A recurrent neural network-based method for training probabilistic Support Vector Machine

Hadi Sadoghi Yazdi*

Computer Department,
Ferdowsi University of Mashhad,
Mashhad, Iran
E-mail: h-sadoghi@ferdowsi.um.ac.ir
E-mail: sadoghi@sttu.ac.ir
*Corresponding author

Sohrab Effati

Department of Mathematics,
Ferdowsi University of Mashhad,
Mashhad, Iran
E-mail: effati@sttu.ac.ir

Zahra Saberi

Department of Mathematics,
Sabzevar Tarbiat Moallem,
University of Sabzevar,
Sabzevar, Iran
E-mail: z_sberi2007@yahoo.com

Abstract: In this paper, Support Vector Machine (SVM) is reformulated to a recurrent neural network model which can be described by the nonlinear dynamic system. In the proposed algorithm, an iterative training procedure is proposed independent of initial point. Also probabilistic constraints are recommended for reducing effect of noisy samples in training procedure and appearance of incorrect Support Vectors (SV). Probabilistic constraints admit using knowledge about distribution function of samples. A set of differential equations is used to modelling of the proposed probabilistic SVM. These equations are converged to optimal solution for SVM. The Euler method is used to solve differential equation. The primal and dual problem of SVM is solved by this model. Enough information is given for finding optimal hyper plane. Capability of the proposed method is shown by experimental results in the Optical Character Recognition (OCR) and synthetic data.

Keywords: recurrent neural network model; differential equation; probabilistic constraints; SVC; support vector machine; OCA; optical character recognition.

Reference to this paper should be made as follows: Yazdi, H.S., Effati, S. and Saberi, Z. (2009) ‘A recurrent neural network-based method for training probabilistic Support Vector Machine’, *Int. J. Signal and Imaging Systems Engineering*, Vol. 2, Nos. 1/2, pp.57–65.

Biographical notes: Hadi Sadoghi Yazdi received the BS Degree in Electrical Engineering from Ferdowsi University of Mashad in 1994, and then he received to the MS and PhD Degrees in Electrical Engineering from Tarbiat Modarres University of Iran, Tehran, in 1996 and 2005 respectively. He works in Computer Department as an Assistant Professor at Ferdowsi University of Mashad and Engineering Department of Tarbiat Moallem University of Sabzevar. His research interests include adaptive filtering, image and video processing, and optimisation in signal processing. He has more than 120 journal and conference publications in subject of interesting area.

Sohrab Effati was born in Ghoochan, Iran. He received the PhD Degree in Applied Mathematics. He works in Mathematics Department as an Associated Professor at Ferdowsi University of Mashad and Mathematics Department of Tarbiat Moallem University of Sabzevar. His research interests include optimisation, and operational research.

Zahra Saberi recived the MS Degree from Tarbiat Moallem University of Sabzevar. He works as an teacher in the Esfahan Teaching and Culture organisation.

1 Introduction

Support Vector Machines (SVMs) as originally introduced by Vapnik within the area of statistical learning theory and structural risk minimisation (Vapnik, 1995) and create a classifier with minimised Vapnik-Chernovenkis (VC) dimension. In statistical learning theory, the SVM has been developed for data classification and function estimation and it is used in wide range applications such as Optical Character Recognition (OCR) (LeCun et al., 1995; Liu et al., 2003) text categorisation (Joachims, 1998), face detection in images (Osuna et al., 1997), vehicle tracking in video sequence (Avidan, 2004), nonlinear equalisation in communication systems (Sebald and Bucklew, 2000) and generating of fuzzy rule based system using SVM framework (Chiang and Hao, 2004; Chen and Wang, 2003). This paper emphasis over noisy learning data and learning scheme. Following text proceeds to literature about these problems.

As shown in (Guyon et al., 1996; Zhang, 1999), SVM is very sensitive to outliers or noises since the penalty term of SVM treats every data point equally in the training process. This may result in the occurrence of over fitting problem if one or few data points have relatively very large values of slack variable. The Fuzzy SVM (FSVM) to deal with the over fitting problem. FSVM is an extension of SVM that takes into account the different significance of the training samples. For FSVM, each training sample is associated with a fuzzy membership value. The membership value reflects the fidelity of the data; in other words, how confident we are about the actual class information of the data. The higher its value, the more confident we are about its class label. The optimisation problem of the FSVM is formulated in (Lin and Wang, 2002; Huang and Liu, 2002) and have used in works such as (Liu and Chen, 2007; Wang and Chiang, 2006; Yang et al., 2006). In this method slack variable is scaled by the membership value. The fuzzy membership values are used to weight the soft penalty term in the cost function of SVM. The weighted soft penalty term reflects the relative fidelity of the training samples during training. Important samples with larger membership values will have more impact in the FSVM training than those with smaller values.

In addition SVM is a constraint quadratic programming problem and can be solved using existing methods. However, traditional algorithms for digital computers may not be efficient since the computing time required for a solution is greatly dependent on the dimension and structure of the problems. One possible and very promising approach to real-time optimisation is to apply artificial neural networks. Because of the inherent massive parallelism, the neural network approach can solve optimisation problems in running time at the orders of magnitude much faster than those of the most popular optimisation algorithms executed on general-purpose digital computers.

