ADAPTIVE SVC FOR NONLINEAR CHANNEL EQUALIZATION

Fernando Pérez-Cruz*and Antonio Artés-Rodríguez
DTSC Universidad Carlos III de Madrid.
e-mail: fernandop@ieee.org

ABSTRACT

We propose a new scheme for adapting support vector classifiers (SVC) in non-stationary environments. This adaptive SVC (ASVC) relies in interpreting the margin and penalty factor of the SVC as a relevance measure over the samples and on an iterative re-weighted least square (IRWLS) approach for optimizing it, which resembles the RLS filtering for adaptive equalization. The ASVC capabilities are shown by means of computer experiments.

1 Introduction

Channel equalization is a major issue in digital communications, because the channel effects the transmitted sequence with both linear (inter-symbol interference) and nonlinear (amplifiers and converters) distortions. Also, the communication channel can not be considered stationary, because in some cases, such as mobile communications or voice-band modems, the channel is significantly modified with time. Support Vector Classifiers (SVCs) are **block** state-of-the-art tools for knowledge discovery [7], that have been successfully applied to solve the stationary channel equalization problem [1, 5, 8]. But the non-stationary channel equalization problem has not been tackled with it yet.

We will address this problem first by noticing that the SVC margin and penalty factor can be regarded as the relevance of each input pattern and, consequently, modifying (reducing) it for former samples as the communication channel varies. We will then show that the IRWLS procedure [5] can be used to solve an SVC with a different and time varying margin and penalty factor for each sample and that can be naturally modified to include an extra sample each time a new one needs to be equalized. There has been several attempts to formulate an adaptive SVM (none for channel equalization), but they either re-train a block SVM each time a threshold is surpassed [4] or work with every training sample [3], and none of then can operate in a decision directed manner.

*This work has been partially supported by CAM grant 07T/0046/2000.

In Section 2 the margin and penalty factor modification are discussed. The ASVC is optimized in Section 3. Section 4 is devoted to show by means of computer experiments the ASVC capabilities. The paper ends with some concluding remarks in Section 5.

2 Margin and Penalty Factor Modifications

Channel equalizers have to produce the best prediction given the previous samples generated by the channel. In stationary environments every sample will be equally relevant for this task, because they have been independently drawn from the same probability density function, but in non-stationary environments recently known samples are more likely to resemble future samples than formerly known ones, and we should rely most on them. In order to deal with non-stationary environments, we will define a confidence figure for each sample that will be decreased as new samples are known, former samples will lose relevance to solve the actual problem.

The SVC margin and penalty factor can account for the relevance of each input pattern, i.e. a sample with a large margin needs to be further apart from the classification boundary to escape being penalized, and i.e. a sample with high penalty factor means that, if incorrectly classified, the learning machine will incur in a high penalty in its objective function. These two parameters can be regarded as confidence over each sample (the larger they are, the most relevant the sample becomes). The SVC with a different margin and penalty factor for each sample (Modified-SVC) has to solve[1]

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} C_i\xi_i \tag{1}$$

subject to

$$y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) \geq M_i - \xi_i \qquad \forall i = 1,\dots,n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1,\dots,n \tag{3}$$

[1]Given a labeled training set $((\mathbf{x}_i, y_i),\ i = 1,\dots,n,\ \mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$, where i indicates the order in which the samples are received and n is an increasing number).

where $\boldsymbol{\phi}(\cdot)$ defines a nonlinear transformation to a higher dimensional space ($\mathbf{x} \in \mathbb{R}^d \longrightarrow \boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^H$ and $d \leq H$), and $\mathbf{w}$ and b define the linear equalizer in the feature space (nonlinear in the input space, unless $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$). M_i and C_i are, respectively, the margin and penalty factor applied over each sample.

The confidence figure $c(n-i) = c_i^n$ is a decreasing function of $n - i$ ($M_i = 1c_i^n$ and $C_i = Cc_i^n$). This formulation can be either solved using Quadratic Programming schemes [7] or can be solved using the IRWLS procedure [5]. The advantage of the second alternative is that the solution for the $n+1$-sample problem can be readily derived from the n-sample problem solution as we will show in the incoming section.

