Digital Signal Processing 20 (2010) 944–955

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Complex-valued support vector classifiers

Li Zhang *, Weida Zhou, Licheng Jiao

Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, PR China

ARTICLE INFO

Article history:
Available online 16 September 2009

Keywords:
Complex-valued classification
Support vector machines
Quadrature amplitude modulation
Digital equalization
Mercer kernel

ABSTRACT

In this paper, we present a family of complex-valued support vector classifiers (CSVCs) based on the definition of the complex sign inspired by the modulation in digital communications and the complex-valued kernel functions. We also propose a theorem to construct the complex-valued Mercer kernels and the corresponding kernel function groups. CSVC algorithms include binary (2-state) CSVC (BCSVC), quadrature (4-state) CSVC (QCSVC) and some multi-state CSVCs. In this paper, we focus on QCSVC. For a quadrature complex-valued classification problem, QCSVC is identical to the 4-quadrature amplitude modulation demodulation methods in digital communications. Finally, the simulated experiments confirm the validity and the efficiency of CSVCs.

1. Introduction

Support vector machines (SVMs) based on statistical learning theory have a good generalization performance for obtaining a right balance between the empirical risk and the machine capacity by minimizing the structural risk [1–5]. The Mercer kernel technique is used to construct nonlinear SVMs. SVMs have some other striking properties. The training procedure of SVMs amounts to solving a linearly constrained quadratic optimization problem; the solution of SVMs is a unique global extremum. The complexity of an SVM is independent of the dimension of the input space, and depends only on its number of support vectors. Recently, SVMs have received considerable attention in many fields such as machine learning, signal processing [6,7], data mining and others.

The data for applications of interest here is generally complex. Since it is very convenient and effective for complex variables to express the phase and the frequency of signals, they are usually adopted to represent signals in electrical engineering, communication engineering and others [8–10]. These signals include communication signals, array signals, time-frequency signals and transformed signals generated by the Fourier transform, discrete cosine transform, Gabor transform and others.

In order to process complex-valued signals, many methods are available for filtering, denoising, classification and so on. A series of artificial neural networks (ANNs) with complex-valued models and learning rules were developed from the 1990s in [11–17,32,33]. Leung [11] proposed a complex-valued back propagation (BP) learning algorithm to implement complex pattern recognition and complex regression estimation. Chen [12] and Benvenuto [13] studied a complex-valued radial basis function (RBF) network and its learning algorithm. Cartro [14] and Pomponi [15] presented a complex-valued Hebbian learning rule for extracting the principle and the independent components of complex-valued samples. These methods have been well applied to pattern recognition, series estimation and forecasting, channel equalization, blind source separation and so on. Jankowski [16,17] suggested a complex-valued Hopfield network to implement denoising, memory and other functions for gray images by an N-state quantification of complex phase.

* Corresponding author.
E-mail address: zhangli@mail.xidian.edu.cn (L. Zhang).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2009.09.005

The goal of this paper is to extend the application of SVMs into complex-valued classification problems. A support vector classifier (SVC) is a kind of SVM algorithms for classification. Since the solution of an SVC adopts a numerical computation method, they cannot be simply generalized to the case of the complex variables, which form a non-ordered field. Firstly, the phase modulation methods in communication inspired us to define N-state complex-valued sign function. Then we introduce complex-valued kernel functions and propose a theorem on the construction of complex-valued kernel functions. For the 4-state complex-valued classification problem, QCSVC is identical to the demodulation of 4-QAM in digital communications. We have paid more attention to QCSVC.

The structure of this paper is arranged as follows. Section 2 discusses the complex-valued sign functions. The complex-valued kernel functions and the corresponding kernel function groups are also introduced in Section 2. In Section 3, we propose QCSVC. The experiments on the 4-QAM equalizer are presented in Section 4. Finally, some brief discussion and conclusion are given in Section 5.

2. Complex-valued sign and kernel functions

This section consists of two parts: the complex-valued sign functions and the complex-valued kernel functions.

2.1. Complex-valued sign functions

The complex-valued sign functions are not only limited to a binary real-valued sign function. According to the technique of phase modulation in digital communications, we define the following N-state complex-valued sign function on the complex plane:

$$y = \operatorname{csign}_N\big(f(x)\big) \stackrel{\text{def}}{=} \begin{cases} z_0 \exp(i0\varphi) & \varphi_0 - \varphi \leqslant \arg(f(x)) < \varphi_0 \\ \vdots & \vdots \\ z_0 \exp(im\varphi) & \varphi_0 + (m-1)\varphi \leqslant \arg(f(x)) < \varphi_0 + m\varphi \\ \vdots & \vdots \\ z_0 \exp(i(N-1)\varphi) & \varphi_0 + (N-2)\varphi \leqslant \arg(f(x)) < \varphi_0 + (N-1)\varphi \end{cases} \tag{2.1}$$

where the sign factor $z_0 = \|z_0\| \exp(i\theta)$, $\theta \in [0, \varphi]$, $i = \sqrt{-1}$, the number of the states is $N \in \mathbf{N}$, an initial phase $\varphi_0 \in [0, \varphi]$ can be chosen arbitrarily and $\varphi = 2\pi/N$. The above complex-valued sign function can be regarded as dividing the complex plane or quantifying the complex-valued phase into N parts uniformly, giving an identifier $z_0 \exp(ik\varphi)$ for every region or class on the divided or the discretized complex plane. The definition of similar complex-valued sign functions can be found in [16]. In the following, we show three examples of complex-valued sign functions in detail. Let csign denote complex-valued sign functions:

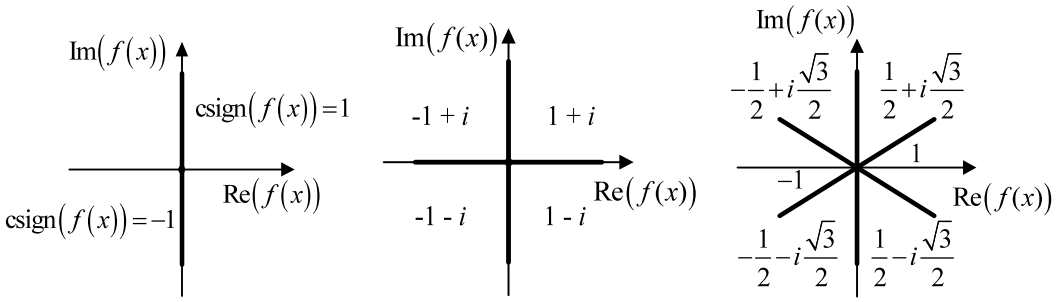
- Binary (2-state) complex-valued sign function
  Let $N = 2$, $\varphi_0 = \pi/2$ and $z_0 = 1$ in (2.1). Thus we can obtain a binary complex-valued sign function

$$y = \operatorname{csign}_2\big(f(x)\big) = \begin{cases} +1, & -\frac{\pi}{2} \leqslant \arg(f(x)) < \frac{\pi}{2} \\ -1, & \frac{\pi}{2} \leqslant \arg(f(x)) < \frac{3\pi}{2} \end{cases} \tag{2.2}$$

- Quadrature (4-state) complex-valued sign function
  Let $N = 4$, $\varphi_0 = \frac{\pi}{2}$ and $z_0 = \sqrt{2}\exp(i\pi/4) = 1 + i$ in (2.1). We have a quadrature complex-valued sign function:

$$y = \operatorname{csign}_4\big(f(x)\big) = \begin{cases} +1+i, & 0 \leqslant \arg(f(x)) < \frac{\pi}{2} \\ -1+i, & \frac{\pi}{2} \leqslant \arg(f(x)) < \pi \\ -1-i, & \pi \leqslant \arg(f(x)) < \frac{3\pi}{2} \\ +1-i, & \frac{3\pi}{2} \leqslant \arg(f(x)) < 2\pi \end{cases} \tag{2.3}$$

- 6-state complex-valued sign function
  Let $N = 6$, $\varphi_0 = \pi/6$ and $z_0 = 1$ in (2.1). We obtain a 6-state complex-valued sign function:



(a) $y = \operatorname{csign}_2(f(x))$ (b) (b) $y = \operatorname{csign}_4(f(x))$ (c) $y = \operatorname{csign}_6(f(x))$

Fig. 2.1. Complex-valued sign functions: (a) binary, (b) quadrature and (c) 6-state complex-valued sign function. The black bold lines are boundaries.