In Hopfield and Tank (1985) and Tank and Hopfield (1986) proposed a neural network for solving linear programming problems. Their main work has inspired many researchers to investigate alternative neural networks for solving linear and nonlinear programming problems. In 1987, Kennedy and Chua (1988) proposed an improved model that always guaranteed convergence. However, their new model converges to only an approximation of the optimal solution. In 1990, Rodriguez-Vazquez et al. (1990) proposed a class of neural networks for solving optimisation problems. In 1996, Wu et al. (1996) introduced a new model that solves both the primal and dual problems of linear and quadratic programming problems. This new model always globally converge to the solutions of the primal and dual problems. In 2004, Effati and Baymani (2004) presented a new nonlinear neural network that not only retains the advantages of Wu and Xia's model but also has a much faster converges. Unlike Xia's model which is piecewise linear, this model is based on a nonlinear dynamical system.

Contribution

Main problems in the SVM are how defining cost function, constraints and how finding variables. New notes in this paper are,

- SVM is converted to form of nonlinear neural network model which can be described by the nonlinear dynamical system.
- A set of differential equations is used to modelling of dynamics and is converged to stable state which is optimal solution for the SVM.
- Probabilistic constraints were used in SVM (PC-SVM) for superiority at noise.

The rest of the paper is organised as follows: Section 2 introduces the SVM with probabilistic constraints and learning scheme. Experimental discussion is mentioned in section 3. Conclusions are given in Section 4.

2 The recurrent neural network-based method for solving the probabilistic constraints SVM

At first, this section appropriates to a brief introduction on SVM, then the proposed method is studied.

2.1 Support Vector Machine formulation

Let $S = \{(x_i, d_i)\}_{i=1}^{n}$ be a set of n training samples, where $x_i \in R^m$ is an m-dimensional sample in the input space, and $d_i \in \{-1, 1\}$ is the class label of x_i. SVM finds the optimal separating hyper plane with the minimal classification errors. Let w_0 and b_0 denote the optimum values of the weight vector and bias respectively. The hyper plane can be represented as:

$$w_0^T x + b_0 = 0 \tag{1}$$

where $w = [w_1, w_2, \ldots, w_m]^T$ and $x = [x_1, x_2, \ldots, x_m]^T$. w is the normal vector of the hyper plane, and b is the bias that is a scalar. The optimal hyper plane can be obtained by solving the following optimisation problem (Vapnik, 1995):

$$\text{Minimise} \quad \frac{1}{2}\|w\|^2$$
$$\text{s.t.} \quad d_i(w^T x_i + b) \geq 1 - \xi_i, \tag{2}$$
$$\xi_i \geq 0, \quad i = 1, ..., n.$$

2.2 The proposed algorithm

Optimal hyper plane can be obtained by substitution constraints in equation (2) by following constraints:

$$\Pr(d_i(w^T x_i + b) \geq u_i) \geq \delta \tag{3}$$

where u_i are independent random variables with known distribution functions and $0 \leq \delta_i \leq 1$ is value of effect of ith samples in the position of optimal hyper plane or selection of SV. Then equation (3) can be written as,

$$d_i(w^T x_i + b) \geq F_i^{-1}(\beta_i) \tag{4}$$

where $\beta_i = 1 - \delta_i$, and $F_i^{-1}(.)$ is the inverse normal distribution function of the variable u_i, with $i = 1, \ldots, n$; which has to be continuous. In there if distance of samples and SV is more than between mean of samples and SVs; we suppose maximum probability for δ_i is 0.95, else calculate it by normal distribution function. Similar to the conventional SVM, the optimisation problem of PC-SVM can be transformed into its dual problem. There are two reasons for moving to the dual form of the problem; one is constraints are significantly simpler than primal form and the other is the training data will appear to dot products form.

$$F_i^{-1}(\beta_i) - d_i(w^T x_i + b) \leq 0 \tag{5}$$

Optimisation procedure continues as follows,

$$J(w, b, \alpha) = \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i [F^{-1}(\beta_i) - d_i(wx_i + b)]$$
$$\frac{\partial J}{\partial w} = w - \sum_{i=1}^{n} \alpha_i d_i x_i = 0 \tag{6}$$
$$\frac{\partial J}{\partial b} = \sum_{i=1}^{n} \alpha_i d_i = 0.$$

For solving this problem, it is converted to dual form. With given the training sample $\{(x_i, d_i)\}_{i=1}^n$ find the Lagrange multipliers $\{\alpha_i\}_{i=1}^n$ that maximise the objective function.

$$\text{Maximise} \quad \sum_{i=1}^{n} \alpha_i F^{-1}(\beta_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j d_i d_j x_i x_j \tag{7}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} d_i \alpha_i = 0, \tag{8}$$
$$0 \leq \alpha_i, \quad i = 1, ..., n.$$

2.2.1 Mathematical background

Consider QP problem be of the form:

$$\text{Minimise} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j d_i d_j x_i x_j$$
$$\text{s.t.} \quad F_i^{-1}(\beta_i) - d_i b - d_i\left(\sum_{j=1}^{n} \alpha_j d_j x_j\right) x_i \leq 0 \tag{9}$$
$$\alpha_i \geq 0.$$