Both modifications (margin and penalty factor) are simultaneously needed because each one of them accounts for the different kind of support vectors in the SVC solution. The margin modification varies the weight on the solution given by the margin samples ($y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) = M_i$) and the penalty factor modifies the weight on the solution given by the non-margin samples ($y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) < M_i$). So, in order to effectively modify the relevance of every support vector, we need to simultaneously reduce the margin and penalty factor of every given sample.

3 Adaptive SVC

In the previous section we have shown that the SVC margin and penalty factor can be interpreted as a confidence figure over the samples. In this section we will use the M-SVC to construct an Adaptive SVC (ASVC), which needs to solve a new M-SVC each time a new sample becomes known. The M-SVC can be solved using the 3-step IRWLS procedure [5]:

1. Solve:
$$\begin{bmatrix} \mathbf{H} + \mathbf{D_a}^{-1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{D_y}\mathbf{con} \\ 0 \end{bmatrix}$$
where $\mathbf{w} = \sum_{i=1}^n \beta_i \boldsymbol{\phi}(\mathbf{x}_i)$.

2. Recompute:
$$a_i = \begin{cases} 0, & e_i y_i < 0 \\ \dfrac{Cc_i^n}{e_i y_i}, & e_i y_i \geq 0 \end{cases}$$
where $e_i = c_i^n y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b$.

3. Repeat until convergence.

being

$$\begin{aligned} (\mathbf{H})_{ij} &= \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) \\ (\mathbf{D_a}^{-1})_{ij} &= \frac{\delta(i-j)}{a_i} \\ (\mathbf{D_y})_{ij} &= y_i \delta(i-j) \\ \mathbf{con} &= [c_1^n, \ldots, c_n^n]^T \end{aligned}$$

where the $\mathbf{H}$ is known as the kernel matrix, because it is only formed by inner products of the training samples in the feature space. Consequently, the M-SVC does not need to know explicitly the nonlinear mapping, $\boldsymbol{\phi}(\cdot)$, but only its kernel $\kappa(\cdot,\cdot)$, because neither the minimizing procedure nor the use of the equalizer needs to work with $\boldsymbol{\phi}(\cdot)$. The column vectors $\mathbf{y}$, $\mathbf{a}$ and $\boldsymbol{\beta}$ present the obvious expressions. The needed transformations to obtain the IRWLS procedure from (1)-(3) are detailed in [5] for the standard SVC, the modifications for the M-SVC readily follows from it.

3.1 Recursive resolution

The ASVC needs to solve an M-SVC for each new sample. We will now show the n-sample problem can be solved using the solution to the $n-1$-sample problem. In the first step of the IRWLS algorithm the solution to the $n-1$-sample problem provides the inverse of

$$\mathbf{M}_{n-1} = \begin{bmatrix} \mathbf{H}_{n-1} + \mathbf{D}^{-1}_{\mathbf{a}_{n-1}} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}$$

where the subscript denotes the last sample considered in the matrix[2]. The solution to the n^{th} sample problem must initially solve

$$\left[\begin{array}{c|cc} \kappa(\mathbf{x}_n, \mathbf{x}_n) + \dfrac{1}{a_n} & \mathbf{K}_n^T & 1 \\ \hline \mathbf{K}_n & \mathbf{H}_{n-1} + \mathbf{D}^{-1}_{\mathbf{a}_{n-1}} & \mathbf{1} \\ 1 & \mathbf{1}^T & 0 \end{array}\right] \begin{bmatrix} \beta_n \\ \boldsymbol{\beta}_{n-1} \\ b \end{bmatrix} = \begin{bmatrix} y_n c_n^n \\ \mathbf{D}_{\mathbf{y}_{n-1}} \mathbf{con}_{n-1} \\ 0 \end{bmatrix} \tag{5}$$

in which we can distinguish $\mathbf{M}_{n-1}$ and where

$$\mathbf{K}_n = [\kappa(\mathbf{x}_n, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}_{n-1})]^T \tag{6}$$

