy-Schwarz inequality, we have

$$\|z(\epsilon_k)\| \leq \|\bar{z}\|. \tag{24}$$

This proves that $\{z(\epsilon_k) : \epsilon_k > 0\}$ is bounded. □

By Lemma 3.2, we see that $\lim_{\epsilon_k \to 0} \epsilon_k \|z(\epsilon_k)\| = 0$. Then by Theorem 3.1 of [17], $\{z(\epsilon_k)\}$ forms a minimizing sequence for the D-gap function $g_{\alpha\beta}(\cdot)$ of the original BVI(F, X), i.e.,

$$\lim_{\epsilon_k \to 0} g_{\alpha\beta}(z(\epsilon_k)) = 0. \tag{25}$$

The result below further shows that $\{z(\epsilon_k)\}$ converges to the least 2-norm solution of BVI(F, X). Similar results for complementarity problems can be found in [18, 19].

Theorem 3.3 Let $\{\epsilon_k\}$ be a positive sequence monotonically decreasing to 0 and let $z(\epsilon_k)$ denote the solution of BVI(F_{ϵ_k}, X). Then $\{z(\epsilon_k)\}$ converges to the least 2-norm solution $z^* \in S$ of BVI(F, X) as $\epsilon_k \downarrow 0$.

Proof. By Lemma 3.2, $z(\epsilon_j) \to z^*$ for some subsequence $\epsilon_j \to 0$. Then by (25) and the fact that $g_{\alpha\beta}(z)$ is continuous on $R^{\tilde{m}+3}$, we get

$$g_{\alpha\beta}(z^*) = 0. \tag{26}$$

Recall that $g_{\alpha\beta}(z^*) = 0$ if and only if z^* solves BVI(F, X). Hence $z^* \in S$. By (24), we get that for any $\bar{z} \in S$, $\|z^*\| \leq \|\bar{z}\|$. Since S is convex, z^* is the unique least 2-norm solution of BVI(F, X). So we get that any convergent subsequence of $\{z(\epsilon_k)\}$ will converge to z^*. Now we prove the fact that $\{z(\epsilon_k)\}$ converges to z^* as $\epsilon_k \downarrow 0$, i.e., for any $\varepsilon > 0$, there exists $\delta > 0$ such that when $\epsilon_k < \delta$,

$$\|z(\epsilon_k) - z^*\| \leq \varepsilon.$$

If it is not true, for $\frac{1}{m}$ $(m = 1, 2, \cdots)$, there exists $\epsilon_m < \frac{1}{m}$ such that

$$\|z(\epsilon_m) - z^*\| > \varepsilon. \tag{27}$$

However, since $\{z(\epsilon_m)\}$ is bounded and infinite, there must exist a subsequence of $\{z(\epsilon_m)\}$ which converges to z^*. This contradicts (27), and the proof is complete. □

The sequence $\{z(\epsilon_k)\}$ converges to the solution of BVI(F, X) where the value of the D-gap function is zero. We use this fact as the termination criterion in the following regularization method.

Algorithm 3.4 (The regularization method for BVI(F, X))

```
(0) Choose $\epsilon_0 > 0$, $\lambda \in (0, 1)$, $\bar{\theta} \geq 0$, and set $k = 0$.
(1) Compute an approximate solution $z(\epsilon_k)$ of BVI$(F_{\epsilon_k}, X)$ by Algorithm 3.1.
(2) If $g_{\alpha\beta}(z(\epsilon_k)) \leq \bar{\theta}$, stop.
(3) Set $\epsilon_{k+1} = \lambda\epsilon_k$, set $k = k + 1$, and go to step (1).
```

4 Preliminary numerical results

The datasets used for our numerical experiments are taken from the UCI machine learning repository [4]: iris, wine, glass, vowel, vehicle and segment. Table 1 gives the description of the six benchmark problems. Since no test data sets are provided in the six benchmark datasets, we use tenfold cross validation to evaluate the performance of the algorithm. That is, each dataset is split randomly into ten subsets and one of those sets is reserved as a test set; this process is repeated ten times. We implemented our numerical experiments using Matlab v6.5 on Intel Pentium IV 3.00GHz PC with 1GB of RAM.

