IMTC 2006 - Instrumentation and Measurement
Technology Conference
Salerno, Italy, 24–27 April, 2006

Data Uncertainty Sensitivity Analysis for Reduced Complexity SVM Classifiers

M. Gubian, A. Boni, D. Petri

DIT – Department of Information and Communication Technology
University of Trento, Italy

Abstract – *In this paper we investigate experimentally how different sources of uncertainty affect the classification performance of an SVM based binary classifier. Our aim is to find statistically sound methods for controlling the detrimental effects of such sources when a classifier is to be implemented in hardware platforms where severe limitations force designers to allocate power, computation and memory resources carefully. At a first analysis, SVM revealed robust in terms of noise on data, whereas training data scarcity is a problem to be investigated further on.*

Keywords – *Support Vector Machines (SVMs), model selection, smart sensors, noise robustness*

I. INTRODUCTION

Wireless Sensors Networks (WSNs) and other ubiquitous computing technologies are a promising and fast growing application area, mainly thanks to the recent advances in micro-electro-mechanical systems (MEMS), wireless communications, and digital electronics.

A WSN [1] is composed of a large number of small sized sensor nodes, each provided with sensing, data processing, and wireless communicating components, which cooperatively solve a given measurement task. The measurement and the environment vary across applications, including among others: remotely monitored physiological data of a patient; foreign chemical agents detection in water, air or soil; detection, recognition and tracking of objects (e.g. for military applications); monitoring of wildlife species and biocomplexity mapping. Among the advantages with respect to traditional sensor systems are the possibility to scatter sensors into inaccessible areas and leave them unattended and capability of each node to (pre)process data locally, avoiding the expensive transmission of raw data. On the other hand, these desirable features pose serious challenges to research and technology. Self-organizing capabilities of communication algorithms and Power-Aware (PA) design techniques are two of them. The latter is justified observing that the growth of chip elements density does not come with a comparable growth of energy density of batteries nor with remarkable improvements in power scavenging techniques.

A typical WSN measurement task includes classification, or more generally, pattern matching: a sensor network is deployed in order to detect and classify a predefined phenomenon, such as the occurrence of harmful environmental conditions for crops and livestock or a queue of cars starting to build-up on a highway. Usually classification algorithms are expensive in terms of computation and memory requirements. In order to implement a WSN for classification two independent constraints have to be taken into account:

- nodes are usually small in terms of processing capabilities and memory;
- the most power consuming activity for a node is data transmission and reception.

Given the above constraints, two contrastive approaches can be adopted:

1. perform centralized classification, i.e. into base stations, which collect poorly processed data from sensor nodes. This alleviates the computational load in each node at the expense of a great deal of power consumption for radio transmission from nodes to stations;
2. perform classification at node level. This is a PA approach that yields the challenge of finding lightweighted classification algorithms, together with sound methods for performance evaluation of such algorithms, in order to measure trade-offs between classification performance and computational costs at design time.

In this paper we are focusing on the latter approach [2].

Classification is one of the objects of study of Machine Learning (ML). Maximum Likelihood estimation, k-Nearest Neighbor, Neural Networks and Support Vector Machines (SVMs) are the most popular among the approaches based on Learning from Examples [3]. We chose to focus our analysis on SVMs, because they have some peculiarities which we find useful for the problem of designing lightweighted, robust and reliable classifiers [4] [5]. Such distinguishing characteristics are:

convexity: the classification problem is solved by the minimization of a convex quadratic function, hence avoiding the problem of local minima, which instead hampers Neural Networks [6];

data sparseness: the solution to the classification problem is expressed in form of a decision function that is a nonlinear combination of a (usually) small subset of training examples, i.e. of the data used for building the classifier. Such special training vectors are called *support vectors*.

The former point concerns reliability, whereas the latter can have strong consequences on implementation, because once the training phase has completed, the classifier itself consists of a relatively simple function of a selected number of training vectors, with a save in terms of complexity and memory usage with respect to other techniques such as Nearest Neighbor, which requires the storage of the whole training set [6].