$$y = \operatorname{csign}_6\big(f(x)\big) = \begin{cases} +1 & -\frac{\pi}{6} \leqslant \arg(f(x)) < \frac{\pi}{6} \\ \frac{1}{2} + i\frac{\sqrt{3}}{2} & \frac{\pi}{6} \leqslant \arg(f(x)) < \frac{3\pi}{6} \\ -\frac{1}{2} + i\frac{\sqrt{3}}{2} & \frac{3\pi}{6} \leqslant \arg(f(x)) < \frac{5\pi}{6} \\ -1 & \frac{5\pi}{6} \leqslant \arg(f(x)) < \frac{7\pi}{6} \\ -\frac{1}{2} - i\frac{\sqrt{3}}{2} & \frac{7\pi}{6} \leqslant \arg(f(x)) < \frac{9\pi}{6} \\ +\frac{1}{2} - i\frac{\sqrt{3}}{2} & \frac{9\pi}{6} \leqslant \arg(f(x)) < \frac{11\pi}{6} \end{cases} \tag{2.4}$$

In Fig. 2.1, we plot the three complex-valued sign functions on the complex plane. The binary complex-valued sign function (2.2) reduces to a real-valued sign function. The quadrature complex-valued sign function (2.3) is consistent with 4-ary QAM in digital communications. Hence the complex-valued sign function defined in this paper can be applied to complex-valued signal detection and classification.

2.2. Complex-valued kernel function

Firstly, we describe Mercer's condition [21,22]. A function is called a positive Hilbert–Schmidt kernel (or Mercer kernel) if and only if Mercer's condition holds

$$\iint\limits_{L_2 \otimes L_2} k(\mathbf{z}, \mathbf{z}^*) g(\mathbf{z}) g(\mathbf{z}^*)\, d\mathbf{z}\, d\mathbf{z}^* \geqslant 0 \tag{2.5}$$

where $\mathbf{z} \in \mathcal{Z} \subset \mathbf{R}^d$ ($\mathcal{Z}$ is an input space), $g(\mathbf{z}) \in L_2(\mathbf{R}^d)$ and $k(\mathbf{z}, \mathbf{z}^*) \in L_2(\mathbf{R}^d) \otimes L_2(\mathbf{R}^d)$. A Mercer kernel can be expressed as $k(\mathbf{z}, \mathbf{z}^*) = \Phi^T(\mathbf{z})\Phi(\mathbf{z}^*)$, where $\Phi(\mathbf{z})$ is a nonlinear mapping function, or $\Phi : \mathcal{Z} \mapsto \mathcal{F}$ where $\mathcal{F}$ is some high-dimensional feature space.

Mercer's condition also applies to the case of the complex-valued kernel functions [21,22]. Notice that the order of the kernel function arguments is now important, and they cannot be interchanged as readily as in the real case. As we know, many real-valued kernel functions are already presented [23]. These kernel functions have been successfully used in the real-valued SVCs. But these kernel functions cannot be directly used in our algorithm.

Consider a complex-valued Mercer kernel $k(\mathbf{x}, \mathbf{y})$. According to Mercer's condition, there exists a mapping $\mathbf{v}$ which can map the data in $\mathbf{C}^d$ to a high-dimensional complex Euclidean space $\mathcal{F}$. Namely $\mathbf{x} \xrightarrow{\mathbf{v}} \mathbf{v}(\mathbf{x}) = \mathbf{v}_{\mathrm{R}}(\mathbf{x}) + i\mathbf{v}_{\mathrm{I}}(\mathbf{x})$. Thus the complex-valued kernel function can be expanded as

$$k(\mathbf{x}, \mathbf{y}) = \big[\mathbf{v}_{\mathrm{R}}(\mathbf{x}) + i\mathbf{v}_{\mathrm{I}}(\mathbf{x})\big]^H \big[\mathbf{v}_{\mathrm{R}}(\mathbf{y}) + i\mathbf{v}_{\mathrm{I}}(\mathbf{y})\big] = \big[k_{\mathrm{RR}}(\mathbf{x}, \mathbf{y}) + k_{\mathrm{II}}(\mathbf{x}, \mathbf{y})\big] + i\big[k_{\mathrm{RI}}(\mathbf{x}, \mathbf{y}) - k_{\mathrm{IR}}(\mathbf{x}, \mathbf{y})\big] \tag{2.6}$$

where

$$k_{\mathrm{RR}}(\mathbf{x}, \mathbf{y}) = \mathbf{v}_{\mathrm{R}}^T(\mathbf{x})\mathbf{v}_{\mathrm{R}}(\mathbf{y}), \quad k_{\mathrm{RI}}(\mathbf{x}, \mathbf{y}) = \mathbf{v}_{\mathrm{R}}^T(\mathbf{x})\mathbf{v}_{\mathrm{I}}(\mathbf{y}), \quad k_{\mathrm{IR}}(\mathbf{x}, \mathbf{y}) = \mathbf{v}_{\mathrm{I}}^T(\mathbf{x})\mathbf{v}_{\mathrm{R}}(\mathbf{y}), \quad k_{\mathrm{II}}(\mathbf{x}, \mathbf{y}) = \mathbf{v}_{\mathrm{I}}^T(\mathbf{x})\mathbf{v}_{\mathrm{I}}(\mathbf{y}) \tag{2.7}$$

which are defined as a kernel function group with respect to the complex-valued kernel function. In order to solve the nonlinear CSVC, the kernel function group is required.

Theorem 2.1. *Given data* $\mathbf{x}, \mathbf{y} \in \mathbf{C}^d$. *Assume a complex-valued kernel function* $k(\mathbf{x}, \mathbf{y}) = [k_{\mathrm{RR}}(\mathbf{x}, \mathbf{y}) + k_{\mathrm{II}}(\mathbf{x}, \mathbf{y})] + i[k_{\mathrm{RI}}(\mathbf{x}, \mathbf{y}) - k_{\mathrm{IR}}(\mathbf{x}, \mathbf{y})]$ *and the corresponding kernel function group* $\{k_{\mathrm{RR}}(\mathbf{x}, \mathbf{y}) \in \mathbf{R}, k_{\mathrm{RI}}(\mathbf{x}, \mathbf{y}) \in \mathbf{R}, k_{\mathrm{IR}}(\mathbf{x}, \mathbf{y}) \in \mathbf{R}, k_{\mathrm{II}}(\mathbf{x}, \mathbf{y}) \in \mathbf{R}\}$ *is defined in* (2.7). *If and only if* $k_{\mathrm{RR}}(\mathbf{x}, \mathbf{y})$ *and* $k_{\mathrm{II}}(\mathbf{x}, \mathbf{y})$ *satisfy Mercer's condition, then* $k(\mathbf{x}, \mathbf{y})$ *satisfies Mercer's condition.*

Notice that $k_{\mathrm{RI}}(\mathbf{x},\mathbf{y})$ and $k_{\mathrm{IR}}(\mathbf{x},\mathbf{y})$ depend on $k_{\mathrm{RR}}(\mathbf{x},\mathbf{y})$ and $k_{\mathrm{II}}(\mathbf{x},\mathbf{y})$. Now we can construct the complex-valued kernel function and the corresponding kernel function group. Some examples are given in the following. Other complex-valued kernel functions can be constructed in the similar way:

- Complex-valued linear kernel function $k(\mathbf{x},\mathbf{y}) = \mathbf{x}^H\mathbf{y}$, the corresponding linear kernel function group

$$k_{\mathrm{RR}}(\mathbf{x},\mathbf{y}) = \big((\mathbf{x}_{\mathrm{R}})^T(\mathbf{y}_{\mathrm{R}})\big),\quad k_{\mathrm{RI}}(\mathbf{x},\mathbf{y}) = \big((\mathbf{x}_{\mathrm{R}})^T(\mathbf{y}_{\mathrm{I}})\big),\quad k_{\mathrm{IR}}(\mathbf{x},\mathbf{y}) = \big((\mathbf{x}_{\mathrm{I}})^T(\mathbf{y}_{\mathrm{R}})\big),\quad k_{\mathrm{II}}(\mathbf{x},\mathbf{y}) = \big((\mathbf{x}_{\mathrm{I}})^T(\mathbf{y}_{\mathrm{I}})\big) \tag{2.8}$$

- Complex-valued Gaussian RBF kernel function

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{(\mathbf{x}-\mathbf{y})^H(\mathbf{x}-\mathbf{y})}{2p^2}\right) \tag{2.9}$$

whose output is real. The corresponding kernel function group can be expressed as

$$k_{\mathrm{RR}}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\exp\left(-\frac{(\mathbf{x}-\mathbf{y})^H(\mathbf{x}-\mathbf{y})}{2p^2}\right),\quad k_{\mathrm{RI}}(\mathbf{x},\mathbf{y}) = k_{\mathrm{IR}}(\mathbf{x},\mathbf{y}) = k_{\mathrm{II}}(\mathbf{x},\mathbf{y}) = k_{\mathrm{RR}}(\mathbf{x},\mathbf{y}) \tag{2.10}$$

- Complex-valued Gaussian window Gabor kernel function.

