

Synthesis of digital circuits with ability to generalize

WILIAN SOARES LACERDA¹
ANTÔNIO DE PÁDUA BRAGA²

¹UFLA - Federal University of Lavras
DCC - Department of Computer Science
P.O. Box 3037 - Campus da UFLA 37200-000 - Lavras (MG) - Brazil
lacerda@dcc.ufla.br

²UFMG - Federal University of Minas Gerais
DELT - Department of Electronic
Av. Antônio Carlos 6627 - 31270-901 - Belo Horizonte (MG) - Brazil
apbraga@cpdee.ufmg.br

Abstract. A method for synthesis of digital hardware classification circuits is presented in this paper. The method works by first selecting data from the truth table that are important for generalization of the circuit. The selected subset is provided to a Boolean minimization algorithm (Espresso) that, by hypercube expansion, generates a classifier with a smoother separation surface between classes. The results show that obtained circuits have a generalization performance comparable to Support Vector Machines and Multilayer Perceptron.

Keywords: synthesis, digital circuit, generalization.

(Received January 01, 2010 / Accepted May 20, 2010)

1 Introduction

In general, the synthesis of digital circuits aims to find the circuit with fewer gates (logic minimization) and serving the truth table of binary data. Thus, the generalization ability of the circuit is not an objective of minimizing Boolean. In this paper a method for hardware implementation of digital classifier circuits with good generalization is proposed. This method is based on the idea of separation margin maximization with prior sample selection according to the k nearest neighbor (KNN) rule [4].

Large margin classifiers [12] result on smoother and unbiased separation surfaces, yielding improved generalization performance. The approach presented in this paper aims at obtaining large margin classifiers by taking advantage of hypercube expansion of Boolean minimizing algorithms as described next. The performance of the proposed method is compared with popular al-

gorithms of Artificial Neural Networks [8] and Support Vector Machines [15].

2 The proposed method

An overview of the steps required to generate the final circuit is presented in Figure 1. Initially, a subset of patterns just off the separation margin between the classes (outputs 0 and 1) is selected from the initial truth table. The algorithm works by first selecting a seed subset that corresponds to those patterns within the separation margin. The final truth table is then formed by the k nearest neighbors of the seed patterns, not including any pattern within the margin.

The value of k determines the size of the final truth table, which corresponds to a subset of lateral patterns just off the separation margin. The higher the k value, the greater the size of the lateral data. The lateral patterns are then provided to the Boolean minimizing that

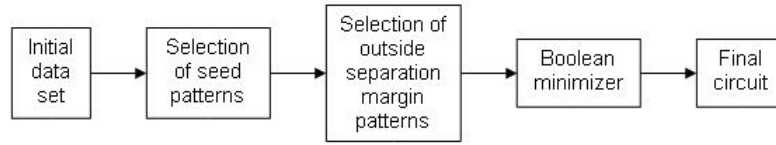


Figure 1: Overview of the subset selection method for circuit generation.

generates the final circuit. The hypercube expansion of Boolean minimizing such as Espresso or Quine-McCluskey [3] applied to the lateral patterns results on the final separation surface between the classes.

The final circuit tends to be a large margin classifier, since the expansion occurs concurrently in both sides of the margin and the effect of noisy patterns is minimized in the selected data. If the whole initial training set is provided directly to the Boolean minimizing without subset selection, a poor generalization circuit is obtained, since in this situation the hypercube expansion occurs only within the classes, not in the margins. In the method described in this paper, the margin patterns are not included in the final training set, what allows an even expansion of hypercubes from the lateral patterns in the direction of the margins.

3 Sample selection

This section explains the method for selecting data for good generalization. Suppose a data set Ω with samples z (truth table entries), divided into two subsets (outputs 0 and 1 from the truth table):

$$X = \{x_0, x_1, \dots, x_n\}, \text{ e } Y = \{y_0, y_1, \dots, y_m\}$$

An algorithm for selecting the samples z ($z \in X$, or $z \in Y$), named here RRS (Reduced Remaining Subset), is presented:

1. Select the samples that are closer to the separation boundary [6].

For each sample x_i ($0 \leq i \leq n$), find the sample y_j ($0 \leq j \leq m$) so that the distance $d(x_i, y_j)$ is minimum. Sort the distances $d(x_i, y_j)$ in ascending order. Add, iteratively, to a initially empty data set Θ , each pair (x_i, y_j) with smallest distance until the data set Θ classifies correctly all the z samples from Ω using the simple nearest neighbour rule:

If $\min[d(z, x_i)] < \min[d(z, y_j)] \forall x_i, y_j \in \Theta$ and $\forall z \in \Omega$,

- then $z \in X$,

- else $z \in Y$.

