Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Estimation of Action Potential of the Cellular Membrane using Support Vectors Machines

Cesar Seijas [1], Antonino Caralli [1], *Member, IEEE* and Sergio Villazana [1]
[1]Universidad de Carabobo, Facultad de Ingeniería
Valencia, Venezuela, 2001, e-mail: cseijas@uc.edu.ve, acaralli@ieee.org, svillaza@uc.edu.ve

***Abstract*.- In this article an application of the Support Vectors Machines (SVM) is presented in the problem of the estimate of the action potential of the cellular membrane V, which is, a temporary function, highly non-linear, of the ionic concentrations of sodium and potassium. A model, for the estimate of V, is the Hodgkin-Huxley (HH) model that describes the dynamics of V similar to an electric circuit with passive elements representing the biochemical variables involved in the process. SVM are algorithms of emergent computation that have demonstrated an excellent performance in classification and regression applications; in this article they are used for the estimate of V and the result is compared with that obtained using HH, demonstrating that SVM are a promising alternative in modelling problem of biological processes.**

I. Introduction

THE modelling of the biological processes and in particular, the electrophysiological modelling is one of the problems of more challenge actually. Mathematical functions that relate the involved biochemical variables are highly non-linear, and specifically, in the case of the action potential of the cellular membrane V, are time-varying functions. Hodgkin-Huxley (HH) [1] is one of the models accepted that describes the dynamics of V similar to an electric circuit with passive elements that represent the biochemical variables involved in the process and current sources to describe any external current injected during the course of the experiment. In this article an application of the Support Vectors Machines is presented (SVM) in the problem of the estimate of the potential of action of the cellular membrane V.

SVM are algorithms of emergent computation developed by Vapnik and their collaborators [2] that have demonstrated an excellent performance in classification and regression applications [3], [4], [5], [6], [7], [8], [9]. In contrast to well known Artificial Neural Network (ANN), *SVM* implement the principle of structural risk minimization (SRM) that tries to minimize an upper bound of the generalization error instead of the empirical risk minimization (ERM) that minimizes the training error used in ANN. This induction principle is based on the fact that the generalization error is bounded by the sum of the training error and a confidence interval term depending of the Vapnik-Chervonenkis' dimension [2]. SRM allows to the SVM achieve an optimal network structure with a perfect balance between empirical risk and confidence interval VC. Another highlighting characteristic of the SVM is the training is equivalent to solve a quadratic programming problem with linear constraints that implies the solution reached by SVM is unique, optimal and without of local minimums in contrast to ANN that requires non-linear optimization during its training, with the risk to be trapped by local minimums. SVM were used originally in classification problems, but with the introduction of *e-insensitive* loss functions [2] they have been applied in non-linear regression problems [4], [8], [10].

The main objective of the present article is to evaluate the performance of SVM like estimator of V, what will be reflected in an important contribution in the modelling problem from biological data. This article is organized in the following way: section I is the introduction to the topic, section II describes in summary the HH model, section III shortly presents the theory of SVM in regression applications, section IV explains the design of the estimator of V that includes the tuning process of parameters of the SVM, and section V finally describes the experimental results and the conclusions.

II. Hodgkin-Huxley Model

The action potential of the cellular membrane (V), it is a temporary function, highly non-linear, of the ionic concentrations of sodium and potassium. A model, for the estimate of this potential, is the Hodgkin-Huxley model (HH) [1]. In HH the membrane is modelled like an electric circuit, in which, the total current through the membrane, Im, is calculated using the Kirchoff's law, as the sum of the currents due to ions of potassium (I_K), of sodium (I_{Na}) and a leakage component, IL that considers other ions that move passively for the membrane, besides an average capacitive, Cm (dV/dt), where Cm represents the time-dependent capacitance of the membrane. The equivalent circuit of the membrane is shown in Fig. 1, the equivalent circuit has three branches formed by the series combination of the conductance of the ion with to voltage source DC that represents the potential of ionic balance (equation of Nernst) [1], [11] and a fourth branch that models the capacitance of the membrane.

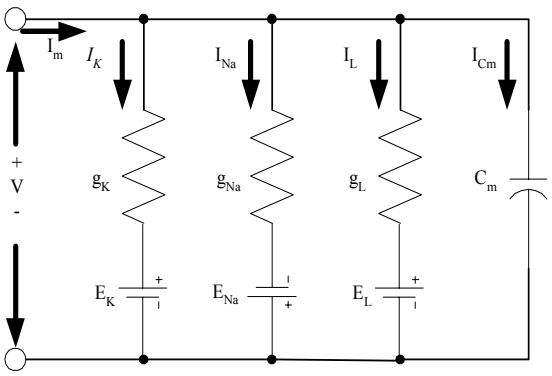

Fig.1. Circuital Model of the cellular membrane.