The inverse of the matrix in (5) can be computed using the inversion of a partitioned matrix lemma [6]:

$$\begin{bmatrix} \beta_n \\ \boldsymbol{\beta}_{n-1} \\ b \end{bmatrix} = \begin{bmatrix} \dfrac{1}{f} & -\dfrac{\mathbf{r}^T}{f} \\ -\dfrac{\mathbf{r}}{f} & \mathbf{T} \end{bmatrix} \begin{bmatrix} y_n c_n^n \\ \mathbf{D}_{\mathbf{y}_{n-1}} \mathbf{con}_{n-1} \\ 0 \end{bmatrix}$$

where

$$f = \kappa(\mathbf{x}_n, \mathbf{x}_n) + \frac{1}{a_n} - \widehat{\mathbf{K}}_n^T \mathbf{M}_{n-1}^{-1} \widehat{\mathbf{K}}_n$$

---

[2] In order to ease the development of the ASVC, we have chosen the following notation for matrices and vectors:

- A matrix (vector) with a numeric subscript denotes that its elements covers the range from the first element to the one point it out by the subscript. For example, the matrix $\mathbf{H}_{n-1}$ is formed by: $(\mathbf{H}_{n-1})_{ij}$ $\forall i, j = 1, \ldots, n-1$.
- A matrix (vector) with a subset as a subscript denotes the matrix (vector) is formed by the elements in the subset. For example, $\mathbf{H}_{s_1,s_3}$ is formed by: $(\mathbf{H}_{s_1,s_3})_{ij}$ $\forall i \in s_1, \forall j \in s_3$.

The vectors and matrices that do not follow the above notation will be clearly defined in the text.

$$\mathbf{r} = \mathbf{M}_{n-1}^{-1}\widehat{\mathbf{K}}_n$$
$$\mathbf{T} = \mathbf{M}_{n-1}^{-1} + \frac{\mathbf{r}\mathbf{r}^T}{f}$$

and $\widehat{\mathbf{K}}_n = [\mathbf{K}_n^T \quad 1]^T$. Now we can compute the second step of the IRWLS procedure and continue until there is no further changes in either $\boldsymbol{\beta}_n$ or b.

We have written down an algorithmic implementation of the ASVC procedure in Table 1, which is based in solving an M-SVC with the IRWLS procedure and including an extra sample using the inverse of a partitioned matrix lemma. This algorithm takes N_r iterations of the IRWLS procedure for each new sample. The number of initials samples (NI) must be as low as possible and contain at least one sample per class.

We have divided the samples in three sets to reduce the number equations in the IRWLS procedure. The division in sets is fully justified in [5], where the IRWLS procedure was developed for the standard SVC. The sets are defined according to the following rules: set s_1: samples whose $\beta_i y_i \in (0, Cc_i^n)$; set s_2: samples whose $\beta_i y_i = 0$; and set s_3: samples whose $\beta_i y_i = Cc_i^n$.

4 Computer experiments

Two experiments have been designed. The first one is a simple and well-know channel to show that, even in this case, the SVC is not able to give a good approximation if the channel varies. The second one deals with a realistic nonlinear channel model in which we show that the ASVC is able to adapt to the variations in it and out-performs classical adaptive equalization techniques.

4.1 Non-stationary linear channel

We have devised a simple non-stationary example where the recived symbols are obtained from the transmitted symbols by $r[n] = t[n] + a[n]t[n-1] + z[n]$, being

$$a[n] = \begin{cases} 0.5, & n < 200 \\ 0.5 - 0.005(n-200), & 200 \le n < 600 \\ -0.5, & 600 \le n < 800 \\ -0.5 + 0.005(n-800), & 800 \le n < 1200 \\ 0.5, & 1200 \le n < 1400 \end{cases}$$

and $z[n]$ is a zero mean additive white gaussian noise with $\sigma_z = 0.2$. This problem is well-known and can be linearly solved with a second order equalizer ($\mathbf{x}_n = [r[n] \quad r[n-1]]^T$ and $y_n = t[n]$). We have solved this problem using the standard SVC for each new sample, and the ASVC with $c_i^n = 0.99^{n-i}$, the first 50 samples were used as a training sequence and the remaining 1350 samples were use in decision directed mode (the equalizer outputs were fed back to itself). We have set $N_r = 1$, meaning that only a single iteration of the IRWLS per sample is necessary to keep track of the variations in the channel. For each sample we have computed the Bit Error Rate (BER) for the SVC and the ASVC and have plotted in Figure 1 the mean result for 20 independent trials, together with the Bayes error.