A Gaussian window Gabor kernel function can be written as

$$k(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y})\exp\big(-i\mathbf{w}_0^H(\mathbf{x}-\mathbf{y})\big) \tag{2.11}$$

where $g(\mathbf{x},\mathbf{y})$ is the normalized Gaussian window function that can be expressed as

$$g(\mathbf{x},\mathbf{y}) = \frac{1}{\sqrt{2\pi}\,p}\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2p^2}\right)$$

In signal processing, $\mathbf{w}_0$ and $\mathbf{x}$ denote frequency and time, respectively. Here let $\mathbf{w}_0, \mathbf{x} \in \mathbf{R}^d$. The corresponding Gaussian window Gabor kernel function group can be expressed as

$$k_{\mathrm{RR}}(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y})\cos\big(\mathbf{w}_0^H\mathbf{x}\big)\cos\big(\mathbf{w}_0^H\mathbf{y}\big),\quad k_{\mathrm{RI}}(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y})\cos\big(\mathbf{w}_0^H\mathbf{x}\big)\sin\big(\mathbf{w}_0^H\mathbf{y}\big)$$
$$k_{\mathrm{IR}}(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y})\sin\big(\mathbf{w}_0^H\mathbf{x}\big)\cos\big(\mathbf{w}_0^H\mathbf{y}\big),\quad k_{\mathrm{II}}(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y})\sin\big(\mathbf{w}_0^H\mathbf{x}\big)\sin\big(\mathbf{w}_0^H\mathbf{y}\big)$$

Since the Gaussian window function $g(\mathbf{x},\mathbf{y})$ satisfies Mercer's condition, it is easy to prove $k_{\mathrm{RR}}(\mathbf{x},\mathbf{y})$ and $k_{\mathrm{II}}(\mathbf{x},\mathbf{y})$ satisfy Mercer's condition. So, the complex-valued Gaussian window Gabor kernel function also satisfies Mercer's condition.

3. Complex-valued support vector classifier (CSVC)

For the binary complex-valued classification problem, BCSVC is identical to the traditional binary SVC in which complex-valued samples in $\mathbf{C}^d$ are transformed to real-valued samples in $\mathbf{R}^{2d}$ called double dimension samples. In other words, BCSVC and the traditional binary SVC have the same formulas. The multi-state complex-valued classification problem can be solved by using the methods of regression estimation. Here, we only consider the 4-state complex-valued classification problem.

The linear quadrature complex-valued classification problem can be described as follows. Given a set of independent and identically distributed (i.i.d.) complex-valued samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x} = \mathbf{x}_{\mathrm{R}} + i\mathbf{x}_{\mathrm{I}} \in \mathbf{C}^d$ and $y = y_{\mathrm{R}} + iy_{\mathrm{I}} \in \{\pm 1 \pm i\}$. The quadrature classification problem is to separate these complex-valued samples into four groups by selecting a right function from the set of linear functions $\{f(\mathbf{x}) = \mathbf{w}^H\mathbf{x} + b, \mathbf{w} \in \mathbf{C}^d, b \in \mathbf{C}\}$. This problem is identical to the one of the demodulation of QPSK or the 4-QAM in digital communications. For this problem, we need adopt the 4-state complex-valued sign function csign_4 (2.3). Now we introduce a structural risk similar to the real-valued SVC, and have

PQP:

$$\min \frac{1}{2}\mathbf{w}^H\mathbf{w} + \lambda\cdot\sum_{j=1}^{l}\big[\mathrm{Re}(\eta_j) + \mathrm{Im}(\eta_j)\big] \tag{3.1}$$
$$\text{subject to}\quad \mathrm{Re}(y_j)\,\mathrm{Re}\big[f(\mathbf{x}_j)\big] \geqslant 1 - \mathrm{Re}(\eta_j)$$
$$\mathrm{Im}(y_j)\,\mathrm{Im}\big[f(\mathbf{x}_j)\big] \geqslant 1 - \mathrm{Im}(\eta_j)$$
$$\mathrm{Re}(\eta_j) \geqslant 0,\quad \mathrm{Im}(\eta_j) \geqslant 0,\quad j = 1,\ldots,l$$

where $\lambda \in \mathbf{R}$ is a penalty factor, $\eta_j \in \mathbf{C}$, $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ denote the operation for the real and the imaginary part, respectively. The derivation of the Wolfe dual programming of PQP is omitted for the limit of space. By using Lagrange multipliers techniques, we obtain the Wolfe dual programming of PQP:

DQP:

$$\min \frac{1}{2}\begin{bmatrix} \mathbf{a}^T(H_{\mathrm{RR}}+H_{\mathrm{II}})\mathbf{a}+\mathbf{a}^T(H_{\mathrm{RI}}-H_{\mathrm{IR}})\mathbf{b} \\ +\mathbf{b}^T(H_{\mathrm{IR}}-H_{\mathrm{RI}})\mathbf{a}+\mathbf{b}^T(H_{\mathrm{RR}}+H_{\mathrm{II}})\mathbf{b} \end{bmatrix} - \sum_{j=1}^{l}\big((\alpha_{\mathrm{R}})_j+(\alpha_{\mathrm{I}})_j\big) \tag{3.2}$$

$$\text{subject to} \quad \mathbf{a}^T\mathbf{1}=0, \quad \mathbf{b}^T\mathbf{1}=0$$
$$0\leqslant(\alpha_{\mathrm{R}})_j\leqslant\lambda, \quad 0\leqslant(\alpha_{\mathrm{I}})_j\leqslant\lambda, \quad j=1,\ldots,l$$

where $\mathbf{a}=[(\alpha_{\mathrm{R}})_1(y_{\mathrm{R}})_1,\ldots,(\alpha_{\mathrm{R}})_l(y_{\mathrm{R}})_l]^T$, $\mathbf{b}=[(\alpha_{\mathrm{I}})_1(y_{\mathrm{I}})_1,\ldots,(\alpha_{\mathrm{I}})_l(y_{\mathrm{I}})_l]^T$, $(\alpha_{\mathrm{R}})_j$, $(\alpha_{\mathrm{I}})_j$, $j=1,\ldots,l$ are the Lagrange multipliers, $\mathbf{1}$ denotes the vector of all ones, and matrices

$$H_{\mathrm{RR}}=\big((\mathbf{x}_{\mathrm{R}})_m^T(\mathbf{x}_{\mathrm{R}})_j\big)_{m,j=1,\ldots,l}, \quad H_{\mathrm{RI}}=\big((\mathbf{x}_{\mathrm{R}})_m^T(\mathbf{x}_{\mathrm{I}})_j\big)_{m,j=1,\ldots,l}$$
$$H_{\mathrm{IR}}=\big((\mathbf{x}_{\mathrm{I}})_m^T(\mathbf{x}_{\mathrm{R}})_j\big)_{m,j=1,\ldots,l}, \quad H_{\mathrm{II}}=\big((\mathbf{x}_{\mathrm{I}})_m^T(\mathbf{x}_{\mathrm{I}})_j\big)_{m,j=1,\ldots,l} \tag{3.3}$$