Applying the Kirchoff's Law to the circuit in Fig. 1:

$$I_m = C_m \frac{dV}{dt} + g_K(V-E_K) + g_{Na}(V-E_{Na}) + g_L(V-E_L) \quad (1)$$

Sodium and potassium conductances are non-linear functions of V and time, given by the following expressions:

$$g_{Na}(V,t) = g_0^{Na} m^3(V,t) h(V,t) \quad (2)$$

$$g_K(V,t) = g_0^K n^4(V,t) \quad (3)$$

Where the activation variables m and n and of inactivation h are given by:

$$m(t) = (m_0 - m_\infty) \times e^{\frac{-t}{\tau_m}} + m_\infty \quad (4)$$

The expressions of n and h are obtained substituting m for n and h in (4), for each function; the time constants in the same ones (with i = m, n, h for each time constant), they are in the way:

$$\tau_i = \frac{1}{\alpha_i + \beta_i} \quad (5)$$

And the αi and derived βi of experimental data:

$$\alpha_m = 0.10 \frac{V+35.0}{e^{-0.1(V+350)} - 1.0} \quad (6)$$

$$\beta_m = 4.0 \times e^{\frac{-(V+60.0)}{18.0}} \quad (7)$$

$$\alpha_h = 0.07 \times e^{-0.5(V+60.0)} \quad (8)$$

$$\beta_h = \frac{1.0}{\left(1.0 + e^{-1.0(V+30.0)}\right)} \quad (9)$$

$$\alpha_n = -\frac{0.01(V+50.0)}{\left(e^{-.1(V+50.0)} - 1.0\right)} \quad (10)$$

$$\beta_n = 0.125 \times e^{-0.0125(V+60.0)} \quad (11)$$

III. REGRESSION USING SVM

A *SVM* used as a regressor or *SVR* (Support Vector Regression) estimates a non-linear function using a set of linear functions defined in a hyperdimensional space. That is, for a set of data points $\mathbf{G} = \{(\mathbf{x}_i, d_i)\}_i^n$ (where $\boldsymbol{x}_i$ is the input vector, d_i is the expected output and n is the number of data patterns), *SVR* approximates the regression function utilizing:

$$y = f(x) = w\,\phi(x) + b \quad (12)$$

where $\phi(\boldsymbol{x})$ is the hyperdimensional feature space on the input space $\boldsymbol{x}$ is mapped (non linearly). Coefficients $\boldsymbol{w}$ and b are estimated minimizing:

$$R_{SVMs}(C) = C\frac{1}{n}\sum_{i=1}^{n} L_s(d_i, y_i) + \frac{1}{2}\|w\|^2 \quad (13)$$

where

$$L_s(d,y) = \begin{cases} |d-y| - \varepsilon & |d-y| \geq \varepsilon \\ 0 & \end{cases} \quad (14)$$

In the risk function given by (13), the term $C(1/n)\sum_{i=1}^{n} L_\varepsilon(d_i, y_i)$ is the empirical (risk) error while the term $(1/2).\|\boldsymbol{w}^2\|$ is the regularization term. Parameter C is known as regularized constant or capacity of the *SVM* and determines the trade off between the empirical risk and the regularization term. In (14), ε is known as size of the hyperdimensional cylinder that wraps the function and is equivalent to the approximation accuracy on the training data points. C and ε are parameters to be setting by the designer, in a tuning process during the training stage of the *SVM*.