Rewrite the primal problem to the following form:

$$\text{Minimise} \quad \frac{1}{2}\alpha^T A \alpha$$
$$\text{s.t.} \quad a - Db - A\alpha \leq 0 \tag{10}$$
$$\alpha \geq 0$$

where A is an $n \times n$ symmetric positive semi definite matrix, D, a, $\alpha \in R^n$. We define the dual problem (DQP) as follows**:**

$$\text{Maximise} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j d_i d_j x_i x_j + \sum_{i=1}^{n} \alpha_i F^{-1}(\beta_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{n} d_i \alpha_i \leq 0 \tag{11}$$
$$\alpha_i \geq 0.$$

Rewrite the dual problem to the following problem:

$$\text{Maximise} \quad -\frac{1}{2}\alpha^T A \alpha + a^T \alpha \tag{12}$$
$$\text{s.t.} \quad D^T \alpha = 0, \quad \alpha \geq 0.$$

Theorem 1: *For* $G = \{(\alpha, b)/\alpha, \geq 0\}\,\alpha$, b *are optimal solutions to QP, DQP , respectively, if and only if* α *and* b *satisfy the following constraints:*

$$D^T \alpha = 0$$
$$a - Db - A\alpha \leq 0 \tag{13}$$
$$\alpha^T(a - Db - A\alpha) = 0, \quad \alpha \geq 0$$

Proof: See Bazaraa et al. (1992).

2.2.2 The recurrent neural network-based method for solving QP problems

The neural network model can be described by the following nonlinear dynamical system:

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = a - A\left(\alpha + k\frac{\mathrm{d}\alpha}{\mathrm{d}t}\right) - D\left(b + k\frac{\mathrm{d}b}{\mathrm{d}t}\right), \quad \alpha \geq 0 \tag{14}$$

$$\frac{\mathrm{d}b}{\mathrm{d}t} = D^T\left(\alpha + k\frac{\mathrm{d}\alpha}{\mathrm{d}t}\right) \tag{15}$$

where, coefficient k is some positive constant. The main property of the above system is mentioned in the following theorem.

Theorem 2: *Recurrent neural network with dynamic described using differential equations* (14) *and* (15) *is converged to optimal solution for the QP problem* ((9) *or* (10)) *and its dual problem DQP* ((11) *or* (12)).

Proof: Let α_I be the ith component of α in equation (14) can be rewritten as:

$$\frac{d\alpha_i}{dt} = \left\{ a - A\left(\alpha + k\frac{d\alpha}{dt} \right) - D\left(b + k\frac{db}{dt} \right) \right\} \quad \text{if } \alpha_i \geq 0 \tag{16}$$

$$\frac{d\alpha_i}{dt} = \max\left\{ \left[a - A\left(\alpha + k\frac{d\alpha}{dt} \right) - D\left(b + k\frac{db}{dt} \right) \right]_i, 0 \right\} \quad \text{if } \alpha_i = 0 \tag{17}$$

For ensuring that lower bound of α is zero, condition (17) has been used. Let α^* and b^* be the limit of α and b respectively, that is

$$\lim_{t\to\infty} \alpha(t) = \alpha^*, \quad \lim_{t\to\infty} b(t) = b^*.$$

For stability of convergence, we have $d\alpha^*/dt = 0$ and $db^*/dt = 0$. So equations (16) and (17) becomes:

$$[a - A\alpha^* - Db^*]_i = 0 \quad \text{if } \alpha_i^* > 0 \tag{18}$$

$$\max\{[a - A\alpha^* - Db^*]_i, 0\} = 0, \quad \alpha_i^* = 0. \tag{19}$$

In other words:

$$[a - A\alpha^* - Db^*]_i \leq 0 \tag{20}$$

$$\alpha_i^*[a - A\alpha^* - Db^*]_i = 0 \tag{21}$$

Or

$$[a - A\alpha^* - Db^*] \leq 0 \tag{22}$$

$$\alpha_i^*[a - A\alpha^* - Db^*] = 0. \tag{23}$$

Similarly, taking the limit of equation (15) gives:

$$D^T\alpha^* = 0. \tag{24}$$

The constraints (22) and (23) show that α^* and b^* are the feasible solutions for QP and DQP problems. After simplification of equation (23) and using equation (24) and according (Bazaraa et al., 1992), we have:

$$\begin{aligned} &\alpha^T\alpha^* - \alpha^{*T}A\alpha^* - b^*D^T\alpha^* = 0 \\ &\frac{1}{2}\alpha^{*T}A\alpha^* = -\frac{1}{2}\alpha^{*T}A\alpha^* + a^T\alpha^*. \end{aligned} \tag{25}$$

By QP Duality theorem (Theorem 1) from equations (22)–(25) shows that α^* and b^* are the optimal solutions for the QP and DQP problems (10), (12).

Note: The Euler method can be used to solve differential equations (14) and (15). The following Matlab pseudo-code describes the discrete implementation of our neural network. For simplification, we select coefficient k equal to dt simplify the calculations.

3 Experimental results

The proposed method is studied in this section include checking effect of probabilistic constraints, convergence behaviour of neural network model for SVM, operation of the proposed method over OCR application.