The implementation uses the following parameters: In Algorithm 3.4, $\epsilon_0 = 0.1$, $\lambda = 0.1$, $\bar{\theta} = 10^{-4}$. In Algorithm 3.1, $\alpha = 0.99$, $\beta = 1.001$, $\zeta = 10^{-8}$, $\tau = 0.5$, $\sigma = 10^{-4}$, $p = 1.1$, $\eta = 0.9$, $\theta = 10^{-5}$. The initial point z^0 is first set 0 and then the solution obtained in the kth iteration is used as the starting point of the next $(k+1)$th iteration. The parameters ν_1 and ν_2 in (3) are both set 0.01. In Algorithm 3.1, we replaced the standard Armijo-rule in (S.3) by

Table 1: Six benchmark datasets from UCI

name	iris	wine	glass	vowel	vehicle	segment
#pts	150	178	214	528	846	2310
#ats	4	13	9	10	18	19
#cls	3	3	6	11	4	7

#pts: the number of training data;
#ats: the number of attributes of patterns;
#cls: the number of classes.

a nonmonotone Armijo-rule [9], and employed the following two termination criteria in (S.1):

$$\|r_\epsilon(z^k)\| \leq \theta \tag{28}$$

and

$$\|r_\epsilon(z^k)\| \leq \max\{\theta, \epsilon_k\}. \tag{29}$$

We report the results of Algorithm 3.4 for the six benchmark datasets with termination criteria (28) and (29) in Table 2 and Table 3, respectively. It can be observed from Table 2 and Table 3 that the testing correctness is almost the same for the two termination criteria (28) and (29), but the training time with (29) is faster than the one with (28).

Table 2: Results of Algorithm 3.4 with termination criterion (28)

name	iris	wine	glass	vowel	vehicle	segment**
kernel	G, $\kappa = 10$	P, $d = 1$	G,$\kappa = 1$	G, $\kappa = 1$	G, $\kappa = 1e + 4$	G, $\kappa = 6e + 3$
PER	0.67	[0.56, 10]*	[27.14, 36.67]	0.38	[8.93,15.12]	[1.52, 2.55]
time	0.63	1.56	8.24	180.84	60.94	3.03

kernel: the employed kernel: 'P' denotes polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$, $d \in N$, and 'G' denotes Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\kappa)$, $\kappa > 0$;
PER: the percentage of error on the validation set;
time: execution time for tenfold training (in seconds).
* $[\cdot, \cdot]$ in 'PER' row refers to the case where two or more classes get the most votes in a tie. The first and second numbers in the brackets are the percentage of error when patterns are assigned to the right and the wrong classes, respectively, among those with the most votes.
** For segment, 10% data of each class were randomly chosen in order to obtain a smaller kernel matrix that would fit in memory. A similar way to reduce the kernel matrix can be found in [14].

Table 3: Results of Algorithm 3.4 with termination criterion (29)

name	iris	wine	glass	vowel	vehicle	segment
kernel	G, $\kappa = 10$	P, $d = 1$	G, $\kappa = 1$	G, $\kappa = 1$	G, $\kappa = 1e + 4$	G, $\kappa = 6e + 3$
PER	0.67	[0.56, 10]	[26.67, 29.52]	0.38	[8.33,13.10]	[1.52, 2.55]
time	0.42	1.41	4.97	83.79	31.36	2.80

In [23], the optimization problems (3)–(8) are solved by the standard Matlab routine. Table 4 shows the execution time (in seconds) for tenfold training by the standard routine in [23] for iris, wine and glass datasets.

Table 4: Results of the standard Matlab routine for the ν-K-SVCR method in [23]

name	iris	wine	glass
time	104.81	134.92	2266.9

It can be observed from Table 2, Table 3 and Table 4 that the regularized nonsmooth Newton method performs much more efficiently than the standard routine. Specifically, for the iris, wine and glass datasets, the training time is substantially small. We note that for the vowel dataset, since it has 11 classes, 55 optimization problems need to be solved to assign a pattern to a class. So, for the vowel as well as the large datasets vehicle and segment, the training time becomes the major issue. The regularized nonsmooth Newton method has shown good performance on these datasets.