Many works in the ML literature deal with performance assessment of different classification algorithms, but usually they focus mainly on classification performance rather than on implementation issues [7] – [10]. In particular, SVMs have been widely studied recently, and, among other topics, a great deal of attention is put into the objective of reducing the number of support vectors [11], an issue having also an undisputed practical relevance. Nevertheless, such investigations do not delve more into equally important implementation matters, such as noise robustness or quantization effects.

In this paper we have started to address such investigations. We performed a study on the solution space of SVMs, and on the effects of different classes of noise on that space. Our ultimate goal is to find quantitative criteria for determining to what extent the classification is to be considered reliable when we are going to face noisy conditions, because of the physical environment or because of fixed point representation of data. The application of such criteria may become a design tool for reduced complexity SVM based classifiers.

As a last remark, in order to be as objective as possible in our somewhat qualitative analysis, we performed statistical hypothesis tests whenever possible, following the guidelines suggested by Cohen [12] and Dietterich [13].

The paper is organized as follow: in section II we introduce SVMs, in section III we present our proposed approach and in section IV we display the results achieved up to now, together with a discussion about projected future results.

II. AN INTRODUCTION TO SVM

As for any Learning by Example algorithm, a SVM is a method for the estimation of a set of parameters based on a (ofter small) set of training examples. An optimality criterion is applied in order to find the best set of parameters which determine a function that is supposed to generalize some property about the overall distribution of data, i.e. able to predict such a property about any previously unseen example.

The formulation of the problem for the case of binary classification can be summarized as follows [3] [14] [15].

1. In order to solve non-linear classification problems, a non-linear function $\varphi(\cdot)$ maps the original input space into another dot product space – the *feature space* – with much higher dimensionality (even infinite).
2. In this space, a hyperplane that correctly separates the two classes in the training set is found according to the optimality principle of maximizing the minimum distance between the hyperplane and any training point (the *margin*).
3. In order to cope with non-separable cases and to penalize complex solutions that may lead to overfitting, a regularization term is included in the optimization problem, which allows to find solution with non-zero training error.
4. Exploiting a property of Hilbert spaces and the particular form of the function to be minimized, it is possible to formulate the problem without the explicit use of the map $\varphi(\cdot)$. Instead, a non-linear function $K(\boldsymbol{u}, \boldsymbol{v})$ called the *kernel* allows to express dot products between mapped vectors as follows:

$$\langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{1}$$

   where $\boldsymbol{x}_i$ is a data sample in the input space and $\langle\cdot\rangle$ is the dot product in the feature space.
5. A possible formulation of the problem, called C–SVM, is the following:

$$\begin{array}{l} \min\limits_{\boldsymbol{\alpha}} \left(\frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha}\right) \\ 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \\ \boldsymbol{y}^T \boldsymbol{\alpha} = 0 \end{array} \tag{2}$$

   for some $C > 0$. Here $Q = y_i y_j \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as in (1), y_i are class labels, N is the number of training examples and α_i are the parameters to be optimized. As pointed out in section I, a SVM solution is usually sparse, meaning that only a fraction of α_i will be greater than 0. Each α_i is associated with a training examples $\boldsymbol{x}_i$, thus only a subset of them, called the Support Vectors (SVs), will contribute to the final solution.
6. An alternative form of problem formulation is referred to as ν-SVM:

$$\begin{array}{l} \min\limits_{\boldsymbol{\alpha}} \left(\frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha}\right) \\ 0 \leq \alpha_i \leq 1, \quad i = 1, \ldots, N \\ \mathbf{1}^T \boldsymbol{\alpha} = \nu N \\ \boldsymbol{y}^T \boldsymbol{\alpha} = 0 \end{array} \tag{3}$$

   A nice property of the parameter $\nu \in (0, 1)$ is that it can be regarded as an upper bound on the fraction of allowed wrong classified training examples, and a lower bound on the fraction of SVs.

The solution to a SVM classification problem stated as above is not a single point, instead it usually spans a two-dimensional space. The reason is that:

1. solutions (2) or (3) depend on an arbitrary parameter (C and ν respectively);
2. different kernel function families exist, typically depending on one parameter. Popular kernels are the linear ($K(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{v}$), the Gaussian ($K(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\gamma \left\|\boldsymbol{u} - \boldsymbol{v}\right\|^2\right)$) and the polynomial ($K(\boldsymbol{u}, \boldsymbol{v}) = (1 + \boldsymbol{u} \cdot \boldsymbol{v})^p$) [15].