The optimal weight vector $\mathbf{w}=\mathbf{w}_{\mathrm{R}}+i\mathbf{w}_{\mathrm{I}}=\sum_{j=1}^{l}(\alpha_j\circ y_j)^H\mathbf{x}_j$, $(x\circ y)\overset{\mathrm{def}}{=}\mathrm{Re}(x)\,\mathrm{Re}(y)+i\,\mathrm{Im}(x)\,\mathrm{Im}(y)$, $\mathbf{w}_{\mathrm{R}}=\sum_{j=1}^{l}(\alpha_{\mathrm{R}})_j(y_{\mathrm{R}})_j\times(\mathbf{x}_{\mathrm{R}})_j+\sum_{j=1}^{l}(\alpha_{\mathrm{I}})_j(y_{\mathrm{I}})_j(\mathbf{x}_{\mathrm{I}})_j$ and $\mathbf{w}_{\mathrm{I}}=\sum_{j=1}^{l}(\alpha_{\mathrm{R}})_j(y_{\mathrm{R}})_j(\mathbf{x}_{\mathrm{I}})_j-\sum_{j=1}^{l}(\alpha_{\mathrm{I}})_j(y_{\mathrm{I}})_j(\mathbf{x}_{\mathrm{R}})_j$. The threshold b can be computed by KKT conditions. Let $\mathbf{sv}_{\mathrm{R}}=\{j|0<(\alpha_{\mathrm{R}})_j<\lambda,\ j=1,\ldots,l\}$ and $\mathbf{sv}_{\mathrm{I}}=\{j|0<(\alpha_{\mathrm{I}})_j<\lambda,\ j=1,\ldots,l\}$ be the index sets of support vectors. Then $b=b_{\mathrm{R}}+ib_{\mathrm{I}}$, where $b_{\mathrm{R}}=\sum_{j\in\mathbf{sv}_{\mathrm{R}}}((y_{\mathrm{R}})_j-\mathbf{w}_{\mathrm{R}}^T(\mathbf{x}_{\mathrm{R}})_j-\mathbf{w}_{\mathrm{I}}^T(\mathbf{x}_{\mathrm{I}})_j)/|\mathbf{sv}_{\mathrm{R}}|$ and $b_{\mathrm{I}}=\sum_{j\in\mathbf{sv}_{\mathrm{I}}}((y_{\mathrm{I}})_j-\mathbf{w}_{\mathrm{R}}^T(\mathbf{x}_{\mathrm{I}})_j+\mathbf{w}_{\mathrm{I}}^T(\mathbf{x}_{\mathrm{R}})_j)/|\mathbf{sv}_{\mathrm{I}}|$.

The decision function can be expressed as

$$y=\mathrm{csign}_4\big[f(\mathbf{x})\big]=\mathrm{csign}_4\left[\sum_{j=1}^{l}(\alpha_j\circ y_j)\big(\mathbf{x}_j^H\mathbf{x}\big)+b\right] \tag{3.4}$$

Substituting

$$H_{\mathrm{RR}}=\big(k_{\mathrm{RR}}(\mathbf{x}_m,\mathbf{x}_j)\big)_{m,j=1,\ldots,l}, \quad H_{\mathrm{RI}}=\big(k_{\mathrm{RI}}(\mathbf{x}_m,\mathbf{x}_j)\big)_{m,j=1,\ldots,l}$$
$$H_{\mathrm{IR}}=\big(k_{\mathrm{IR}}(\mathbf{x}_m,\mathbf{x}_j)\big)_{m,j=1,\ldots,l}, \quad H_{\mathrm{II}}=\big(k_{\mathrm{II}}(\mathbf{x}_m,\mathbf{x}_j)\big)_{m,j=1,\ldots,l} \tag{3.5}$$

into (3.2), we can generalize the linear QCSVC to the nonlinear QCSVC. The output function of the nonlinear QCSVC has the form

$$y=\mathrm{csign}_4\big[f(\mathbf{x})\big]=\mathrm{csign}_4\left[\sum_{j=1}^{l}(\alpha_j\circ y_j)k(\mathbf{x}_j,\mathbf{x})+b\right] \tag{3.6}$$

Remark. Comparison of real-valued multi-class SVC and QCSVC.

Certainly the 4-state complex-valued classification problem also can be solved by the real-valued multi-class SVC (MSVC) which has been described in detailed in [18–20], but the complex-valued classification methods are simpler and more efficient than the real-valued MSVC. For the practical problem of the demodulation of 4-QAM, the real-valued MSVC only considers it as a general classification problem and neglects the phase information of 4-QAM, so we think QCSVC will perform better than the real-valued MSVC.

Usually, a multi-class classification problem can be solved by reducing it into multiple binary classification problems. For the real-valued SVC, typical multi-class algorithms include one-against-one and one-against-all algorithms [18–20]. In the following, we only compare the computation complexity of the real-valued MSVC and QCSVC. The comparison of performance of them will be shown in experiments.

Let $O_{\mathrm{train}}(m,n)$ express the complexity of a linear constraint quadratic programming, where m is the number of variables and n is the number of linear constraints. Then the training complexity of QCSVC is $O_{\mathrm{train}}(2l,4l+2)$, where l is the number of training examples. Suppose each class has $l/4$ examples. Hsu and Lin [20] indicated that the training complexity of the one-against-one quadrature SVC is $6O_{\mathrm{train}}(l/2,l+1)$, that of the one-against-all quadrature SVC is $4O_{\mathrm{train}}(l,2l+1)$ and that of the hybrid real quadrature SVC [18] is $O_{\mathrm{train}}(3l,3l+4)$.

In addition let $O_{\mathrm{test}}(m)$ denote the testing complexity of the traditional SVC for binary classification problems, where m is the number of training examples. The testing complexity of QCSVC is $O_{\mathrm{test}}(2l)$. In the case of 4 classes, the testing complexities of one-against-one, one-against-all and hybrid quadrature SVC are $6O_{\mathrm{test}}(l/2)$, $4O_{\mathrm{test}}(l)$ and $4O_{\mathrm{test}}(l)$, respectively.

4. Simulation

Since the training procedure of QCSVC amounts to solving a real-valued constrained quadratic programming problem, many optimization methods available for solving a real-valued constrained quadratic programming problem can be used for QCSVC. In our experiments, we used the optimization function "quadprog.m" in the optimization toolbox of Matlab 6.1.

4.1. 4-QAM equalization in the finite memory channel

High-speed and broadband data communication techniques have become one of the important research topics with the development of Internet. However, channels — especially wireless ones — cause the transferred signal to have time-delay spread, which makes the signal overlap with its delay signals. It is known as inter-symbol interference (ISI) in digital communication. In addition, the desired signal is also interfered and corrupted by other factors including the other users' signals, thermal noise, impulse noise, the nature of channel itself and others. High symbol-error-rate (SER) and low efficiency caused by the ISI have become a bottleneck of the development of digital communication. The digital equalization methods were mainly proposed for solving the problem of the ISI and decreasing the SER in digital communication. These methods also can be regarded as a pattern recognition problem or a multiple hypothesis test problem. Without loss of generality, in this experiment, we only consider the 4-QAM or QPSK equalization problem in a linear finite-memory channel.

Assume that l_a is the order of channel. We assume that the sequence corrupted by complex-valued additive Gaussian noise and sampled in symbol space can be expressed as

$$r(k) = f_a\big(s(k), s(k-1), \ldots, s(k-l_a+1)\big) + n(k) \tag{4.1}$$

where r is the received sequence, s is the transmitted sequence, and noise $n(k) = n_{\mathrm{R}}(k) + in_{\mathrm{I}}(k), n_{\mathrm{R}}(k) \sim N(0, \sigma_{\mathrm{R}})$ and $n_{\mathrm{I}}(k) \sim N(0, \sigma_{\mathrm{I}})$, generally $\sigma_{\mathrm{R}} = \sigma_{\mathrm{I}}$. The transmitted sequence of 4-QAM is defined by

$$s(k) = s_{\mathrm{R}}(k) + is_{\mathrm{I}}(k) = \{1+i, -1+i, -1-i, 1-i\} \tag{4.2}$$

Moreover, assume that transmitted symbols take values from the constellation (4.2) in an equiprobable manner. In fact, many channels are finite memory linear ones which can be modeled by an FIR filter with a transform $R(z)/S(z) = A(z) = \sum_{j=0}^{l_a-1} h_j z^{-j}$. Thus the received sequence can be expressed as

$$r(k) = \sum_{j=0}^{l_a-1} h_j s(k-j) + n(k) \tag{4.3}$$

which is ostensibly (4.1) in which f_a is a linear function.

