Encyclopedia of Artificial Intelligence

Juan Ramón Rabuñal Dopico
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

Alejandro Pazos Sierra
University of A Coruña, Spain

Information Science REFERENCE

INFORMATION SCI
Hershey · New York

Director of Editorial Content: Kristin Klinger
Managing Development Editor: Kristin Roth
Development Editorial Assistant: Julia Mosemann, Rebecca Beistline
Senior Managing Editor: Jennifer Neidig
Managing Editor: Jamie Snavely
Assistant Managing Editor: Carole Coulson
Typesetter: Jennifer Neidig, Amanda Appicello, Cindy Consonery
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of artificial intelligence / Juan Ramon Rabunal Dopico, Julian Dorado de la Calle, and Alejandro Pazos Sierra, editors.
p. cm.
Includes bibliographical references and index.
Summary: "This book is a comprehensive and in-depth reference to the most recent developments in the field covering theoretical developments, techniques, technologies, among others"--Provided by publisher.
ISBN 978-1-59904-849-9 (hardcover) -- ISBN 978-1-59904-850-5 (ebook)
1. Artificial intelligence--Encyclopedias. I. Rabunal, Juan Ramon, 1973- II. Dorado, Julian, 1970- III. Pazos Sierra, Alejandro.
Q334.2.E63 2008
006.303--dc22
2008027245

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

EA Multi-Model Selection for SVM

Gilles Lebrun
University of Caen Basse-Normandie, France

Olivier Lezoray
University of Caen Basse-Normandie, France

Christophe Charrier
University of Caen Basse-Normandie, France

Hubert Cardot
University François-Rabelais of Tours, France

INTRODUCTION

Evolutionary algorithms (**EA**) (Rechenberg, 1965) belong to a family of stochastic search algorithms inspired by natural evolution. In the last years, EA were used successfully to produce efficient solutions for a great number of hard optimization problems (Beasley, 1997). These algorithms operate on a population of potential solutions and apply a survival principle according to a **fitness** measure associated to each solution to produce better approximations of the optimal solution. At each iteration, a new set of solutions is created by selecting individuals according to their level of fitness and by applying to them several operators. These operators model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. Although the basic idea of **EA** is straightforward, solutions coding, size of population, fitness function and operators must be defined in compliance with the kind of problem to optimize.

Multi-class problems with binary **SVM** (Support Vector Machine) classifiers are commonly treated as a decomposition in several binary sub-problems. An open question is how to properly choose all models for these sub-problems in order to have the lowest error rate for a specific SVM multi-class scheme. In this paper, we propose a new approach to optimize the **generalization capacity** of such SVM multi-class schemes. This approach consists in a global selection of models for sub-problems altogether and is denoted as **multi-model selection**. A multi-model selection can outperform the classical individual model selection used until now in the literature, but this type of selection defines a hard optimisation problem, because it corresponds to a search a efficient solution into a huge space. Therefore, we propose an adapted **EA** to achieve **that multi-model selection** by defining specific **fitness** function and **recombination operator**.

BACKGROUND

The multi-class classification problem refers to assigning a class to a feature vector in a set of possible ones. Among all the possible inducers, Support Vector Machine (SVM) have particular high generalization abilities (Vapnik, 1998) and have become very popular in the last few years. However, **SVM** are binary classifiers and several **combination schemes** were developed to extend **SVM** for problems with more two classes (Rifkin & Klautau, 2005). These schemes are based on different principles: probabilities (Price, Knerr, Personnaz & Dreyfus, 1994), error correcting codes (Dietterich, & Bakiri, 1995), correcting classifiers (Moreira, & Mayoraz, 1998) and evidence theory (Quost, Denoeux & Masson, 2006). All these **combination schemes** involve the following three steps: 1) decomposition of a multi-class problem into several binary sub-problems, 2) **SVM** training on all sub-problems to produce the corresponding binary decision functions and 3) **decoding strategy** to take a final decision from all binary decisions. Difficulties rely on the choice of the **combination scheme** (Duan & Keerthi, 2005) and how to optimize it (Lebrun, Charrier, Lezoray & Cardot, 2005).