The final data set Θ is consistent [7].

2. Obtaining the reduced consistent subset Θ_r [5].

For each sample in the consistent data set Θ it is verified whether it is essential to maintain the consistency of Θ . Considering initially that $\Theta_r = \Theta$, for every pair $(x_i, y_j) \in \Theta_r$, with largest distance $d(x_i, y_j)$:

- If the data set $\{\Theta_r - x_i\}$ correctly classifies every $z \in \Omega$
then: $\Theta_r = \{\Theta_r - x_i\}$
- If the data set $\{\Theta_r - y_j\}$ correctly classifies every $z \in \Omega$
then: $\Theta_r = \{\Theta_r - y_j\}$

At the end of this step, the remaining subset ζ [16] is defined as $\zeta = \Omega - \Theta_r$.

3. Obtaining the reduced remaining subset ζ_r .

The reduced subset ζ_r is obtained by selecting the samples of the subset ζ that are closer to Θ_r . For every x_i and $y_j \in \Theta_r$, k samples x_p and k samples y_q ($x_p, y_q \in \zeta$) are selected in such a way that $d(x_i, x_p)$, $d(x_i, y_q)$, $d(y_j, x_p)$ and $d(y_j, y_q)$ are minimum. The value of k is set by the user or it can be tuned iteratively by the algorithm based on the output performance.

The following distance metrics were used in this algorithm: Discriminating Distance [1], VDM (value difference metric) [13], Overlap [11], Bayes [2], and others metrics as proposed in [10].

4 Results

Artificial binary data was generated for two class problem, having two orthogonal [9] 24 bits long vectors as seeds for the classes. The data set was generated by varying a fixed number of bits in each one of the two

seed vectors, what resulted on 2980 samples. In Figure 2 is displayed the histogram of the data generated for each class, taking into account the frequency which each bit is activated (logic "1"). It is observed that there is a trend in preference for activation of some bits, different for each class. For the data generated for the class "0", the bits between bit 5 and bit 18 have a higher frequency of activation. For the data generated for the class "1", the bits close to 12 have a higher frequency of activation.

The order of storage of samples is random and they can be of any class. Of the total samples, 2384 (80%) were used as reference for the classifier circuit (training), and 596 (20%) were reserved for test. In order to obtain average values in the graphs that follow, the generation of samples was repeated 10 times.

The graph in Figure 3 indicates the average number of selected samples to compose the remaining small subset for each value of k , in each metric used. It is observed that as the parameter k grows, the number of selected samples also increases, obviously. The overlap metric showed a higher rate of growth of samples function of k , while the metric Bayes had the lowest rate. The selected data subset is used by Espresso to obtain logic function of the digital circuit classifier reducing the footprint and consumption of the circuit generated.

The graph in Figure 4 shows the average number of correct classification test data obtained by simulation of the digital circuit obtained from RRS, for the 10 experiments and for each of the 6 metrics tested. It is observed from the graph that the average number of correct answers of classification initially increases from $k = 1$ independent of the metric used, and after a certain value for k the chart tends to stabilize. Best result is obtained for $k = 7$ and using the modified VDM metric. The worst performance of the circuit was obtained using the Bayesian metric (for low values of k , $k < 5$) and overlap (for higher values of k , $k > 5$).

The Figure 5 shows the average number product terms obtained in function of the parameter k . The metric overlap had the highest index product terms in function of k , which explains the low performance of this metric in the index hit the circuit shown in previous figure. The modified VDM metric, in turn, had the lowest rate product terms in function of k , which explains its greater capacity for generalization, i.e., a higher rate of correct answers for the test data.