For the above 4-QAM signal equalization problem of the finite memory channel, the optimal sequence-estimation equalizer is known as maximum-likelihood sequence estimation (MLSE) [24,25], which is based on the principle of maximum likelihood detection of the entire transmitted symbol sequence. MLSE provides the best attainable performance for any equalizer. But this method requires the channel parameters and has a very large computation complexity. In order to reduce the computation complexity, in practice, MLSE is implemented by Viterbi algorithm with a fixed decision delay and is equipped with an adaptive channel estimation in adaptive application [26,27]. However this algorithm still needs to estimate the channel parameter. In doing so, it makes a large estimation delay, in particular for a time varying channel. Presently, both forward neural networks (FNNs) based on the back propagation (BP) algorithm and RBFNN have a performance approximating to the MLSE needing less delay [28–30]. For the complex-valued signals, complex-valued ANNs have been proposed. For examples, Chen et al. presented a complex-valued RBFNN based on an orthogonal least squares (OLS) algorithm and a K-mean clustering plus least mean square (K-MC + LMS) algorithm [12,28] in 1995. Cha [29] proposed an adaptive complex-valued RBFNN based on the stochastic gradient (CRBF/SG) in 1995. These methods obtained good results in detecting 4-QAM signal. In 2002, Deng [30] presented a complex minimal resource allocation network (MRAN) equalizer. This method can adjust the scale of RBFNN adaptively. Although it has a good detection performance, its training speed is very slow. In this paper, we use QCSVC to solve the 4-QAM signal equalization problem. The equalization scheme is shown in Fig. 4.1.

Example 1. Complex linear channel.

The channel transfer function is defined by

$$A(z) = (0.7409 - i0.7406)\big(1 - (0.2 - i0.1)z^{-1}\big)\big(1 - (0.6 - i0.3)z^{-1}\big)$$

which was adopted in [28–30]. The feedforward order of the equalizer (or the dimension of samples) was chosen as $m = 1$. The decision delay was chosen to be $d = 1$. This linear channel has $4^{l_a+m-1} = 64$ output values under the case of non-noise shown in Fig. 4.2 and its decision boundary is nonlinear. Here and below, in all figures which contain samples, we use crosses, circles, solid dots, and stars denote four classes $+1+i, -1+i, -1-i$ and $+1-i$, respectively. The input sequence is defined by (4.2). Here the signal-to-noise ratio is defined by the following equation

$$\mathrm{SNR} = \log(P_s/P_n) = \log\big((s_{\mathrm{R}}^2 + s_{\mathrm{I}}^2)/(\sigma_{\mathrm{R}}^2 + \sigma_{\mathrm{I}}^2)\big) \tag{4.4}$$

Here we compare our algorithm QCSVC with the one-against-one multi-class support vector classifier (MSVC) [18–20], CRBF/SG [29] and the optimal Bayesian. These methods are used to solve the equalization of 4-QAM. The parameters of each equalizer are set as follows:

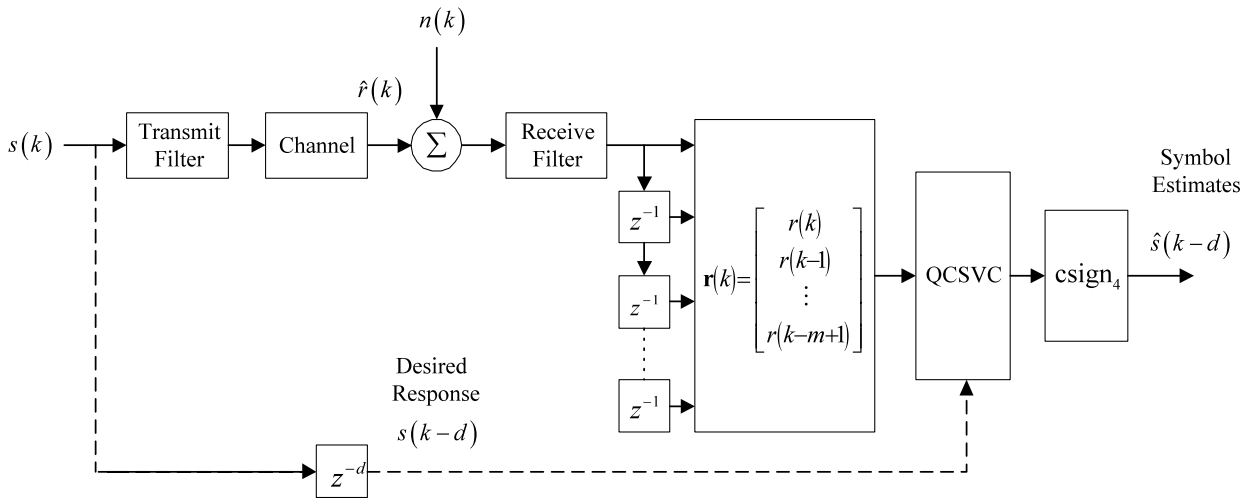


Fig. 4.1. QCSVC equalization scheme.

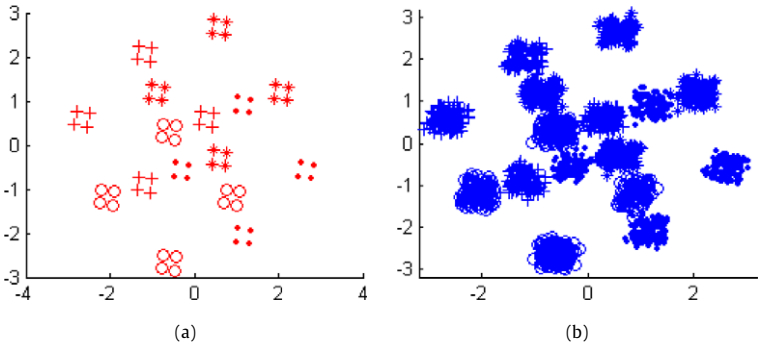


Fig. 4.2. Constellation of the linear channel outputs. (a) Constellation of the channel outputs without noise. There are 64 states. (b) Constellation of 2000 channel outputs for SNR = 20 dB, where crosses, circles, solid dots, and stars denote four classes $+1+i$, $-1+i$, $-1-i$ and $+1-i$, respectively. The abscissa is the real part of the complex-valued samples and the ordinate is the imaginary.

1. QCSVC equalizer: Let $C = 100$. The complex-valued Gaussian RBF kernel is exploited, where its kernel parameter $p = 0.4$. The number of training samples is 200 and that of test ones is 50 000.
2. MSVC equalizer: Let $C = 100$. The real-valued Gaussian RBF kernel is used whose parameter $p = 0.3$. The number of training samples is 200 and that of test ones is 10 000.
3. CRBF/SG equalizer: The parameters of CRBF/SG equalizer are identical to [29] with 16 000 training samples, 50 000 test ones and 50 hidden nodes. The learning rate of the weight, the center and the variance are 0.05, 0.05 and 0.03, respectively. Selecting the learning rate affects learning speed. The initial values of the weight, the center and the variance are chosen randomly.
4. Bayesian equalizer: Assume that the parameters of the channel are given. The number of test samples is 50 000.

The experimental results are shown in Figs. 4.3–4.4. Fig. 4.3 shows the comparison of the boundaries obtained by QCSVC, MSVC, CRBF/SG and the Bayesian equalizer for SNR = 20 dB. From Fig. 4.3 we can see that both QCSVC and MSVC maximize the margin. Roughly, the boundaries obtained by them pass through the middle of every two class samples, and these boundaries are closer to the one obtained by the optimal Bayesian equalizer than that obtained by CRBF/SG.