3.1 Study of probabilistic constraints

In most real life problems the data are not linearly separable. One method is to apply nonlinear transforms to the original data, for example, creating a higher dimensional vector by multiplying all the terms in the feature vector with each other. This leads to the kernel trick that involves implicitly mapping our data from input space into a (usually much higher dimensional) feature space via a nonlinear kernel function, hiding the potentially high dimensionality of that feature space and, thus avoiding the curse of dimensionality (Haykin, 1999). The kernel function $K(x_i, x_j)$ is related to the nonlinear feature mapping function $\varphi(x)$ by $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$.

In the proposed method, constraints in equation (3) is converted to following form,

$$\Pr(d_i(w^T\varphi^T(x_i) + b) \geq u_i) \geq \delta_i. \tag{26}$$

The non-linear mapping function $\varphi(.)$ is performed in accordance with Cover's theorem, which guarantees that the transformed samples are more likely to be linearly separable in the resulting feature space. For example, a nonlinear separable of samples in feature space of (x_2, x_1) as shown in Figure 1 are converted using polynomial kernel $\left(\varphi(x) = \left[1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2\right]\right)$ to six dimension space with optimum weights of [–38.3404, 37.6050, 0.7653, 40.8324, 0.9881, 1.2793] which can recognise samples more than 98% on average.

In high dimensional space, a linear hyper plane can separate with high recognition rate. If noisy data are applied to training or finding optimum hyper plane, low performance rate is achieved in testing phase. So, as shown in equation (26) probabilistic constraints are used for reducing effect of noisy samples in determining SV. Start with one example for explaining of cogency of probabilistic constrains. Figure 2 shows two classes after transfer to high dimensional space (new space is two dimensional) but with these following notes,

- class 1 is condense and class 2 is disperse
- sample of class 1 have high confidence in collecting and measurement but class 2 have not these properties
- training data in class 2 polluted with noise whilst data capturing.

Figure 1 Nonlinear separable of samples in feature space of (x_2, x_1) and Support Vectors in input space (see online version for colours)

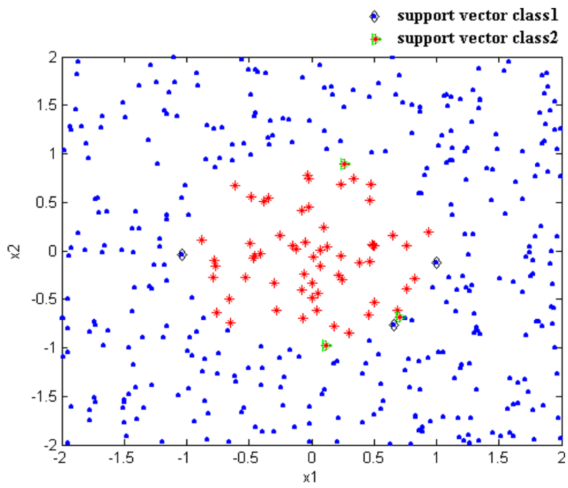
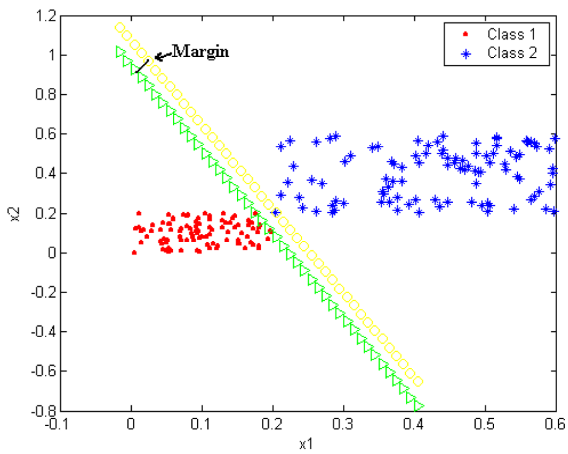


Figure 2 Captured margin using standard SVM (see online version for colours)



Margin have been found in the standard SVM without a priori knowledge about probability density function over confidence in collected data's and level of added noise. If we know about above mentioned notes, we can create one soft margin based of reliability of data's. We assume that, this reliability has semi normal PDF as follows,

Class 1

$$N_s(x,\mu_1,\sigma_1): f_u(x) = \begin{cases} \frac{1}{\sigma_1\sqrt{2\pi}} e^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}} & \forall x^- \\ 1 & \text{otherwise} \end{cases} \quad (27)$$

Class 2

$$N_s(x,\mu_2,\sigma_2): f_u(x) = \begin{cases} \frac{1}{\sigma_2\sqrt{2\pi}} e^{\frac{-(x-\mu_2)^2}{2\sigma_2 2}} & \forall x^- \\ 1 & \text{otherwise} \end{cases} \quad (28)$$

where $N_s(x, \mu_i, \sigma_i)$ is probability density function which shows that reliability of data's. $f_u(x)$ in equations (27) and (28) is PDF of u in equation (26). For x near the support vector (x^- which are in the left of mean class for class 2 and right of mean class for class 1), this PDF is normal and probability for samples far from mean class is one. This work helps for high effect of samples far from support vector and lower effect of samples that is near to SV in calculation parameters of optimal hyper plane. These probabilities show reliability value of class data's. For presentation in one figure, probability of class 1 are shown to negative form and class 2 to positive form in Figure 3. In this figure, old SV in conventional SVM have low reliability (probability near to zero) and centre of class have maximum reliability (probability near to 1 for class 2 and near to –1 for class 1). So it is expected samples far from conventional SV attract hyper planes toward their selves. If reliability for samples of class 1 is bigger than class 2, it is expected hyper plane related to class 2 moves further but hyper plane class 1 have been moved slightly which is illustrated in Figure 4. Margin is incremented asymmetric.