The performance of the circuits generated by the RRS method was compared with other approaches, such as the Multilayer Perceptron [4] trained with the multi-objective learning algorithm [14] and Support Vector Machines [15]. In order to compare the proposed method

with the other training approaches, MLP and SVM were used to reclassify the training data (truth table) that was then provided as an argument to Espresso in order to generate the digital classifier circuits. The reclassification is necessary in order to provide the data set with the functions learned by MLP and SVM and to generate the final circuits with the specific characteristics inherited from each one of the learning machines.

Table 1 shows the results obtained by the different methods for generating the digital circuits for the 10 different artificial data sets. The table shows the number of samples used to generate the digital circuit by Espresso, the time took by Espresso in seconds, the mean number of product terms in Boolean function, and the correct classification performance for each method. The values are in the form of medium and deviation. As can be observed, the RRS method, using $k = 7$ and distance metric modified VDM (VDMm), results on similar performance when compared with the others methods, but with a smaller number of samples. Because the RSR method selects a smaller amount of samples as input to Espresso program, it required significantly less time processing when compared with other methods. The proposed method also generates the digital circuit with less product terms for the Boolean function, which saves logic gates in the implementation of digital circuit, reducing the footprint and consumption of the circuit generated.

5 Conclusions

The proposed method for synthesis of digital hardware classification circuits described in this paper yields circuits with generalization performance compatible with popular pattern classification approaches, such as SVMs and MLPs. The method takes advantage of hypercube expansion of Boolean minimize algorithms in order to generate a smooth separation surface between classes and requires less user intervention than other machine learning approaches.

The classifier digital circuit designed by the proposed selected data RSR method has performance rating similar to the best classification methods used for the corresponding circuit, but with less training data. Some tests were implemented using artificial data, whose results confirm the observations. Moreover, the use the proposed method required less computational effort for minimizing Boolean function (Espresso) due to the smaller number of samples training.

The proposed method has a high computational cost, because it is derived from Nearest Neighbor Rule. The complexity is due mainly to calculation of distances between all training samples to choose the nearest neigh-

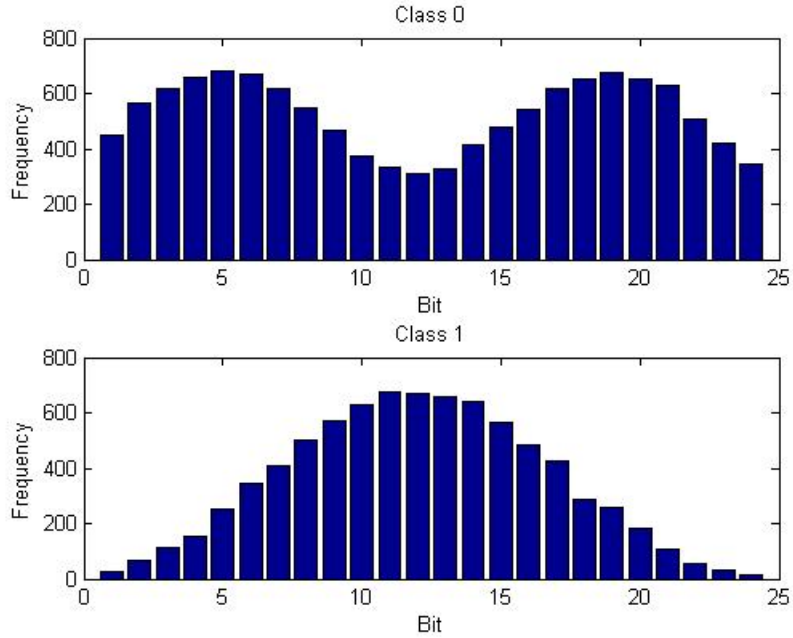


Figure 2: Histogram of the synthetic binary data.