To obtain the estimates of $\boldsymbol{w}$ and b, (14) is transformed to (15), using slack variables ξ_i y $\xi_i^{(*)}$ that represents upper and lower limits in the system output as is showed in Fig. 2, that is, minimizing

$$R_{SVMs}(w, \xi_i, \xi_i^{(*)}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^{(*)}) \quad (15)$$

Subject to the constraints:

$$\begin{aligned} d_i - w\phi(x) - b_i &\leq \varepsilon + \xi_i \\ w\phi(x) + b_i - d_i &\leq \varepsilon + \xi_i^{(*)}, \qquad \xi_i^{(*)} \geq 0 \end{aligned} \quad (16)$$

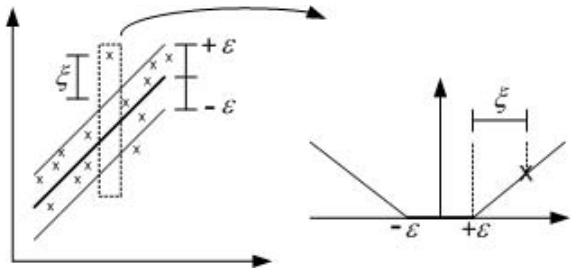

Fig. 2. Pre-established error ε and limits ξ in the ε-insensitive function

finally introducing the Lagrange multipliers a_i y a_i^* [2], the regression function given by (12) left as:

$$f\left(x,a_i,a_i^*\right)=\sum_{i=1}^{n}\left(a_i-a_i^*\right)K\left(x,x_i\right)+b \qquad (17)$$

$K\left(\mathbf{x_i},\mathbf{x_j}\right)$ is termed kernel function and a_i y a_i^* are the Lagrange multipliers meeting the constraints:

$a_i * a_i^* = 0,\ \ a_i \geq 0\ \ y\ \ a_i^* \geq 0,\ \ (i = 1,K\ ,n)$

and they are calculated maximizing the dual function of (11) having the form:

$$R\left(a_i,a_i^*\right)=\sum_{i=1}^{n} d_i\left(a_i-a_i^*\right)-\varepsilon\sum_{i=1}^{n}\left(a_i+a_i^*\right) \qquad (18)$$

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_i-a_i^*\right)\left(a_j-a_j^*\right)K\left(x_i,x_j\right) \qquad (19)$$

Subject to:

$$\sum_{i=1}^{n}\left(a_i-a_i^*\right)=0\ ,\ 0\leq a_i\leq C,\ \ 0\leq a_i^*\leq C,\ i=1,K\ ,n$$

from these Lagrange multipliers, based on the Karush-Kuhn-Tucker (*KKT*) [2] quadratic programming conditions only one certain number of them will have values non zero and the points or vector associated will have errors equal or greater than ε, these data are termed support vectors of the *SVM*. It is evident from (17) these support vectors define to ***f(x)***.

Kernel function is equal to inner product of two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_i)$, this is: $K\left(\mathbf{x_i},\mathbf{x_j}\right)=\phi\left(\mathbf{x_i}\right)*\phi\left(\mathbf{x_j}\right)$. The advantage of using kernel function, the computation is realized in an arbitrary feature space without explicitly using $\phi(x)$. Any function that satisfies the Mercer's conditions [2] is candidate to be a kernel function, among which can mention, polynomial functions of the form $K(x,y) = (x*y + 1)^d$, where d is the polynomial degree, gaussian $K(x,y) = exp(-1/\sigma^2(x - y)^2)$, where σ is the widespread coefficient of the gaussian, also known as radial basis function (*RBF*). From the viewpoint of the implementation, the training of the *SVM* is equivalent to solve a quadratic problem constrained linearly, that is, to solve a convex optimization problem.

IV. Design and Evaluation of the Estimator of V Based on SVR

A SVR was designed and built to estimate V using the toolbox of SVM developed by S. Gunn in the mathematical software MATLAB®, available for academic use [5]. The SVR constructed has two (2) inputs, one of them is the conductance of the Na (g_{Na}) and the other is the conductance of the K (g_K); it has an unique output that corresponds V, as indicated in Fig. 3. The output data of the simulation of HH model was used as training and validation data sets of the designed SVR. This simulation was realized using a software for simulation of the dynamics of the cellular membrane available for academic use [9], written in MATLAB®. Fig. 4 shows V generated when a current pulse step of duration 300 μsec. is applied to the cellular membrane, which polarizes toward +60 mV from the negative rest potential (-60 mV), until reaching the re-polarization again in an approximate time period of 10 msec., this sampled signal in 250 points constitutes the data matrix used for training and validation of the SVR. Input matrix (dimension: 250x2) is built with the values corresponding of the Na and K conductances (g_{Na} and g_K) of the same experiment, shown in Fig. 5