In this paper, we focus on step 2) when steps 1) and 3) are fixed. For that step, each binary problem needs to properly tune the SVM hyper-parameters (model) in

order to have a global low multi-class error rate with the combination of all binary decision functions involved in. The search for efficient values of hyper-parameters is commonly designed by the term of *model selection*. The classical way to achieve optimization of multi-class schemes is an individual model selection for each related binary sub-problem. This methodology overtones that a multi-class scheme based on SVM combination is optimal when each binary classifier involved in that **combination scheme** is optimal on the dedicated binary problem. But, if it is supposed that a **decoding strategy** can more or less easily correct binary classifiers errors, then individual binary model selection on each binary sub-problem cannot take into account error correcting possibilities. For this main reason, we are thinking that another way to achieve optimization of multi-class schemes is a global **multi-model selection** for binary problems altogether. In fact, the goal is to have a minimum of errors on a muti-class problem. The selection of all sub-problem models (multi-model selection) has to be globally performed to achieve that goal, even if that means that error rates are not optimal on all binary sub-problems when they are observed individually. **EA** is an efficient meta-heuristic approach to realize that **multi-model selection**.

EA MULTI-MODEL SELECTION

This section is decomposed in 3 subsections. In the first section, the multi-model optimization problem for muti-class **combination schemes** is exposed. More details than in previous section and useful notations for next subsections are introduced. In the second section, our EA multi-model selection is exposed. Details on fitness estimation of multi-model and crossover operator over them are described. In the third section, experimental protocol and results with our EA multi-model selection are provided.

Multi-Model Optimization Problem

A multi-class combination scheme induces several binary sub-problems. The number k and the nature of binary sub-problems depend on the decomposition involved in the combination scheme. For each binary sub-problem, a **SVM** must be trained to produce an appropriate binary decision function h_i ($1 < i < k$). The quality of h_i is greatly dependent on the selected model θ_i and is characterized by the expected error rate e_i for new datasets with the same binary decomposition. Each model θ_i contains all hyper-parameters values for training a SVM on dedicated binary sub-problem. Expected error rate e_i associated to a model θ_i is commonly determined by cross-validation techniques. All the θ_i models constitute the multi-model $\theta = (\theta_1,..., \theta_k)$. The expected error rate e of a SVM multi-class combination scheme is directly dependent on the selected multi-model θ. Let Θ denote the multi-model space for a multi-class problem (*i.e.* $\forall\theta : \theta \in \Theta$) and Θ_i the model space for the i^{th} binary sub-problem. The best θ^* multi-model is the one for which expected error e is minimum and corresponds to the following optimization problem:

$$\theta^* = \arg\min_{\theta \in \Theta} e(\theta) \quad (1)$$

where $e(\theta)$ denotes the expected error e of a multi-class combination scheme with the multi-model θ. The huge size of the multi-model space ($\Theta = \times_{i \in [1,k]} \Theta_i$) makes the optimization problem (0.1) very hard. To reduce the optimization problem complexity, it is classic to use the following approximation:

$$\tilde{\theta} = \{\arg\min_{\theta \in \Theta} e(\theta_i) \mid i \in [1,k]\} \quad (2)$$

Hypothesis is made that

$$e(\tilde{\theta}) \approx e(\theta^*).$$

This hypothesis also supposes that

$$e(\tilde{\theta}_i) \approx e(\theta_i^*).$$

If it is evident that each individual model θ_i in the best multi-model θ^* must correspond to efficient **SVM** (*i.e.* low value of e_i) on the corresponding i^{th} binary sub-problem, all best individual models ($\theta_1^*,...,\theta_k^*$) do not necessarily define the best multi-model θ^*. The first reason is that all error rates e_i are estimated with some tolerance and combination of all these deviations can have a great impact on the final multi-class error rate e. The second reason is that even if all the binary classifiers of a **combination scheme** have identical e_i error rates for different multi-models, these binary classifiers can have different binary class predictions for a same

example according to the used multi-model. Indeed multi-class predictions by combining these binary classifiers could be different for a same feature vector example since the correction involved in a given **decoding strategy** depends on the nature of the internal errors of the binary classifiers (mainly, the number of errors). Then, multi-class classification schemes with the same internal-errors e_i, but different multi-models θ, can have different capacities of generalization. For all these reasons, we claim that multi-model optimization problem (0.1) can outperform individual model optimization (0.2).

