ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Credit analysis with a clustering RAM-based neural classifier

Douglas O. Cardoso[1], Danilo S. Carvalho[1], Daniel S. F. Alves[1],
Diego F. P. Souza[1], Hugo C. C. Carneiro[1],
Carlos E. Pedreira[1], Priscila M. V. Lima[2] and Felipe M. G. França[1] *

1 - COPPE, 2 - iNCE, Universidade Federal do Rio de Janeiro, BRAZIL

Abstract. Datasets with a large amount of noisy data are quite common in real-world classification problems. Robustness is an important characteristic of state-of-the-art classifiers that use error minimization techniques, thus requiring a long time to converge. This paper presents ClusWiSARD, a clustering customization of the WiSARD weightless neural network model, applied to credit analysis, a non-trivial real-world problem. Experimental evidence show that ClusWiSARD is very competitive with Support Vector Machine (SVM) w.r.t. accuracy, with the difference of being capable of online learning. Nonetheless, it outperforms SVM in both training time, being two orders of magnitude faster, and test time, being slightly faster.

1 Introduction

Data with concept drift increases the complexity of classifiers, since information learnt is likely to degrade classification performance over time, as both feature patterns and target functions can change. This is the case of credit analysis, a recurring problem in the banking business which can be summarized as deciding which requests for credit should be granted. The usual process involves the collection of data, which is used to determine the "quality" of the request, in other words, the risk of a borrower failing to pay his or her debts. The analysis of these data, however, is a complex problem.

One of the aspects which makes this problem considerably harder is the change of patterns over time. The movement of populations, changes in economy, natural catastrophes [1], general news [2], among other factors which may directly affect the relations pertinent to credit. Another aspect to be considered is the bias of the available data: only data about granted requests are usually stored. This means there is not enough information about the bad payers.

An automated learning and classification mechanism could offer a more precise solution, being able to analyse vast amounts of data on credit applications and consider subtle relations between the actual financial data and the borrower profile. Those methods would need to be efficient and robust in order to account for changes in the circumstances and sample biasing. Two classifying mechanisms which exhibit these characteristics are the WiSARD [3] artificial neural network model and the Support Vector Machine (SVM) [4], which we introduce

*This work was partially supported by GE, Inovax, and CAPES, CNPq and FAPERJ Brazilian research agencies.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

in, respectively, sections 2 and 3. This work proposes a new implementation of the WiSARD weightless neural network, as well the necessary preprocessing for the analysed data. For comparison purposes the same data is classified by a Support Vector Machine.

2 Adopted Weightless Model

A Weightless Artificial Neural Network (WANN) is a pattern recognition system whose main difference from other learning methodologies lies on the direct use of information storage instead of error minimization, with Random Access Memories (RAMs) as storage mechanism [3]. The usual operation of a WANN uses the input to build a set of addresses which are used to access RAM nodes contents.

This work adopts WiSARD (**Wi**lkie **S**tonham and **A**leksander **R**ecognition **D**evice) [3], a pioneering WANN architecture that is composed by distinct sets of RAM nodes called discriminators. Each discriminator is assigned to one of the classes of patterns to be recognized, i.e., the number of discriminators in the WiSARD network is the same as the number of classes. A discriminator consists of a single layer of RAM nodes, which are all started with the default value zero (0) in every addressable position. The network has also been extended with a tie breaking capability, called bleaching [5], to deal with inconclusive pattern classifications.

WiSARD is, originally, a Boolean neural network and thus any input given to the architecture must be converted into a binary string. The preferred binary encodings for numeric features are the ones with a *Hamming distance* related to the numeric distance, so that the input can be compared for similarity. Encodings which do not have this characteristic, e.g. IEEE 754, should be avoided. After the conversion is made, the input is shuffled according to a fixed pseudorandom mask (defined at the creation of the network) and split to generate input addresses of all RAM nodes. During the training phase, some memory locations at the RAM nodes in the discriminator corresponding to the trained class are accessed according to each input pattern. Each access increments by one the value stored in the addressed content. During the classification phase every discriminator retrieves the information addressed by the input pattern. Each RAM node accessed this way outputs one (1) if its addressed memory position holds a value higher than the bleaching threshold, and zero (0) otherwise. A discriminator response is the sum of the outputs of each of its RAM nodes, as seen in Figure 1. The discriminator with the highest response is chosen for the classification. If two or more discriminators share the highest response then the bleaching threshold must be incremented by one and a new classification iteration is performed. Training and classification can be interleaved during runtime. By doing so, WiSARD can be employed in continuous (online) learning tasks.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

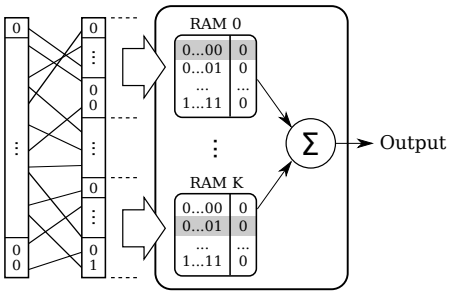


Fig. 1: Example of access in a discriminator.

2.1 ClusWiSARD

The combination of very different input patterns of the same category in a WiSARD discriminator can enable the recognition of test patterns very dissimilar from those learned previously by this unit. ClusWiSARD avoids this by creating input pattern clusters, according to the acceptance threshold of each of them. This resembles ART [6], but the use of discriminators as clusters representatives is the main difference from it.