1. Initialization: compute $\mathbf{H}_{NI}$, set $s_1 = 1, \ldots, NI$, $s_2 = \emptyset$ and $s_3 = \emptyset$, and $\mathbf{a}_{s_1} = C$ and $k = 0$.
2. Compute: $M_{s_1,s_1}^{-1} = \begin{bmatrix} \mathbf{H}_{s_1,s_1} + \mathbf{D}_{\mathbf{a}_{s_1}}^{-1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1}$
$\begin{bmatrix} \boldsymbol{\beta}_{s_1} \\ b \end{bmatrix} = M_{s_1,s_1}^{-1} \left[\begin{bmatrix} \mathbf{D}_{\mathbf{y}_{s_1}} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{s_1,s_3}\boldsymbol{\beta}_{s_3} \\ \mathbf{1}^T\boldsymbol{\beta}_{s_3} \end{bmatrix} \right]$
3. Compute: $\mathbf{e}$ and $\mathbf{a}$ $\forall i \le n$.
4. Subsets reordering:
   (a) Move samples in s_1 with $\beta_i < 0$ to s_2.
   (b) Move samples in s_1 with $\beta_i = Cc_i^n y_i$ to s_3.
   (c) Move samples in s_3 with $e_i y_i < 0$ to s_2.
   (d) Move samples in s_2 with $e_i y_i \ge 0$ to s_1.
5. Set $k = k + 1$, if $k < N_r$ go to step 2.
6. Set $n = n + 1$. Obtain the n^{th} sample ($\mathbf{x}_n$,y_n). Compute $\mathbf{K}_n$ as in (6). Solve the linear equation system in (5). Set $k = 0$ and go to step 2.

Table 1: IRWLS-ASVC procedure.

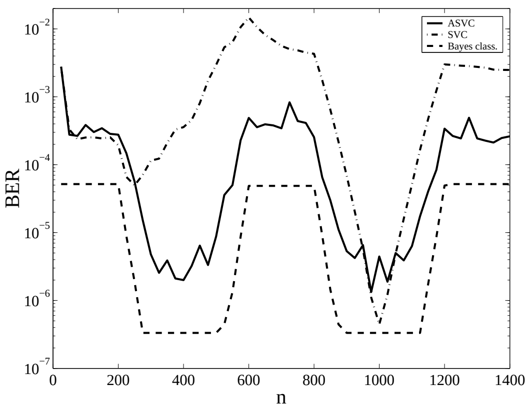

Figure 1: The BER is plotted for the ASVC (solid), the SVC (dash-dotted) and the Bayes equalizer (dashed).

We must point out that while the channel is invariant (the first 200 samples) the ASVC and the SVC give the same result but once the channel varies the SVC is not able to track its variations and ends fixing the equalizer to the mean value of the given data. In this case, the SVC solution is fixed around $a[n] = 0$, explaining why the SVC gives the best solution for $n = 1000$. The ASVC is able to track the channel changes as can be inferred from its error curve (parallel to Bayes error), and it uses an almost constant number of support vectors (SVs) (between 3 and 7). The SVC has an increasing number of SVs and ends with more than 30.

4.2 Non-stationary nonlinear channel

In order to test the ASVC in a more realistic environment we have taken a channel model from [2] in which a real radio channel was characterized by

$$r[n] = \sum_{j=0}^{k} a_j[n]t[n-j] + b[n]\left(\sum_{j=0}^{k} a_j[n]t[n-j]\right)^2 + z[n]$$

$a_1[n]$ and $b[n]$ were, respectively set to 0.87 and 0.2, as proposed in [2], and have varied $a_0[n] = a_2[n] = a[n]$:

$$a[n] = \begin{cases} 0.3, & n < 400 \\ 0.3 + 0.01(n-400), & 400 \le n < 450 \\ 0.35, & 450 \le n < 650 \\ 0.35 - 0.01(n-650), & 650 \le n < 750 \\ 0.25, & 750 \le n < 950 \\ 0.25 + 0.01(n-950), & 950 \le n < 1000 \\ 0.3, & 1000 \le n < 1200 \end{cases}$$