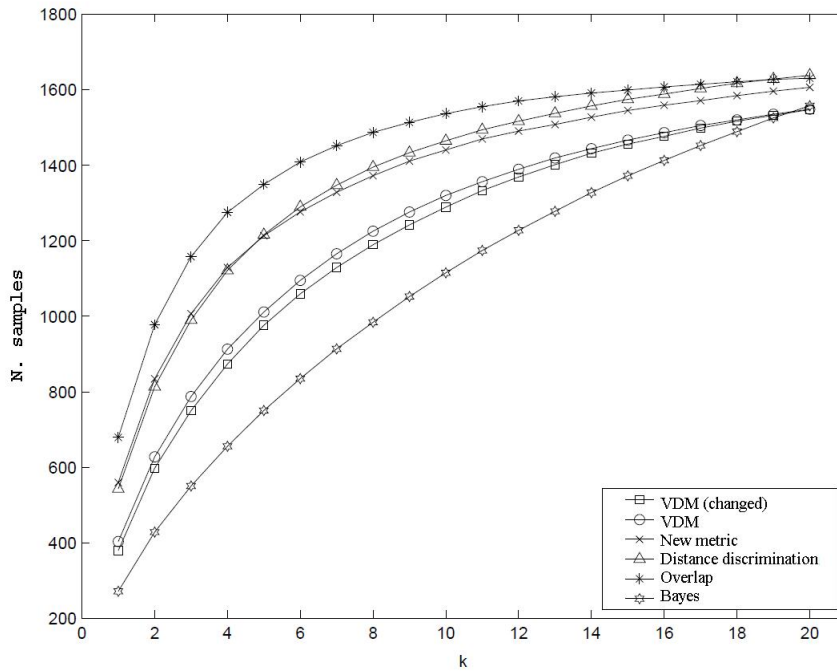


Figure 3: Selected subset mean size ζ_r as a function of k.

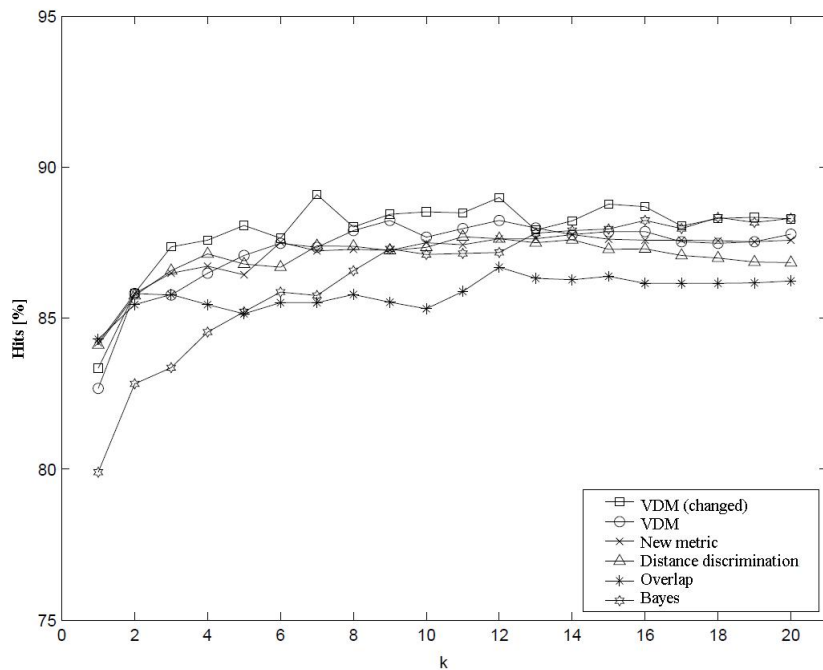


Figure 4: Mean performance of the final circuit classification as a function of k.