Evolutionary Optimization Method

Within our AE multi-model selection method, a **fitness** measure f is associated to a multi-model θ which is all the more large as the error e associated to θ is small; this enables to solve (0.1) optimization problem. **Fitness** value is normalized in order to have f=1 when error e is zero and f=0 when error e corresponds to a random draw. Moreover, the number of examples in each class are not always well balanced for many multi-class datasets; to overcome this, the error e corresponds to a Balanced Error Rate (**BER**). As regards these two key points, the proposed fitness formulation is:

$$f = \frac{1}{1-\frac{1}{n_c}}\left(1-\frac{1}{n_c}-e\right) \quad (3)$$

with n_c denoting the number of classes in a multi-class problem. In the same way, the internal-fitness f_i is defined as $f_i = 1 - 2e_i$ for the i^{th} binary classifier with corresponding **BER** e_i.

The EA **crossover operator** for the combination of two multi-models θ^1 and θ^2 must favor the selection of most efficient models in these two multi-models. It is worth noting that one should not systematically select all the best models to produce an efficiency child multi-model θ as explained in previous sub-section. For each sub-problem, internal-fitness f_i^1 and f_i^2 are used to determine the probability

$$p_i = \frac{\left(f_i^1\right)^2}{\left(f_i^1\right)^2+\left(f_i^2\right)^2} \quad (4)$$

to select the i^{th} model in θ^1 as the i^{th} model in the child multi-model θ. f_i^j denotes the internal fitness of the i^{th} binary classifier with the multi-model θ^j. For the child multi-models generated by the **crossover operator**, an important advantage is that no new SVM training is necessary if all the related binary classifiers were already trained. In contrast, only the **BER** error rates of all child multi-models have to be evaluated. **SVM** Training is only necessary for the first step of the **EA** and when models go through a mutation operator.

The rest of our **EA** for **multi-model selection** is similar to other **EA** approaches. First, at initialization step, a population of λ multi-models is generated at random. Each model $\theta_i^j (1 \le i \le \lambda, 1 \le j \le k)$ corresponds to an uniform random within all possible values of SVM hyper-parameters. New multi-models are produced by combination of multi-models couples selected by a Stochastic Universal Sampling (**SUS**) strategy. A fixed selective pressure p_s is used for the **SUS** selection. Each model θ_i^j has a probability of p_m/k to mutate (uniform random as for the initialization step of EA). **Fitness** f of all child multi-models are then evaluated. A second selection step is used to define the population of the next iteration of our **EA**. λ individuals are selected by a **SUS** strategy (same selective pressure p_s is used) from both the λ parents and the λ children. Its become the multi-models population in the next iteration. The number of iterations of **EA** is fixed to n_{max}. At the end of the **EA**, the multi-model with the best **fitness** f from all these iterations is selected as θ^*.

Experimental Results

In this section, three well known multi-class datasets are used: Satimage (n_c = 6), Letter (n_c = 26) from the Statlog collection (Blacke & Merz, 1998), and USPS (n_c = 10) dataset (Vapnik, 1998). In (Wu, Lin & Weng, 2004), two sampling sizes of 300/500 and 800/1000 are used to constitute training/testing datasets. For each sampling sizes, 20 random splits are generated. We have used the same sampling sizes and the same split for the 3 datasets: Satimage, Letter and USPS. Two optimization methods are used for the selection of the best multi-model θ^* for each training datasets. The first one is the classical individual model selection and the second one is our **EA multi-model selection**. For both methods, two combination schemes are used:

Table 1. Average BER with individual model selection (column $\bar{e}_{classic}$) and our EA multi-model selection (column $\bar{e}_{EA}$). Negative values in column $\Delta\bar{e}$($\Delta\bar{e} = \bar{e}_{EA} - \bar{e}_{classic}$) correspond to an improvement of the performance of a multi-class combination scheme when our EA multi-model selection method is used.