We have solved this task using a linear and a RBF[3] ASVC equalizers and have also solved it using the well-known RLS and LMS techniques. We use a 3^{rd} order equalizer ($\mathbf{x}_n = [r[n] \quad r[n-1] \quad r[n-2]]^T$ and $y_n = t[n-1]$). The confidence figure was $c_i^n = 0.995^{n-i}$, the width of the RBF kernel was set to 1 and $N_r = 1$ for both linear and nonlinear ASVC. The RLS and LMS adapting parameters were, respectively, set to 0.997 and 0.01. The value of $\sigma_z = 0.15$ and we used the first 100 samples as training sequence and from then on the equalizer operate in decision directed mode. We have plotted the BER for 20 independent trials in Figure 2.

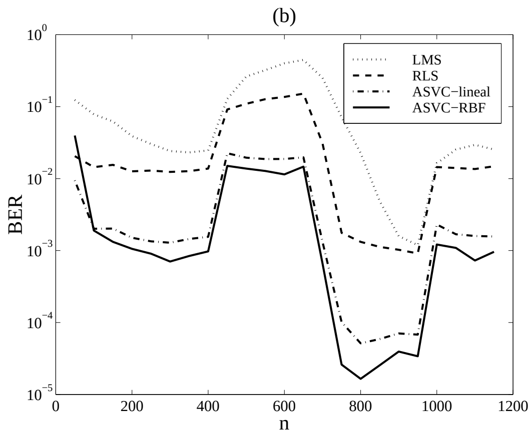

Figure 2: The Bit Error Rate for the proposed equalizing schemes are depicted

For this channel the linear ASVC, which presents the same complexity as the RLS, is able to out-perform both the RLS and LMS schemes. The squared error is not a good error measurement for nonlinear channels, meanwhile the maximum margin SVC property is able to performed well in every situation, explaining the gap in the BER between the linear ASVC equalizer and the LMS or RLS schemes. The ASVC with a nonlinear kernel is more versatile than the ASVC linear equalizer and it is able to further reduce its BER.

[3]The RBF kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$.

5 Conclusions

In this paper we have shown that the margin and penalty factor of the SVC can be interpreted as confidence over the training samples and we have shown that reducing it with time leads to a SVC that adapts to channel variations. Moreover, we have solved the ASVC using an IRWLS schemes that resembles the well-known RLS algorithm for adaptive signal processing, although it is able to attain maximum margin solution and higher generalization as a consequence.

References

[1] S. Chen, S. Gunn, and C. J. Harris. Decision feedback equalizer design using support vector machines. *IEE Proceedings Vision, Image and Signal Processing*, 147:213–219, 2000.

[2] G. Kechriotis, E. Zervas, and E. S. Manolakos. Using recurrent neural networks for adaptive communication channel equalization. *IEEE Transaction on Neural Networks*, 5(2):267–278, March 1994.

[3] J. Kivinen, A. Smola, P. Wankadia, and R. Williamson. Online learning with kernels. In *NIPS workshop: New Perspectives in Kernel-based Learning Methods*, Breckenridge, Colorado, Dec 2000.

[4] D. Mattera, F. Palmieri, and S. Haykin. Adaptive nonlinear filtering with the support vector method. In *Proc. of the EUSIPCO'98*, volume 2, pages 773–776, Rhodes, Greece, September 1998.

[5] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez. SVC-based equalizer for burst TDMA transmissions. *Signal Processing*, 81(8):1681–1693, Aug. 2001.

[6] L. L. Scharf. *Statistical Signal Processing.* Addison-Wesley, New York, 1990.

[7] B. Schoelkopf and A. Smola. *Learning with kernels.* M.I.T. Press, 2001.

[8] D. J. Sebald and J. A. Bucklew. Support vector machines techniques for nonlinear equalization. *IEEE Trans. on Signal Processing*, 48(11):3217–3226, Nov. 2000.