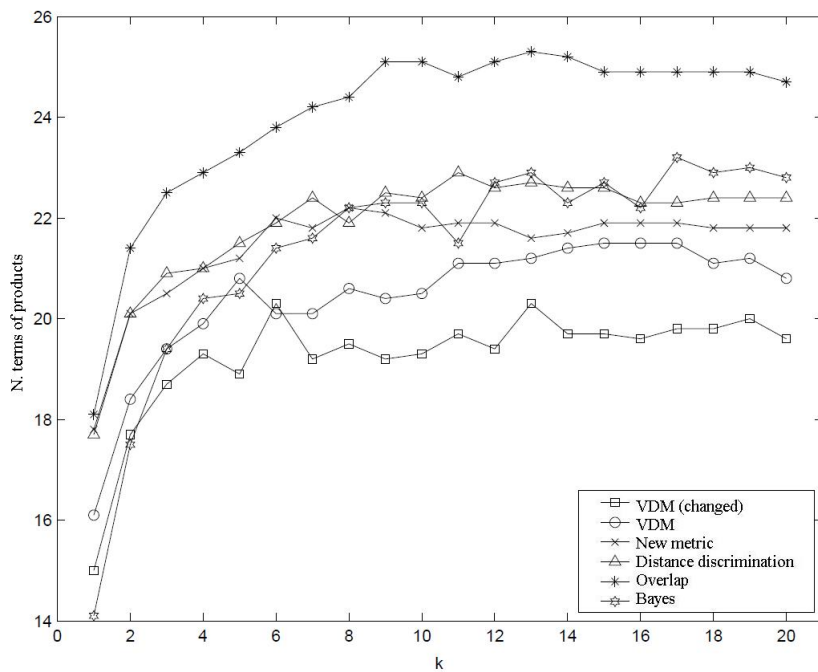


Figure 5: Number of product terms as a function of k.

Table 1: Comparison of the variations of the proposed method with SVMs and MLPs.

Method	Number of samples	Espresso [s]	N. products	Hit [%]
Without selection	2380 (100.00%)	2 ± 0.5	65 ± 3	84.0 ± 1.6
RRS ($k = 7$, VDMm)	1129 ± 44 ($47.4 \pm 1.8\%$)	< 1	19 ± 1	89.1 ± 1.2
MLP (32 neurons)	2380 (100.00%)	6 ± 1	30 ± 2	89.1 ± 1.1
SVM (rbf,w=10,c=1)	2380 (100.00%)	5 ± 1	31 ± 1	88.6 ± 0.8

bors. Moreover, the method uses a process of ordering the values of the distances calculated (Quicksort was used) to decision making. The complexity of the method is also increased by the 3 different steps performed in sequence, using iteration for each step. Despite this, the largest bottleneck for determination of the digital circuit is the function Boolean minimizing (Espresso).

References

- [1] Aleksander, I., Clarke, T. J. W., and Braga, A. P. Binary neural systems: combining weighted and weightless properties. *Intelligent Systems Engineering*, 3(4):211–221, 1994.
- [2] Blanzieri, E. and Ricci, F. Probability based metrics for nearest neighbor classification and case-based reasoning. *Lecture Notes in Computer Science*, 1650:14–29, 1999.
- [3] De Micheli, G. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill International, Cingapura, 1994. 579 p.
- [4] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley & Sons, New York, 2000. 654 p.
- [5] Gates, G. W. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18(3):431–433, 1972.
- [6] Gowda, K. C. and Krishna, G. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Transactions on Information Theory*, 25(4):488–490, July 1979.
- [7] Hart, P. E. The condensed nearest neighbor. *IEEE Transaction on Information Theory*, IT(14):515–516, 1968.
- [8] Haykin, S. *Redes Neurais: Princípios e Prática*. Bookman, Porto Alegre, 2^a edition, 2001. 900 p.
- [9] Kanerva, P. *Sparse Distributed Memory*. A Bradford Book, London, 1988. 155 p.
- [10] Lacerda, W. S. *Projeto e implementação de circuitos classificadores digitais com controle da generalização baseado na regra do vizinho-mais-próximo modificada*. PhD thesis, Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte, February 2006. 192p.
- [11] Liu, W. Z. and White, A. P. Metrics for nearest neighbour discrimination with categorical attributes. In *Research and Development in Expert Systems XIV: Proceedings of the 17th Annual Technical Conference of the BCES Specialist Group*, pages 51–59, 1997.
- [12] Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, C. *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 1999.
- [13] Stanfill, C. and Waltz, D. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213 – 1228, December 1986.
- [14] Teixeira, R. A. *Treinamento de Redes Neurais Artificiais Através de Otimização Multi-objetivo: Uma Nova Abordagem para o Equilíbrio entre a Polarização e a Variância*. PhD thesis, Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte, agosto 2001. 144 p.
- [15] Vapnik, V. N. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998. 736p.
- [16] Wilson, D. L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE - Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, July 1972.