Since we are going to adopt a Gaussian kernel, our solution space Γ will be either (C, γ) or (ν, γ).

With *model selection* we generally intend the process aimed at determining values in the solution space Γ that optimize the behavior of the corresponding SVM. Optimality is usually expressed in terms of an estimate of the minimum classification error [16], and a regular grid search approach on Γ

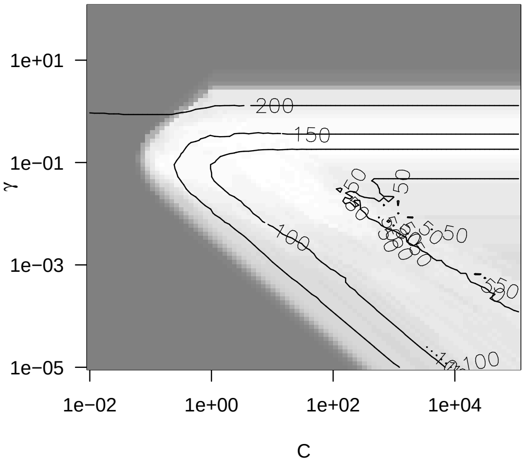


Fig. 1. UCI Ionosphere dataset. A full search in the (C, γ) SVM solution space on a 90 by 90 logarithmic grid. Accuracy is represented in gray levels from black to white, and contour lines display the corresponding number of SVs.

is performed. In the context of reduced complexity SVM design, an important issue to optimality is the number of SVs, which is linearly related to computation and memory requirements. This leads to a multi-objective approach, in which classification error and number of SVs are minimized simultaneously. Articulated solutions have been proposed; throughout our analysis we will adopt one of the simplest multi-objective approaches, the ϵ-constrained method [17] , which involves the minimization of a primary objective while expressing the other objectives in the form of inequality constraints. Our objective space is two-dimensional, hence there are only two ways for applying such principle: either minimize classification error imposing an upper limit on the number of SVs, or minimize the number of SVs without exceeding an upper limit on the classification error. We will apply both.

III. DATA UNCERTAINTY SENSITIVITY ANALYSIS

Our rationale, also inspired by simple qualitative inspections of the SVM solution space, can be summarized as follows:

- Figure 1 shows an example of solution space: here Γ is (C, γ), classification accuracy increases from darker to lighter regions, while contour lines display the number of SVs. The central white area is characterized by small variations of accuracy level and by variations in the number of SVs. Since this trend was found to be rather general, an approach based on a multi-objective search in Γ, where both a high accuracy and small number of SVs are searched for simultaneously, is likely to lead to advantageous trade-offs, i.e. it is possible to pay a little of accuracy for a large reduction in the number of SVs;
- we intend to investigate how different sources of uncertainty such as noise, quantization and training data scarcity affect optimal solutions, in order to adapt our search criterion to more realistic conditions imposed by hardware platforms.

Given the above informal statement of the problem, our analysis is organized as follows.

Our investigation is experimental: we analyzed both real and artificial problems. Real data are taken from the UCI repository [18], and we selected problems close to a typical sensor measurement situation (i.e. continuous rather than nominal features); artificial data have been also included because they do not bring all the drawbacks connected to data scarcity, as explained below. We investigated both $\Gamma = (C, \gamma)$ and $\Gamma = (\nu, \gamma)$ space, because even if the latter formulation offers a closer control on the number of SVs, we also need to carry out an exploratory analysis on how a given set of solutions degrade with noise, and the former formulation gives greater freedom to both our objectives (accuracy and number of SVs) to vary. Among many choices, we selected the following classes of noise:

- additive Gaussian noise on data. This is generally accepted for modelling a generic source of noise in data acquisition systems;
- quantization noise on data and on SVM α_i coefficients. This is intended to model in first approximation the effect of fixed point representation of numbers;

We carried out both a qualitative inspection of results and a quantitative assessment. The latter is organized as follows: we selected some representative conditions, like “clean”, “low noise”, “high noise”, and “few SVs”, “many SVs”. Then we performed pairwise statistical hypothesis tests in order to determine the degree of similarity between couples of factors. For example: compare “clean” vs. “low noise” given “few SVs” provides us a measure of the robustness of a solution with a reduced number of SVs with respect to a low level of noise.