Figure 3 Confidence of data's for two class. For better demonstration, probability of class 1 has been multiplied by minus and class 2 with positive probability (see online version for colours)

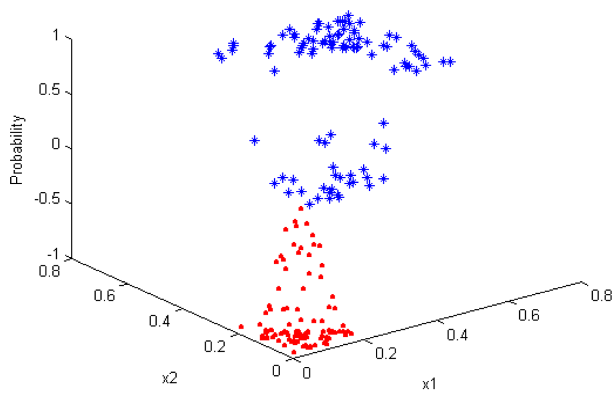
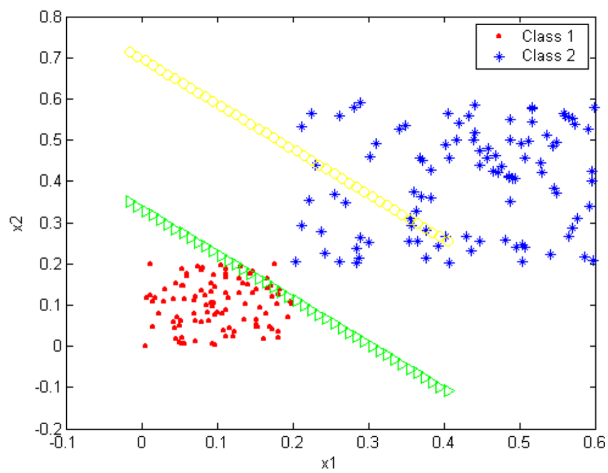


Figure 4 Effect of reliability over hyper planes and moving of hyper plane as asymmetric. Hyper plane related to class 2 moves further but hyper plane class 1 have been moved slightly (see online version for colours)

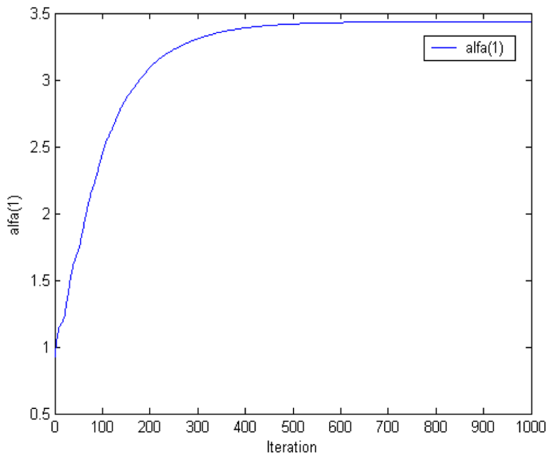


One note must be interested about data points falls on wrong side of the optimal decision surface in the standard SVM. In the proposed SVM, a priori knowledge can emphasis over these data points and use it in obtaining optimal hyper plane but in conventional SVM do not exist this capability.

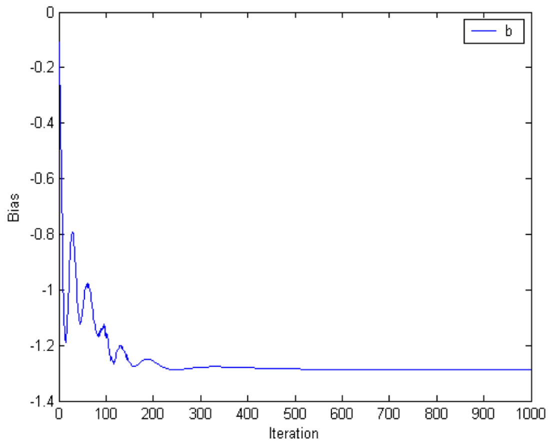
3.2 Checking convergence behaviour of recurrent neural network model for SVM

For convergence stability of the proposed neural network model (14), (15) $d\alpha/dt = 0$ and $db^*/dt = 0$ in limit state are achieved. In above example first Lagrange multiplier (α_1) and bias (b) are obtained. As shown in Figure 5(a) and (b), these coefficients are converged to optimal value and have been stabilised using Euler method. A step size (μ) is used to $b = b + \mu db$, $\alpha = \alpha + \mu d\alpha$ in Euler method for increasing convergence speed, which decrease per iteration of course increasing dt give similar result like μ.