In Fig. 4.4, there is shown the performance comparison of four equalizers: QCSVC, CRBF/SG, MSVC and the optimal Bayesian for SNR from 2 dB to 26 dB with an interval 2 dB. The histogram of the variances of 30 SERs versus SNR for QCSVC and CRBF/SG equalizer is shown in Fig. 4.4(b). Fig. 4.4(c) shows the number of SVs versus SNR for QCSVC and MSVC equalizer. From Fig. 4.4, we can see that if SNR is large (say >16 dB), the SER and its variance of QCSVC is lower than that of CRBF/SG, and the SER curve of QCSVC is closer to that of the optimal Bayesian equalizer than CRBF/SG is, which shows our method has higher detection precision and robustness. In the low SNR case (<16 dB) SER and its variance of QCSVC equalizer is larger than that of CRBF/SG. In our opinion, there are many cross and overlapped samples in the low SNR case. So in order to improve the performance, it needs more training samples. Note that the number of training samples of QCSVC is only 200 (much less than 16 000 of CRBF/SG), and that there are 64 output states of samples for this linear channel without noise. It just shows that SVC has good generalization performance. In the high SNR case, the number

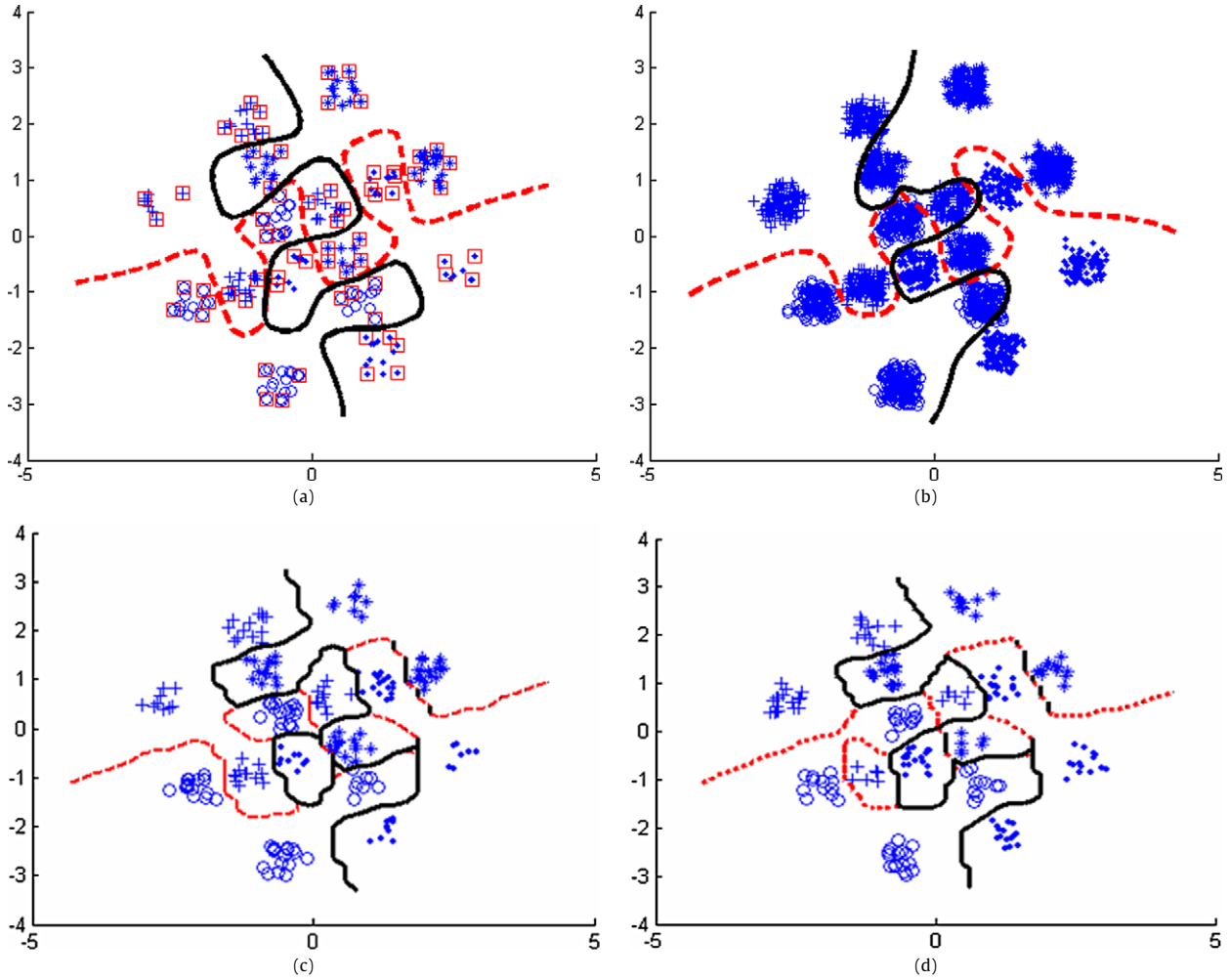


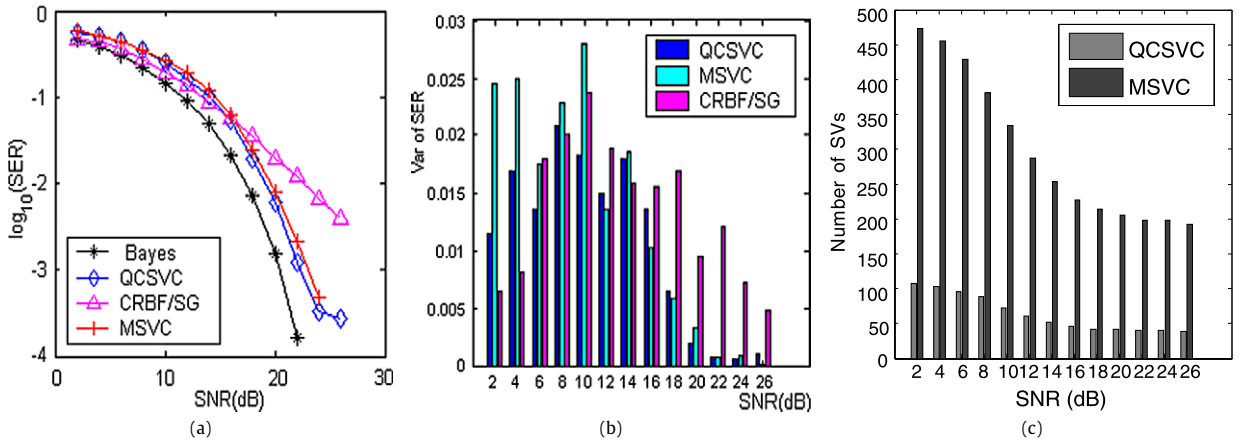
Fig. 4.3. Performance comparison of QCSVC, CRBF/SG, the Bayesian equalizer and MSVC for SNR = 20 dB. Training data points for each class are represented as crosses, circles, solid dots and stars, respectively. (a) Decision areas obtained by QCSVC. There are 200 training samples among which support vectors are indicated by the extra squares. (b) Decision areas obtained by CRBF/SG with the last 2000 training samples instead of all 16 000 ones. Decision areas were obtained by the optimal Bayesian equalizer (c) and MSVC (d), respectively. All 200 training samples in MSVC are SVs.

of the cross or overlapped samples is small. So it only needs smaller training samples for QCSVC to implement a better classification. In addition, in the high SNR case the SER variance of an equalizer mainly depends on the performance and the robustness of the equalizer. Since there is no local minimum in QCSVC, the variance of SER is small. CRBF/SG has many local minimums, which leads to a large SER variance. The Bayesian equalizer has the best performance. QCSVC and MSVC have a similar performance. But QCSVC has a faster training and test speed than MSVC does. The training complexity of QCSVC is denoted by $C_{\mathrm{QP}}(2l, 4l+2)$ which means the training complexity of a quadratic programming with $2l$ variables and $4l+2$ linear constraints where l is the number of training samples. The training complexity of the one-against-one MSVC can be expressed as $m(m-1)C_{\mathrm{QP}}(l/2, l+1)/2 = 6C_{\mathrm{QP}}(l/2, l+1)$, where the number of classes is $m=4$ for the 4-QAM equalizer.

Observation on Fig. 4(c) reveals that the number of support vectors obtained by QCSVC is much smaller than that obtained by MSVC. For quadrature complex-valued classification problems, we only needs one QCSVC, but 6 SVCs in a MSVC. If a training sample is SVs in all 6 SVCs, it would appear in 6 decision functions and is taken as 6 different SVs. In this meaning, the number of SVs in MSVC is generally more than that of QCSVC, which results MSVC has a slower test speed than QCSVC does. In order to further improve test speed, both QCSVC and MSVC can adopt the method proposed by Burges [31] to reduce the number of support vectors. CRBF/SG has the worst performance among these equalizers.

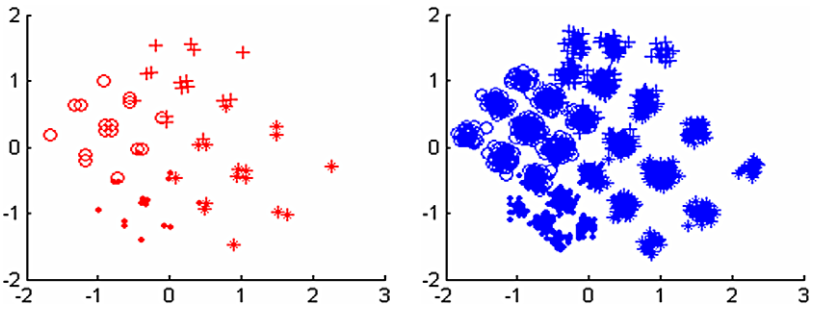
Example 2. Complex nonlinear channel.