IV. EXPERIMENTAL RESULTS

We performed a set of experiments using the LIBSVM tool [19] and real-world datasets taken from the UCI Machine Learning repository. These datasets are small (a few hundreds of examples), each example being composed by a vector of continuous valued features and a binary class label. Each dataset was split into Train and Test set (Tr/Te) in a 2:1 proportion, and care was taken in balancing class occurrence in the Train set and in having a minimum number of examples from both classes in the Test set. Five different of such Tr/Te splits were performed, and each experiment was repeated five times accordingly. After scaling all feature vector components to the range (-1,1) [20], additive Gaussian noise and quantization noise at different levels was applied, each level affecting all the features alike. For the Gaussian case, a different noise realization was generated for each Tr/Te split and for each SNR.

C–SVM was used and a grid search approach was adopted on $\Gamma = (C, \gamma)$. Our grid is 90 by 90 points, and each dimension is sampled in geometrical progression with a step of 1.2, because we wanted to span many orders of magnitude (the number 1.2 is a compromise solution after trying different values). A set of sub-optimal solutions was selected from each experiment by minimizing the classification error and imposing different upper bounds on the number of SVs (ϵ-constrained method, see Section II). Figure 2 shows the obtained results. Three factors vary: upper bound on the number of SVs, noise levels and Tr/Te random splits. All of them are measured w.r.t. the classification error on the test set. A first qualitative inspection reveals that test error can vary widely w.r.t. different Tr/Te random split realizations. Conversely it is much more stable w.r.t. the other two factors, as shown by the average lines.

The above shown effect is a quite interesting and harmful consequence of data scarcity hampering any Learning by Example algorithm, and in order to investigate it more it is useful to use artificial data, because they are virtually unlimited. Our objective is to find experimentally quantitative relations between the size of the Train set and the statistical behavior of accuracy. Real-world data will be then employed for validating such relationships, in the limits imposed by their intrinsic scarcity.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci *Wireless sensor networks: a survey* Computer Networks 38 (2002), 393–422.

[2] A.Boni, L. Gasperini, F. Pianegiani, and D. Petri, *Low-power and low-cost implementation of SVMs for SMART sensors*, Instrumentation and Measurement Technology Conference, Ottawa, Canada, 2005.

[3] T. Poggio and S. Smale, *The Mathematics of Learning: Dealing with Data*, Notices of the AMS, V. 50, N. 5, pp. 537–544, 2003.

[4] M. Duarte, Y. H. Hu, *Vehicle Classification in Distributed Sensor Networks*, Journal of Parallel and Distributed Computing, Vol. 64, No. 7, pp. 826-838, 2004.

[5] M. Pardo and G. Sberveglieri, *Learning From Data: A Tutorial With Emphasis on Modern Pattern Recognition Methods*, IEEE Sensors Journal, Vol. 2,N. 32,pp. 203–217, jun 2002.

[6] C.M.Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[7] David Meyer, Friedrich Leisch, Kurt Hornik *The support vector machine under test*, Neurocomputing 55 (2003) 169-186.

[8] Morawski, R.Z. Miekina, A. Barwicz, A., *Curve-fitting algorithms versus neural networks when applied for estimation of wavelength and power in DWDM systems*, IEEE Transactions on Instrumentation and Measurement, Oct. 2005.

[9] Yulei Jiang, *Uncertainty in the output of artificial neural networks*, IEEE Transactions on Medical Imaging, July 2003

[10] S. Mukkamala, G. Janoski, A. H. Sung *Intrusion Detection Using Neural Networks and Support Vector Machines* IEEE International Joint Conference on Neural Networks (IJCNN). (Honolulu, USA) 2002.

[11] Lin, K.-M. and C.-J. Lin *A study on reduced support vector machines*, IEEE Transactions on Neural Networks, 2003.