Figure 5 Convergence behaviour in neural network model: (a) first Lagrange multiplier (α_1) and (b) bias (b) (see online version for colours)



(a)

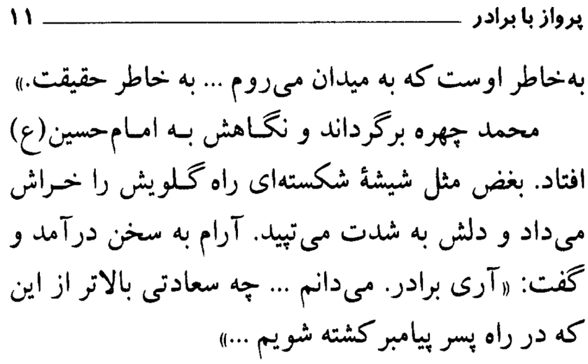


(b)

3.3 Operation of the proposed method in the Optical Character Recognition

In this sub-section, OCR is discussed for comparing the proposed method and standard SVM. Character recognition is commonly known as OCR which deals with the recognition of optical characters. OCR has wide applications in modern society: Document reading and sorting, postal address reading, bank check recognition, form recognition, signature verification, digital bar code reading, map interpretation, engineering drawing recognition, and various other industrial and commercial applications. The difficulty of the text recognition greatly depends on the type of characters to be recognised. The difficulty varies from that needed to process relatively easy mono fonts to that of extremely difficult cursive text. We focus over Farsi OCR (FOCR) for converting Farsi document image as Figure 6 to text. Main target of this paper is test of the proposed method but we point abstractly to all parts of FOCR.

Figure 6 A sample of Farsi document image


پرواز با برادر ۱۱

به‌خاطر اوست که به میدان می‌روم ... به خاطر حقیقت.»
محمد چهره برگرداند و نگاهش به امام‌حسین(ع)
افتاد. بغض مثل شیشهٔ شکسته‌ای راه گلویش را خراش
می‌داد و دلش به شدت می‌تپید. آرام به سخن درآمد و
گفت: «آری برادر. می‌دانم ... چه سعادتی بالاتر از این
که در راه پسر پیامبر کشته شویم ...»

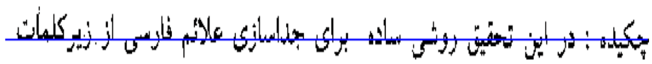
Main parts of the FOCR are as follows:

- image capturing as Figure 6
- line separation and global baseline detection (Figure 7)
- word and sub-word segmentation (Figure 8) and global baseline correction (Figure 9)
- character extraction from word (Figure 11)
- point recognition
- character recognition using the proposed method
- finding final result using string checking.

Line separation and global base line detection

Using horizontal profile of capture image separates lines, then maximum peak of horizontal profile of each line is determined as global base line in Figure 7.

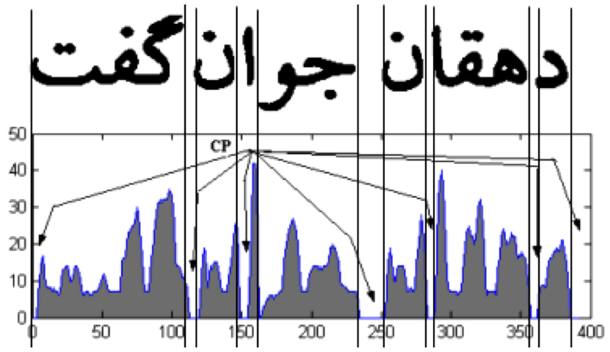
Figure 7 Global base line detection procedure (global base line is shown with blue colour) (see online version for colours)



Word and sub-word segmentation and local base line correction

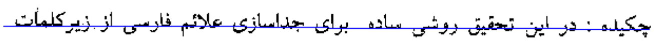
For word and sub-word segmentation, vertical profile is obtained and valleys are specified as Cutting Position (CP) of image to word or sub-word as shown in Figure 8.

Figure 8 Word and sub-word segmentation (see online version for colours)



Tilt may occur during scanning of document, so baseline must be corrected. This work can be performed using local baseline or contour in each word. Result of baseline correction for Figure 7 is shown in Figure 9.

Figure 9 Global baseline correction using word contour (see online version for colours)



Character extraction from word

In next stage, word is divided to characters and each point is allocated to word body with distance criteria as shown in Figure 10. The word body is segmented to characters using vertical profile or over image contour (Figure 11).

Figure 10 Point allocation to word body (see online version for colours)

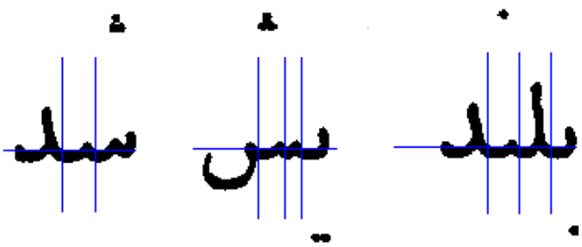
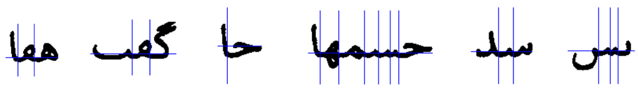


Figure 11 Character extraction from words (see online version for colours)



Point recognition

In this part, points categorised to three types (Figure 12): one point or two or three points. This work is performed using features as volume of pixel in point per pen width, symmetry or asymmetry in point image, height per width.