5 Conclusion

We have reformulated the optimization problem in the ν-K-SVCR method as an affine BVIP with a positive semi-definite matrix and proposed a regularized version of the nonsmooth Newton method for solving the resulting problems. This method fully exploits the special feature of the problem and the numerical experiments have shown that this method is considerably faster than the standard routine. In particular, it also has shown good performance on large datasets for which the standard routine may not be suitable.

References

[1] E. L. Allwein, R. E. Schapire and Y. Singer (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, **1**, 113–141.

[2] C. Angulo, X. Parra and A. Català (2003). K-SVCR. A support vector machine for multi-class classification. *Neurocomputing*, **55**, 57–77.

[3] K.P. Bennett (1999). Combining support vector and mathematical programming methods for classification. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 307–326. MIT Press, Cambridge, MA.

[4] C. L. Blake and C. J. Merz (1998). UCI repository of machine learning databases. University of California. [www http://www.ics.uci.edu/∼mlearn/MLRepository.html]

[5] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Müller, E. Sackinger, P. Simard and V. Vapnik (1994). Comparison of classifier methods: a case study in handwriting digit recognition. in: IAPR (Ed.), *Proceedings of the International Conference on Pattern Recognition*, pp. 77–82. IEEE Computer Society Press.

[6] K. Crammer and Y. Singer (2002). On the algorithmic implementation of multiclass kernel-basd vector machines. *Journal of Machine Learning Research*, **2**, 265–292.

[7] T.G. Dietterich and G. Bakiri (1995). Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**, 263–286.

[8] M. Fukushima (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical Programming*, **53**, 99–110.

[9] L. Grippo, F. Lampariello and S. Lucidi (1986). A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23**, 707–716.

[10] T. J. Hastie and R. J. Tibshirani (1998). Classification by pairwise coupling. in: M. I. Jordan, M. J. Kearns and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, **10**, 507–513. MIT Press, Cambridge, MA.

[11] C. Kanzow and M. Fukushima (1998). Solving box constrained variational inequalities by using the natural residual with D-gap function globalization. *Operations Research Letters*, **23**, 45–51.

[12] U. Kreßel (1999). Pairwise classification and support vector machines. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 255–268. MIT Press, Cambridge, MA.

[13] Y. Lee, Y. Lin and G. Wahba (2001). Multicategory support vector machines. *Computing Science and Statistics*, **33**, 498–512.

[14] Y.-J. Lee and O. L. Mangasarian (2001). RSVM: Reduced support vector machines. Proceedings of the First SIAM International Conference on Data Mining, Chicago. [ftp://ftp.cs.wise.edu/pub/dmi/tech-reports/00-07.ps.]

[15] J. M. Peng (1997). Equivalence of variational inequality problems to unconstrained optimization, *Mathematical Programmimg*, **78**, 347–356.

[16] J. Platt, N. Cristianini and J. Shawe-Taylor (2000). Large margin DAGs for multiclass classification. in: S. A. Solla, T. K. Leen and K. -R. Müller (Eds.), *Advances in Neural Information Processing Systems*, **12**, 547–553. MIT Press, Cambridge, MA.

[17] H. D. Qi (1999). Tikhonov regularization methods for variational inequality problems. *Journal of Optimization Theory and Applications*, **102**, 193–201.

[18] P.K. Subramanian (1988). A note on least two norm solutions of monotone complementarity problems. *Applied Mathematics Letters*, **1**, 395–397.

[19] R. Sznajder and M. S. Gowda (1998). On the limiting behavior of the trajectory of regularized solutions of a P_0-complementarity problem. in: M. Fukushima and L. Qi (Eds.), *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp. 371–379. Kluwer Academic Publishers.

[20] V. Vapnik (1998). *Statistical Learning Theory.* Wiley, New York.

[21] J. Weston and C. Watkins (1998). Multi-class support vector machines. CSD-TR-98-04 Royal Holloway, University of London, Egham, UK.

[22] N. Yamashita, K. Taji and M. Fukushima (1997). Unconstrained optimization reformulations of variational inequality problems. *Journal of Optimization Theory and Applications.* **92**, 439–456.

[23] P. Zhong and M. Fukushima (2004). A new multi-class support vector algorithm. *Optimization Methods and Software*, to appear.