[12] Paul R. Cohen *Empirical Methods for Artificial Intelligence* The MIT Press - ISBN 0-262-03225-2 (HC). 1995

[13] Dietterich, T. G., *Approximate statistical tests for comparing supervised classification learning algorithms*, Neural Computation, 10 (7), 1998.

[14] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[15] B. Scholkopf, A. Smola, *Learning with kernels*, The MIT Press, 2002.

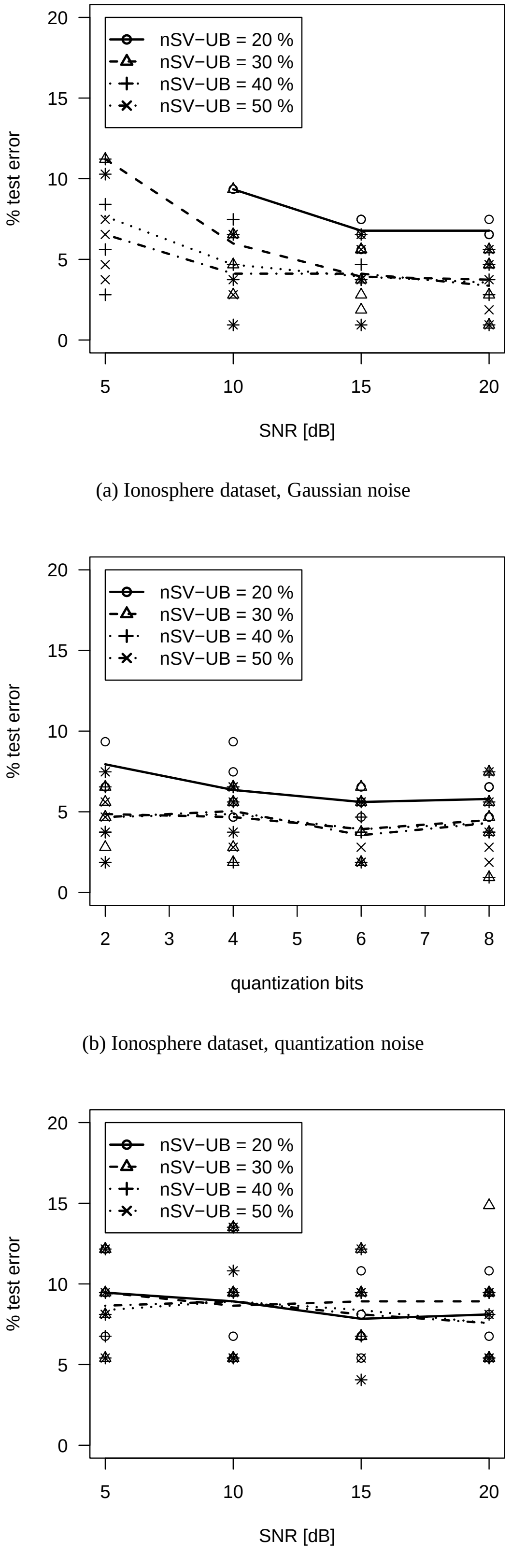
[16] D. Anguita, A. Boni, S. Ridella, F. Rivieccio, F. Sterpi, *Theoretical and Practical Model Selection Methods for Support Vector Classifiers*, Book Chapter in Lipo Wang (Editor), Support Vector Machines: Theory and Applications, Springer, 2005.

[17] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, 2001.

[18] C. Blake, C. Merz, *UCI repository of machine learning databases*, http://www.ics.uci.edu/ mlearn/MLRepository.html

[19] LIBSVM, SVM tool, http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[20] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin *A Practical Guide to Support Vector Classification*, http://www.csie.ntu.edu.tw/ cjlin/libsvm/



(a) Ionosphere dataset, Gaussian noise

(b) Ionosphere dataset, quantization noise

(c) Glass dataset, Gaussian noise

Fig. 2. Noise vs. test error rate (in percentage) for various Upper Bounds (UB) on the number of SVs (in train set percentages). Dots represent single Tr/Te splits (in different shapes), lines are averages over the 5 Tr/Te splits