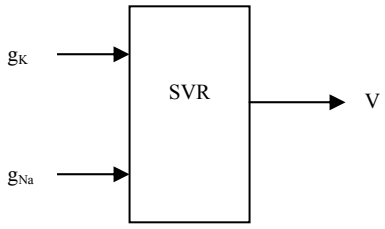

Fig.3. SVR to estimate V

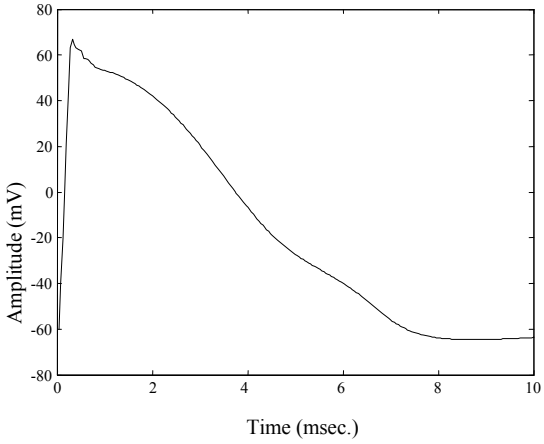

Fig.4. Intracellular Action Potential (V)

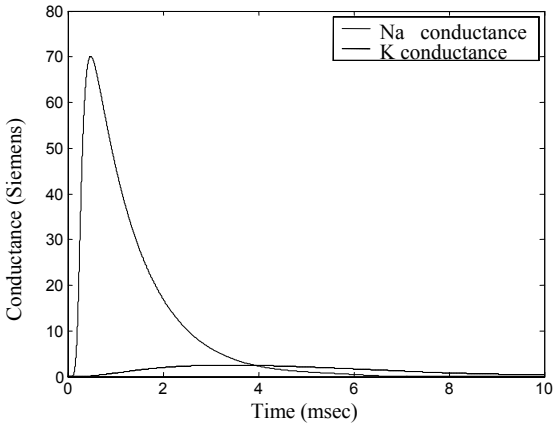

Fig.5. Na and K Conductances

Training input matrixes x and y were built with the first 150 vectors of the previous matrixes, this is:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} g_{Na}(150\times 1) \\ g_K(150\times 1) \end{bmatrix} \tag{20}$$

$$y = \left[V(150\times 1)\right] \tag{21}$$

Validation matrixes were built with the same previous strategy, but considering the following one hundred (100) data points.

V. Experimental Results

The first step in the design phase of a SVR is the selection of the kernel function to be used. Different functions were tested and the performance quality was measured of each one through the statistical metric: forecasting mean square error (FMSE), that is, calculated on a validation set, built in this case, with the ten (10) immediate data to those used in the training matrixes. This strategy tries to evaluate the generalization capacity of the SVR, results are shown in Table I.

TABLE I
Selection Of Kernel

Kernel Function	FMSE
Lineal	6.4598
Polinomial	3.5707
RBF	0.0027
Spline	0.5362

In this table can be observed the Kernel with better performance (smaller FMSE) is the RBF function, which was used in the implementation of the SVR. Now is necessary to define the parameters of the SVR: C, ε and σ. In the Table II is shown a summary of the experiments for the selection of the parameters.

TABLE II
Parameters of The SVR With Kernel RBF

σ	C	ε	T_e (sec)	NVS	FMSE
0.1	10	10^{-1}	87.2	136	16.8832
1.0	10^2	10^{-2}	88.0	148	0.7161
2.0	10^3	10^{-4}	86.0	150	0.2714
3.0	10^4	10^{-5}	87.7	150	0.0027

In the Table II, the columns of execution time and the number of support vectors (NVS) of the SVR are included By inspection of this table it was concluded that the best combination of parameters (or tuned parameters) it is:
[σ, C, ε] = [2.0, 104, 10-5]

Table III presents the results of the execution of the trained SVR, with input data coming from simulation with different current pulses of external excitation applied to the cellular membrane, and compared with those obtained in the HH model. The pulse duration was varied (ΔT) in each applied pulse.

TABLE III
SVR vs HH

T_e (sec)	NVS	ΔT (msec)	FMSE
85.6	150	0.50	0.0027
92.6	149	1.0	0.0031
87.5	150	2.0	0.0139
88.8	150	5.0	5.88×10^{-5}
84.3	150	10.0	1.42×10^{-5}

Results demonstrated the good performance of the SVR as estimator of V in the HH model. Fig. 6 illustrates the graphic comparison between the HH model and the estimate carried out by the SVR, where it is confirmed the excellent behaviour of the SVR model.