Size	500			1000		
	$\bar{e}_{classic}$	$\bar{e}_{EA}$	$\Delta\bar{e}$	$\bar{e}_{classic}$	$\bar{e}_{EA}$	$\Delta\bar{e}$
one-versus-one						
Satimage	14.7 ± 1.8 %	14.5 ± 2.1 %	-0.2 %	11.8 ± 0.9 %	11.8 ± 1.0 %	-0.0 %
USPS	12.8 ± 1.2 %	11.0 ± 1.8 %	-1.8 %	8.9 ± 0.9 %	8.4 ± 1.6 %	-0.5 %
Letter	40.5 ± 3.0 %	35.9 ± 2.9 %	-4.6 %	21.4 ± 1.7 %	18.6 ± 2.1 %	- 2.8 %
one-versus-all						
Satimage	14.6 ± 1.7 %	14.5 ± 2.0 %	-0.1 %	11.5 ± 0.8 %	11.6 ± 1.0 %	+0.1 %
USPS	11.9 ± 1.3 %	11.2 ± 1.5 %	-0.7 %	8.8 ± 1.3 %	8.5 ± 1.6 %	-0.3 %
Letter	41.9 ± 3.3 %	36.3 ± 3.3 %	-5.6 %	22.1 ± 1.3 %	19.7 ± 1.8 %	-2.4 %

one-versus-one and *one-versus-all* (Rifkin & Klautau, 2004)[1]. For each binary problem, a SVM with Gaussian kernel $K(u,v) = \exp(-\gamma\|u - v\|^2)$ is trained (Vapnik, 1998). Possible values of SVM hyper-parameters for a model are C trade-off SVM constant (Vapnik, 1998) and widthband γ of gaussian kernel function ($\theta_i \equiv (C_i, \gamma_i)$). For all binary problems: $\theta_i \in \Theta_i = [2^{-5}, 2^{-3},...,2^{15}] \times [2^{-5}, 2^{-3},...,2^{15}]$. Individual space model Θ_i is based on grid search techniques (Chang & Lin, 2001). **BER** e on a multi-class problem and **BER** e_i on binary sub-problems are estimated by five-fold **cross-validation** (CV). These **BER** values are used by our **EA** for the **multi-model selection**. Final **BER** e of a selected multi-model by our **EA** is estimated on a test datasets not used during the **multi-model selection** process. Our EA has several constants that must be fixed and we have made the following choices: $p_s = 2$, $\lambda = 50$, $n_{max} = 100$, $p_m = 0.01$.

Table 1 gives average **BER** under all 20 split sets of previously mentioned datasets for each training set size (row size of table 1). This is done for the two combination schemes (*one-versus-one* and *one-versus-all*), and for the two above mentioned selection methods (columns $\bar{e}_{classic}$ and $\bar{e}_{EA}$). Column $\Delta\bar{e}$ provides the average variation of **BER** between our **multi-model selection** and classical one. Results of that column are particularly important. For two datasets (USPS and Letter) our optimization method produces SVM **combination schemes** with best **generalization capacities** than the classical one. That effect appears to be more marked when number of classes in the multi-class problem increases. A reason is that the multi-model space search size exponentially increases with the number k of binary problems involved in a **combination scheme** (121^k for those experiments). This effect is directly linked to the number of classes n_c and could explain why improvements are not measurable with Satimage dataset. In some way, a classical optimization method explores the multi-model space Θ in blink mode, because cumulate effect of the combination of k **SVM** decision functions could not be determined without estimation of e. That effect is emphasized when estimated **BER** e_i are poor (*i.e.* training and testing data size are low). Comparison of $\Delta\bar{e}$ values when training/testing dataset size change in table 1 illustrates this one.

FUTURE TRENDS

The proposed **EA multi-model selection** method has to be tested with other **combination schemes** (Rifkin & Klautau, 2004), like error-correcting output codes in order to measure their influence. Effect with others datasets, which have a great range in number of classes,

must also be tested. Adding feature selection (Fröhlich, Chapelle & Schölkopf, 2004) abilities to our AE muti-model selection is also of importance.

Another key point to take into account is the reduction of the learning time of our **EA** method which is actually expensive. One way to explore this is to use fast **CV** error estimation technique (Lebrun, Charrier, Lezoray & Cardot, 2006) for the estimation of BER.