Figure 12 Three types of points

Character recognition using the proposed method

Points in Farsi OCR have an important role. Cutting word to characters follows point assignment to characters or sub-characters and with regards to points recognition which is explained in the above sub-section; we can categorise characters to six groups.

a without points (Figure 13(a))

b with one point in top (Figure 13(b))

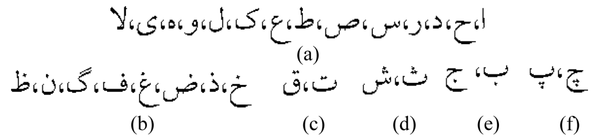
c with two points in top (Figure 13(c))

d with three points in top (Figure 13(d))

e with one point in bottom (Figure 13(e))

f with three points in bottom (Figure 13(f))

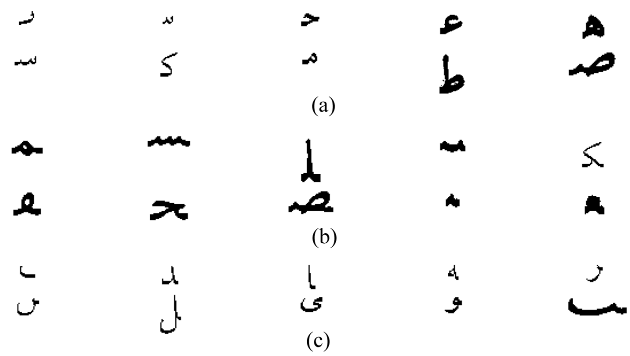
Figure 13 six groups of characters per points location



Next parts of character recognition procedure are body of characters recognition. Body of characters is categorised to four types:

- isolated body include 18 categories (some of those are shown in Figure 13(a))
- first blob include 11 categories (blob is black part or part of body) (some of those are shown in Figure 14(a))
- middle blob include 18 categories (some of those are shown in Figure 14(b))
- end blob include 11 categories (some of those are shown in Figure 14(c)).

Figure14 Characters body types: (a) first blob; (b) middle blob and (c) end blob



For each type of character body, we perform a multiclass classifiers using SVM over 1856 training samples in four types of character body in four fonts. Noise in training samples forces us for searching of robust SVM against noise which this motivation obtains the proposed method. Sample of document image is shown in following figure.

Parameters of the proposed method are determined manually. In equation (26) for each class u_i has distribution function with variance 0.4 and in equation (26) δ_i is selected 90%. In neural network-based training procedure (16, 17) dt is selected 0.02 and in Figure 15 for Euler method iteration number n is selected 150 also for increasing convergence speed a step size ($\mu = 2.1$) is selected and Euler method has been changed to form of $b = b + \mu db$, $\alpha = \alpha + \mu db$. 3016 samples are used in testing phase. Results show 3.7% increasing in recognition rate for the proposed method relative standard SVM. Of course with increasing of noise to signal, superiority of the proposed method has been shown.

Figure 15 Matlab pseudo-code describes the discrete implementation of our neural network

```
For i = 1 : n
    db = dt * (D^T * (α + dα));
    b = b + db;
    dα = dt * (a − A * (α + dα) − D * (b + db));
    dα = max(α + dα, 0) − α;
    α = α + dα;
end
```

Figure 16 Noise in character body: (a) extra adhesive pixels; (b) low resolution scanning; (c) fragmental body and (d) noise in contour

(a)

(b)

(c)

(d)

A note is an important in tuning procedure in the proposed method. If you use training data without noise or low noise, we proposed you use the proposed method with δ_i bigger than 95%. If you use it to form of standard SVM or δ_i equal 100% then results is not as well as δ_i smaller than 100%. We think around this problem and we find that some types of noise disturb character body. Applied feature for recognition of character body is Loci which is explained as follows.

Loci feature is defined for each white pixel in image and usually is obtained in horizontal and vertical direction. As shown in Figure 17, loci vector in horizontal and vertical direction is acquired from intersection number of horizontal and vertical lines with character body. If these intersection is truncated to 2, a 4-digit number is resulted in radix three for number of intersection in directions of top, down, left and right of following form in radix 10.

$$\begin{aligned}\text{Cutting Number} = {} & \text{Right Num Of Cut} \times 3^0 \\ & + \text{Top Num Of Cut} \times 3^1 \\ & + \text{Left Num Of Cut} \times 3^2 \\ & + \text{Down Num Of Cut} \times 3^3. \end{aligned} \tag{29}$$

Figure 17 Loci feature extraction for Farsi character س (Sin)

For example for Farsi character س (Sin) and in point of P, loci feature is,

$$\begin{aligned}\text{Cutting Number} &= 0\times 3^0 + 1\times 3^1 + 1\times 3^2 + 0\times 3^3 \\ &= (0110)_3 = 12.\end{aligned} \tag{30}$$

So, for each white pixel a number in radix 10 is obtained which is between zeros to 80. Finally, frequency of each number of zero to 80 is obtained one vector of 81 elements. Loci vector is obtained with normalisation of obtained vector to number of white pixels.

Point allocation and checking string

After recognition of character body, points are allocated to body and sit to one of groups in Figure 13. Jointing characters to form of word is performed based on checking string. Some rules for string checking are,

- if ♠ is appeared before س is deleted
- if two bodies of ♠ are appeared then it is converted to س
- if ژ is occurred before one or two ♠ then it is converted to ش

More than 30 rules correct final string and results include Farsi text. Of course main target of this paper is presentation of new SVM but we try point some notes in FOCR.