The nonlinear channel transfer function is given by



(a) (b) (c)

Fig. 4.4. Performance comparison of the optimal Bayesian, QCSVC, CRBF/SG and MSVC equalizer. SNR takes values from 2 dB to 26 dB with an interval 2 dB. (a) SER versus SNR for the optimal Bayesian, QCSVC, CRBF/SG and MSVC equalizer, where the SER curves of the last three equalizers are average results on 30 runs. (b) Histogram of variances of 30 SERs versus SNR for QCSVC, MSVC and CRBF/SG equalizers. (c) Histogram of the number of SVs versus SNR for QCSVC and MSVC equalizers.



(a) (b)

Fig. 4.5. Constellation of the nonlinear channel outputs. (a) Constellation of the first dimension channel outputs without noise. There are 1024 states. (b) Constellation of the first dimension channel outputs of 2000 samples for SNR = 20 dB. The abscissa is the real part of the complex-valued samples and the ordinate is the imaginary.

$$o(k) = (0.34 - i0.27)s(k) + (0.87 + i0.43)s(k-1) + (0.34 - i0.21)s(k-2)$$
$$h(k) = o(k) + 0.1o^2(k) + 0.05o^3(k)$$

This example was also adopted in Refs. [29] and [30] which also used 4-QAM. For the sake of comparison, the symbol sequence is given by

$$s(k) = s_{\mathrm{R}}(k) + is_{\mathrm{I}}(k) \in \{0.7 + i0.7, -0.7 + i0.7, -0.7 - i0.7, 0.7 - i0.7\}$$

which is defined in [29] and [30]. Let feedforward order of the equalizer (the dimension of samples) $m = 3$ and the decision delay $d = 1$. There are $4^{l_a+m-1} = 1024$ states for the constellation of the output of this nonlinear channel. Only the first dimension of these samples is shown in Fig. 4.5. The SNR is defined by

$$\mathrm{SNR} = \log(P_s/P_n) = \log\left(\left(s_{\mathrm{R}}^2 + s_{\mathrm{I}}^2\right)/\left(\sigma_{\mathrm{R}}^2 + \sigma_{\mathrm{I}}^2\right)\right)$$

We also compare our QCSVC with MSVC, CRBF/SG and the optimal Bayesian equalizer. The parameters of each equalizer are set as follows:

1. QCSVC equalizer: Let $C = 100$. The complex-valued RBF kernel is used, where its parameter $p = 1.8$. The number of training samples is 400 and that of test ones is 50 000.
2. MSVC equalizer: Let $C = 100$. The real-valued Gaussian RBF kernel is used, where its parameter $p = 1.8$. The number of training samples is 400 and test 10 000.
3. CRBF/SG equalizer: The parameters for CRBF/SG are identical to [29]: the training sample number is 16 000, the test one is 50 000 and the hidden node number is 30 and the learning rates of the weight, the center and the spread whose initial values were taken randomly are 0.05, 0.05 and 0.03, respectively.
4. Bayesian equalizer: All parameters of the channel are known. The number of test samples is 50 000.

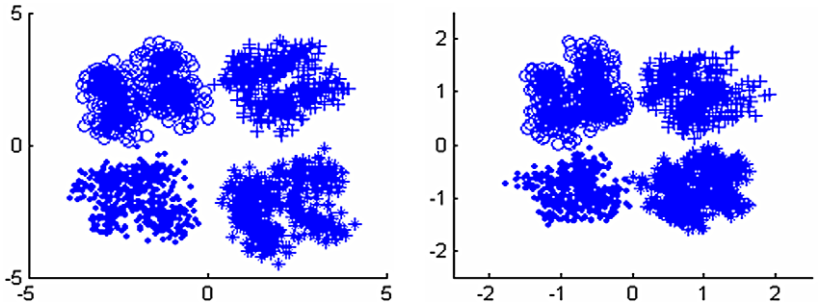


Fig. 4.6. Output distribution obtained by (a) QCSVC and (b) CRBF/SG for SNR = 20 dB. Note that the scale of the coordinate in both figures is different. Since the output of MSVC and the Bayesian equalizer is discrete, their output distribution cannot be plotted here.

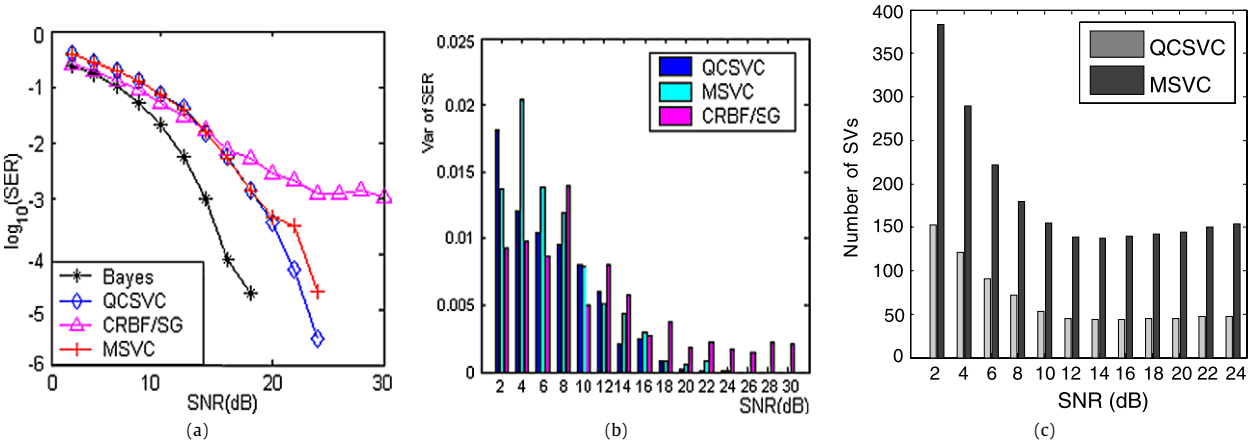


Fig. 4.7. Performance comparison of the optimal Bayesian, QCSVC, CRBF/SG and MSVC equalizer. SNR takes values from 2 dB to 26 dB with an interval 2 dB. (a) SER versus SNR for the optimal Bayesian, QCSVC, CRBF/SG and MSVC equalizer, where the SER curves of the last three equalizers are average results on 30 runs. (b) Histogram of variances of 30 SERs versus SNR for QCSVC, CRBF/SG and MSVC equalizers. (c) Histogram of the number of SVs versus SNR for QCSVC and MSVC equalizers.

The experimental results are shown in Figs. 4.6–4.7. There are shown the output distribution of QCSVC and CRBF/SG for SNR = 20 dB in Figs. 4.6 (a) and (b), respectively. The output distribution of QCSVC is farther from the boundary relative to its variance than that of CRBF/SG, so the performance of QCSVC is better than that of CRBF/SG.

In Figs. 4.7 (a), (b) and (c), there are shown the performance comparison of four equalizers: QCSVC, MSVC, CRBF/SG and the optimal Bayesian equalizer for SNR from 2 dB to 24 dB with an interval 2 dB. The SER versus SNR for QCSVC, CRBF/SG and optimal Bayesian is shown in Fig. 4.7(a), where the results of QCSVC and CRBF/SG equalizer are averaged ones over 30 runs. The histogram of the variances of 30 SERs versus SNR for QCSVC and CRBF/SG equalizer is shown in Fig. 4.7(b). Fig. 4.7(c) shows the number of SVs versus SNR for QCSVC and MSVC equalizer. From Fig. 4.7, we can see that if SNR is large (say >16 dB), the SER and its variance of QCSVC is lower than that of CRBF/SG, and the SER curve of QCSVC is closer to that of the optimal Bayesian equalizer than CRBF/SG is, which shows our method has higher detection precision and robustness. In the low SNR case (<16 dB), namely the samples of every classification overlaps very seriously, SER and its variance of QCSVC equalizer is larger than that of CRBF/SG. It is necessary for QCSVC to take more training samples in the low SNR case. Note that the number of the training samples for QCSVC is only 400 (much less than 16 000 of CRBF/SG), and that there are 1024 output states for this nonlinear channel. The more the output states are, the more the number of training samples is required in order to obtain a high performance. In the high SNR case CRBF/SG has many local minimums, which leads to a large SER variance.