CONCLUSION

In this paper, a new **EA multi-model selection** method is proposed to optimize the generalization capacities of SVM **combination schemes**. The definition of a **cross-over** operator based on internal **fitness** of **SVM** on each binary problem is the core of our **EA** method. Experimental results show that our method increases the **generalization capacities** of *one-versus-one* and *one-versus-all* **combination schemes** when compared with individual model selection method.

REFERENCES

Beasley, D. (1997). *Possible applications of evolutionary computation.* Handbook of Evolutionary Computation. 97/1, A1.2. IOP Publishing Ltd. And Oxford University Press.

Blacke, C., & Merz, C., (1998). *UCI repository of machine learning databases. Advances in Kernel Methods, Support Vector Learning.* University of California, Irvine, Dept. of Information and Computer Sciences.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.* Software available at http://www.csie.ntu.edu.tw/\~cjlin/libsvm.

Dietterich, T. G., & Bakiri, G. (1995). *Solving Multiclass Learning Problems via Error-Correcting Output Codes.* Journal of AI Research. (2) 263-286.

Duan, K.-B., & Keerthi, S. S. (2005). *Which Is the Best Multiclass SVM Method? An Empirical Study.* Multiple Classifier Systems. 278-285.

Fröhlich, F., Chapelle, O., & Schölkopf, B. (2004). Feature Selection for Support Vector Machines Using Genetic Algorithms. International Journal on Artificial Intelligence Tools. 13(4) 791-800.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2005). *Fast Pixel Classification by SVM Using Vector Quantization, Tabu Search and Hybrid Color Space.* Computer Analysis of Images and Patterns. (LNCS, Vol. 3691) 685-692.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2006). *Speed-up LOO CV with SVM classifier.* Intelligence Data Engineering and Automated Learning. (LNCS, Vol. 4224) 108-115.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2007). *An EA multi-model selection for SVM multiclass schemes.* Computational and Ambient Intelligence. 260-267 (LNCS, Vol. 4507).

Moreira, M., & Mayoraz, E. (1998). *Improved Pairwise Coupling Classification with Correcting Classifiers.* European Conference on Machine Learning. 160-171.

Price, D., Knerr, S., Personnaz, L., & Dreyfus, G. (1994). *Pairwise Neural Network Classifiers with Probabilistic Outputs.* Neural Information Processing Systems. 1109-1116.

Quost, B., Denoeux, T., & Masson, M. (2006). *One-against-all classifier combination in the framework of belief functions.* Information Processing and Management of Uncertainty in Knowledge-Based Systems. (1) 356-363.

Rechenberg, I. (1965). *Cybernetic Solution Path of an Experimental Problem.* Royal Aircraft Establishment Library Translation.

Rifkin, R., & Klautau, A. (2004). *In Defense of One-Vs-All Classification*. Journal of Machine Learning Research. (5) 101-141.

Vapnik, V.N. (1998). *Statistical Learning Theory*. Wiley Edition.

Wu, T.-F., Lin, C.-J., & Weng, R. C., (2004). *Probability Estimates for Multi-class Classification by Pairwise Coupling.* Journal of Machine Learning Research. (5) 975-1005.

KEY TERMS

Cross-Validation: A method of estimating predictive error of inducers. Cross-validation procedure splits

that dataset into k equal-sized pieces called folds. k predictive function are built, each tested on a distinct fold after being trained on the remaining folds.

Evolutionary Algorithm (EA): Meta-heuristic optimization approach inspired by natural evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a sub-optimal solution which is close to the optimal one.

Model Selection: Model Selection for Support Vector Machines concerns the tuning of SVM hyper-parameters as C trade-off constant and the kernel parameters.

Multi-Class Combination Scheme: A combination of several binary classifiers to solve a given multiclas problem.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Support Vector Machine (SVM): SVM maps input data in a higher dimensional feature space by using a non linear function and finds in that feature space the optimal separating hyperplane maximizing the margin (that is the distance between the hyperplane and the closest points of the training set) and minimizing the number of misclassified patterns.

Trade-Off Constant of SVM: The trade-off constant, noted C, permit to fix the importance to increase the margin for the selection of optimal hyper-plan in comparison with reducing predictive errors (i.e. examples which not respect margin distance from hyper-plan separator).

ENDNOTE

[1] More details on used combinations schemes are given in (Lebrun, Lezoray, Charrier & Cardot, 2007).