4 Conclusion

This paper presented a neural network-based method for solving SVM with probabilistic constraints. We used the proposed method in Farsi OCR. Important note in using the proposed method is parameters tuning. Experimental showed better results, it is better we assume training data are noisy. In the future work, we hope find a way for finding parameters in the proposed method.

References

Avidan, S. (2004) 'Support Vector Tracking', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, August, pp.1064–1072.

Bazaraa, M.S., John, J. and Jarvis, H.D. (1992) *Sherali, Linear Programming and Network Flows*, John Wiley and Sons, New York.

Chen, Y. and Wang, J.Z. (2003) 'Support vector learning for fuzzy rule-based classification systems', *IEEE Trans. on Fuzzy Systems*, Vol. 11, No. 6, December, pp.716–728.

Chiang, J-H. and Hao, P-Y. (2004) 'Support vector learning mechanism for fuzzy rule-based modeling: a new approach', *IEEE Trans. on Fuzzy Systems*, Vol. 12, No. 1, February, pp.1–12.

Effati, S. and Baymani, M. (2004) 'A new nonlinear neural network for solving convex nonlinear programming problems', *Appl. Math. Comput.*, pp.1–3.

Guyon, I., Matic, N. and Vapnik, V. (1996) 'Discovering informative patterns and data cleaning', in Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (Eds.): *Advances in Knowledge Discovery and Data Mining, Menlo Park*, AAAI Press, CA, pp.181–203.

Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, NJ.

Hopfield, J.J. and Tank, D.W. (1985) 'Neural computation of decisions in optimization problems', *Biol. Cybern.*, Vol. 52, pp.141–152.

Huang, H-P. and Liu, Y-H. (2002) 'Fuzzy support vector machine for pattern recognition and data mining', *Int. J. Fuzzy Syst.*, Vol. 4, No. 3, pp.826–835.

Joachims, T. (1998) 'Text categorization with support vector machines: learning with many relevant features', in Nedellec, C. and Rouveirol, C. (Eds.): *Proc. Europ. Conf. Mach. Learn.*, Berlin, Germany, pp.137–142.

Kennedy, M.P. and Chua, L.O. (1988) 'Neural networks for nonlinear programming', *IEEE Trans. Circ. Syst.*, Vol. 35, pp.554–562.

LeCun, Y., Botou, L., Jackel, L., Drucker, H., Cortes, C., Denker, J., Guyon, I., Muller, U., Sackinger, E., Simard, P. and Vapnik, V. (1995) 'Learning algorithms for classification: a comparison on handwritten digit recognition', *Neural Netw.*, pp.261–276.

Lin, C.F. and Wang, S.D. (2002) 'Fuzzy support vector machines', *IEEE Trans. Neural Networks*, Vol. 13, No. 2, March, pp.464–471.

Liu, C., Nakashima, K., Sako, H. and Fujisawa, H. (2003) 'Handwritten digit recognition: Bench-marking of state-of-the-art techniques', *Pattern Recognit.*, Vol. 36, pp.2271–2285.

Liu, Y-H. and Chen, Y-T. (2007) 'Face recognition using total margin-based adaptive fuzzy support vector machines', *IEEE Trans. on Neural Networks*, Vol. 18, No. 1, January, pp.178–192.

Osuna, E., Freund, R. and Girosi, F. (1997) 'Training support vector machines: An application to face detection', *IEEE Conf. Comput. Vis. Pattern Recognit.*, January, pp.130–136.

Rodriguez-Vazquez, A., Dominguez-Castro, R., Rueda, A., Huertas, J.L. and Sanchez-Sinencio, E. (1990) 'Nonlinear switched-capacitor neural networks for optimization problems', *IEEE Trans. Circ. Syst.*, Vol. 37, pp.384–397.

Sebald, D.J. and Bucklew, J.A. (2000) 'Support Vector Machine techniques for nonlinear equalization', *IEEE Trans. on Signal Processing*, Vol. 48, No. 11, November, pp.3217–3226.

Tank, D.W. and Hopfield, J.J. (1986) 'Simple neural optimization networks: an A/D converter, signal decision circuit and a linear programming circuit', *IEEE Trans. Circ. Syst.*, Vol. 33, p.533.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Wang, T-Y. and Chiang, H-M. (2006) 'Fuzzy support vector machine for multi-class text categorization', *Information Processing and Management*, doi:10.1016/j.ipm.2006.09.011

Wu, Y., Xia, Y., Li, J. and Chen, W. (1996) 'A high-performance neural network for solving linear quadratic programming problems', *IEEE Trans. on Neural Networks*, Vol. 7, pp.643–651.

Yang, C-H., Jin, L-C. and Chuang, L-Y. (2006) 'Fuzzy support vector machines for adaptive Morse code recognition', *Medical Engineering and Physics*, Vol. 28, pp.925–931.

Zhang, X. (1999) 'Using class-center vectors to build support vector machines', *Proc. IEEE Workshop Neural Networks Signal Process. (NNSP'99)*, Madison, WI, pp.3–11.