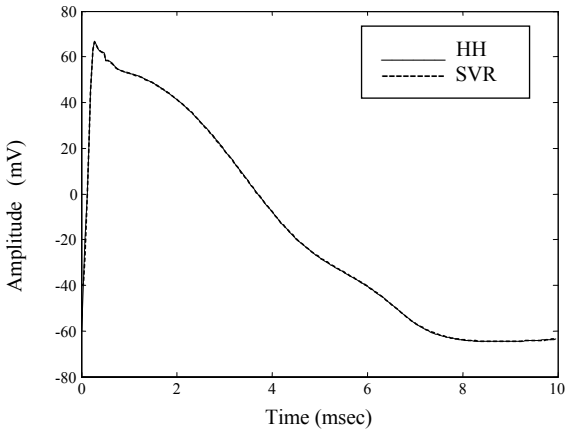

Fig.6. Action potentials of HH and SVR models

VI. Conclusions

SVR demonstrated an excellent performance as estimators of the action potential of the cellular membrane. The best performance of the trained SVR was obtained using the radial basis function (RBF). Results presented in this article express a promissory future to the SVM in the modelling area of biochemical and electrophysiological processes.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions of Aleida Aular for her work in the preparation of this document.

REFERENCES

[1] Gerstner and Kistler, *Spiking Neuron Models, Single Neurons, Populations, Plasticity*, Cambridge University Press, U.S.A., 2002.

[2] V. N. Vapnik,. *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.

[3] A. D. Back (1998, Aug.). Classification using Support Vector Machines. RIKEN Brain Science Institute, Wako-shi, Saitama, Japan [Online]. Available: http://andrewback.com/webpapers/svm/index.php

[4] T. Frontzek, T. Navin and R. Eckmiller (2001, Aug.). Predicting the Nonlinear Dynamics of Biological Neurons using Support Vector Machines with Different Kernels. Department of Computer Science VI, University of Bonn, F. R. Germany [Online]. Available: http://www.kyb.mpg.de/publications/pdfs/pdf1939.pdf

[5] S. Gunn. (1997, November 10). Support Vector Machines for Classification and Regression. Speech and Intelligent Systems Research Group, University of Southampton, U.S.A. [Online]. Available: http://www.isis.ecs.soton.ac.uk/research/svm/svm.html.

[6] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik (2002, Aug.). Gene selection for cancer classification using support vector machines. Machine Learning [Online]. Available: http://www.tsi.enst.fr/~campedel/Biblio/ FeatureSelection/Guyon_GeneSelection_2000.pdf

[7] Richman M. & T. Parks, Detection of Noses and Faces in Consumer Images, Center for applied Mathematics, Cornell University, U.S.A., 1998.

[8] Tay F. & L. Cao, Application of Support Vector Machines in Financial Time Series Forecasting, Department of Mechanical Engineering, National University of Singapore, Singapore, 2001.

[9] Drakos N. & R. Moore, Sequential Minimal Optimization for SVM, Computer Based Learning Unit, Leeds, 2001.

[10] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear Prediction of Chaotic Time Series using a Support Vector Machines," in *Neural Networks for Signal Processing VII-Proceedings of the 1997 IEEE Workshop*, pages 511-520.

[11] J. Peterson, "Biological Information Processing" in *A Simple Sodium - Potassium Gate Model*. [Online]. Available: http://www.ces.clemson.edu/~petersj/BioEng/SimpleGates/SimpleGates001.html, consulted 11/01/2003.

BIOGRAPHIES

César Seijas was born in Venezuela. Graduated in electrical engineering from the University of Carabobo. He received the MSc Electrical Engineering degree. Actually realizes doctoral studies in the Universidad de Carabobo. Interest in Biomedical Electronics, Power Electronics, Support Vector Machines (SVM), Applications of SVM in areas of Control and Industrial Automation.

Antonino Caralli. IEEE Member. Graduated in electrical engineering from the University of Carabobo, Venezuela. He received the MSc Biomedical Engineering and the PhD degrees. Interest in Biomedical Electronics, Support Vector Machines, and Solid State Electronics.

Sergio Villazana was born in Venezuela. Graduated in electrical engineering from the University of Carabobo in 1996. His professional experience was in the electronic equipment design. He realized postgraduate studies in the University of Carabobo and received the MSc Electrical Engineering degree. Actually realizes doctoral studies in the Universidad Central de Venezuela. Interest in Biomedical Electronics, Electrical Drives, Support Vector Machine, and Power Electronics