5. Conclusions

Since it is very convenient and effective to express the phase and the frequency with complex variables, they are generally adopted to represent signals in electrical engineering, communication engineering and others. The goal of this paper is to generalize the application of SVCs to the complex samples such that SVMs can process the classification problem of complex-valued samples. Since the solution of SVCs adopts numerical computation method, SVCs cannot be directly generalized to the case of the complex variables. Here CSVCs are proposed. To summarize, our work has three main contributions: (1) We give the definition of N-state complex-valued sign functions on the complex plane. (2) We present a theorem to con-

struct complex-valued kernel functions. (3) We propose CSVCs for complex-valued classification problems, including BCSVC, QCSVC and multi-state CSVCs. For binary complex-valued classification problem, BCSVC is identical to the binary SVC with double dimension samples. Moreover, multi-state complex-valued classification problem could be solved by using regression estimation methods. Thus, we put emphasis on QCSVC and discuss it in detail. Experimental results on the demodulation of the 4-QAM in the finite memory channel show that QCSVC has a good performance. Compared to MSVC, QCSVC has a similar performance, but a faster training and test speed. The comparison with CRBF/SG shows that QCSVC should use more training samples in the low SNR cases, and performs well in the high SNR cases.

An interesting future work is to extend SVMs to complex-valued regression estimation problems which are more general than classification problems in electrical engineering and communication engineering.

Acknowledgments

Authors would like to Journal Manager M. Sangeneni and the two anonymous reviewers for their valuable comments and suggestions, which have helped to significantly improve this paper. In addition authors wish to acknowledge Steve Gunn for his SVMs toolbox. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60602064, 60872135 and 60970067.

References

[1] V. Vapnik, Statistical Learning Theory, Wiley–Interscience Publication, New York, 1998.
[2] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[3] V. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Networks 10 (1999) 988–999.
[4] C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167.
[5] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998. Available at: http://www.kernel-machines.org/papers/tr-30-1998.ps.gz.
[6] D.J. Sebald, Nonlinear signal processing for digital communications using support vector machines and a new form of adaptive decision feedback equalizer, Ph.D. thesis, University of Wisconsin, Madison, 2000.
[7] D. Sebald, J. Bucklew, Support vector machines and the multiple hypothesis test problem, IEEE Trans. Signal Process. 49 (2001) 2865–2872.
[8] B. Sklar, Digital Communications Fundamentals and Applications, 2nd ed., Prentice Hall, New Jersey, 2001.
[9] J.G. Proakis, Digital Communication, 3rd ed., McGraw–Hill, New York, 1995.
[10] S. Haykin, Adaptive Filter Theory, 3rd ed., Prentice Hall, New Jersey, 1996.
[11] H. Leung, S. Haykin, The complex backpropagation algorithm, IEEE Trans. Signal Process. 39 (1991) 2101–2104.
[12] S. Chen, S. McLaughlin, B. Mulgrew, Complex-valued radial basis function network, Part I. Network architecture and learning algorithms, Signal Process. 35 (1994) 19–31.
[13] N. Benvenuto, M. Marchsi, F. Piazza, A. Uncini, A comparison between real and complex-valued neural networks in communication applications, in: Proc. Int. Conf. Artificial Neural Networks (Espoo, Finland), 1991, pp. 1177–1180.
[14] M.C.F.D. Castro, F.C.C.D. Castro, J.N. Amaral, P.R.G. Franco, A complex valued Hebbian learning algorithm, in: 1998 IEEE World Congress on Computational Intelligence, Int. Joint Conf. Neural Networks, Anchorage, Alaska, 1998, pp. 1235–1238.
[15] E. Pomponi, S. Fiori, F. Piazza, Complex independent component analysis by nonlinear generalized Hebbian learning with Rayleigh nonlinearity, in: Proc. Int. Conf. on Acoustics, Speech and Signal Processing, vol. 2, 1999, pp. 1077–1080.
[16] S. Jankowski, A. Lozowski, J. Zurada, Complex-valued multistate neural associative memory, IEEE Trans. Neural Networks 7 (1996) 1491–1496.
[17] S. Jankowski, A. Lozowski, Complex-valued neural networks. In: Proc. XVI Nat. Conf. Circuit Theory Electronics System, Kolobrzeg, Poland, 1993, pp. 582–587.
[18] J. Weston, C. Watkins, Multi-class support vector machines, Technical Report CSD-TR-98-04, Royal Holloway University of London, 1998. Available at: http://www.cs.rhbnc.ac.uk/research/compint/areas/comp_learn/sv/pub/report98-04.ps.gz.
[19] E. Bredensteiner, K. Bennett, Multicategory classification by support vector machines, Comput. Optim. Appl. (1999) 53–79.
[20] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, IEEE Trans. Neural Networks 13 (2002) 415–425.
[21] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, Philos. Trans. Roy. Soc. London A 209 (1909) 415–446.
[22] S. Saitoh, Theory of Reproducing Kernels and its Applications, Longman Scientific and Technical, Harlow, England, 1988.
[23] C. Saunders, M.O. Stitson, J. Weston, L. Bottou, B. Schölkopf, A. Smola, Support vector machine — reference manual, Technical report CSD-TR-98-03, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998. Available at: http://www.dcs.rhbnc.ac.uk/research/compint/areas/comp_learn/sv/pub/report98-03.ps.
[24] G.D. Forney, Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference, IEEE Trans. Inform. Theory 18 (1972) 363–378.
[25] F.R. Magee, J.G. Proakis, Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference, IEEE Trans. Inform. Theory 19 (1973) 120–124.
[26] H. Kubo, K. Murakami, T. Fujino, An adaptive maximum-likelihood sequence estimator for fast time-varying intersymbol interference channels, IEEE Trans. Commun. 42 (1994) 1872–1880.
[27] S. Chen, S. McLaughlin, B. Mulgrew, P.M. Grant, Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels, IEEE Trans. Commun. 43 (1995) 1937–1946.
[28] S. Chen, S. McLaughlin, B. Mulgrew, Complex-valued radial basis function network, Part II. Application to digital communications channel equalization, Signal Process. 36 (1994) 175–188.
[29] I. Cha, S.A. Kassam, Channel equalization using adaptive complex radial basis function networks, IEEE J. Select. Areas Commun. 13 (1995) 122–131.
[30] D. Jianping, N. Sundararajan, P. Saratchandran, Communication channel equalization using complex-valued minimal radial basis function neural networks, IEEE Trans. Neural Networks 13 (2002) 687–696.
[31] C. Burges, Simplified support vector decision rules, in: L. Saitta (Ed.), Proc. ICML'96, Morgan Kaufmann, San Mateo, CA, 1996, pp. 71–77.
[32] E. Bayro-Corrochano, S. Buchholz, Geometric neural networks, in: Int. Workshop Algebraic Frames for the Perception Action Cycle, AFPAC'97, Lecture Notes in Computer Science, vol. 1315, Springer, Kiel, 1997, pp. 379–394.
[33] E. Bayro-Corrochano, Geometric neural computing, IEEE Trans. Neural Networks 12 (2001) 968–986.

Li Zhang received the B.S. degree in 1997 and the Ph.D. degree in 2002 in electronic engineering from Xidian University, Xi'an, China. From 2003 to 2005, she was a postdoctor at the Institute of Automation of Shanghai Jiao Tong University, Shanghai, China. Now, she is an associate professor at Xidian University. Her research interests have been in the areas of machine learning, pattern recognition, neural networks and intelligent information processing.

Weida Zhou received the B.S. in 1996 and the Ph.D. degree in 2003 in electronic engineering from Xidian University, Xi'an, China. He has been an associate professor at the School of Electronic Engineering at Xidian University, Xi'an, China since 2003. His research interests include machine learning, learning theory and intelligent information processing.

Licheng Jiao received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees from Xi'an Jiao Tong University, Xi'an, China, in 1984 and 1990, respectively. He is currently a professor and the dean of the Electronic Engineering School at Xidian University. His research interests include neural networks, data mining, nonlinear intelligent signal processing, and communication.