Therefore, the main novelty of ClusWiSARD lies in its knowledge storage, which uses a group of discriminators per class. This allows for the distribution of training data into discriminators that better represent natural clusters, i.e., by capturing subpatterns as "subclasses". For each training input pattern, if any of the discriminators of this pattern class gives a recognition response higher than its acceptance threshold, the pattern is learned. If no discriminator accepts the input, a new discriminator is created to learn it. The acceptance threshold is proportional to the number of elements in the cluster.

Classification with ClusWiSARD is similar to the original WiSARD: the input is tested with all discriminators; if there is a tie a bleaching process occurs. The class of the discriminator with the highest response is chosen for the input.

3 SVM

Support Vector Machines (SVM) [4] is a widely used machine learning technique that allows for the determination of a maximum-margin hyperplane that separates data of two distinct classes. In this paper two classes are considered: paid-back and non paid-back loans. Furthermore, by projecting data in a higher dimensional space, one eventually reaches linear separability. This is done by using kernel functions, e.g., polynomial, radial basis function or sigmoidal. A linear kernel function would restrict the procedure to the margin maximization step, without projecting data into a larger dimensional space. Besides kernel selection, it is also possible to use a light version of the SVM mechanism. This light version does not take into account all data points to determine the separation margin, and usually achieves a better generalization.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

4 Data-Preprocessing

As in many applications, data preprocessing may improve the classifier performance. In this paper, three aspects of data analysis concerning preprocessing are covered: noisy data correction, attribute influence evaluation and optimal encoding. Next, a description of the data and the treatment proposed for each preprocessing aspect under consideration are presented.

The dataset used in this work comes from the BRICS-CCI & CBIC Computer Intelligence Algorithm Competition [7]. It contains the data of credit applications, labelled with the status of the approval. Each entry corresponds to a credit request from a client of a private credit card retail chain, labeled as "good" or "bad" depending on the payment behavior. A client is considered "bad" if he or she defaults the debt for more than two consecutive months, and considered "good" otherwise. The attributes for each entry include age, gender, net income, among others. There are 40 attributes in total, which were grouped in four categories: numeric range, numeric cyclic, categorical and Boolean.

The binary enconding used depended on the attribute type in order to be compatible with Hamming distances. The used encoding methods follow:

- Boolean encoding: Maps true values to "1" and false values to "0"; Used on boolean attributes, e.g., registered_id;
- Categorical encoding: Each value is mapped in a way that its Hamming distance is equal to all other values of the attribute; Used on categoric attributes, e.g., city;
- Range encoding: Maps the values while keeping the ordering of the original distance function; Used on numerical attributes, e.g., age;
- Cyclic encoding: The minimum and maximum values have the minimum Hamming distance between them; The maximum distance occurs on opposite sides of the cycle; Used on numeric cyclic attributes, e.g., payment_day;

It is also reasonable to expect that the attributes contribute differently to the classification task and that the dataset contains a large number of attributes with small or no contribution. To address this and reduce the number of attributes, a study was conducted on the degree of influence each attribute has over the payment behavior. The information value [8] was calculated for each of the 463 attributes derived from the preprocessing of the original 40, which were sorted. The hundred attributes with the highest information values were then chosen to be used with the SVM, the ClusWiSARD used the complete dataset. Figure 2 shows the information value of the attributes in logarithmic scale.

5 Classifiers Evaluation

The classifiers were compared through a binary classification task of credit requests. This kind of data is influenced by macroeconomical factors and events, which could be seasonal or not. This leads to concept drift: what characterizes a credit request as good or bad is related to when it was subjected. Taking this

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

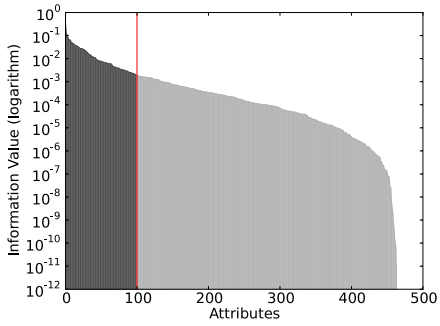


Fig. 2: Information value of the attributes.

into account, the classifiers' performance was evaluated on the observations of each month over a two-year span, after training using data of the three months before it. Therefore, the first three months of 2009 were not used as test targets.

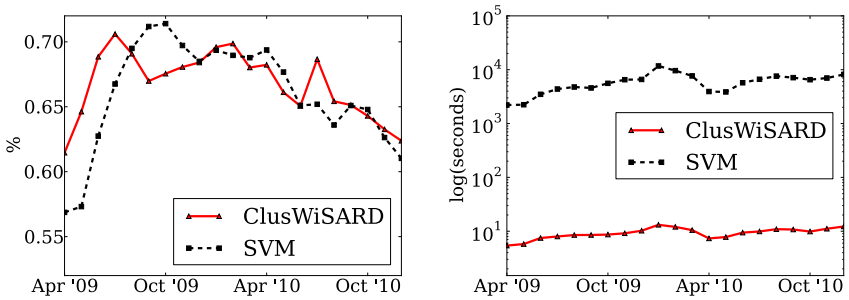
The method used to train the SVM network works as follows. In order to determine the best parameters for this specific problem, a second dataset composed of 12000 observations was created. This dataset was randomly generated from the original one with five hundred observations for each month. Once this dataset was ready, multiple instances of SVM networks were trained varying its parameters. These consisted of the kernel function used (Polynomial, RBF and Sigmoid) along with their configuration parameters: γ, $coef0$ and $degree$ for Polynomial; γ for RBF; and γ and $coef0$ for the Sigmoid kernel. In the end, the polynomial kernel function obtained the best results. The configuration had a $degree$ of 3, $coef0$ equal to zero and the value for γ was approximately 0.01509, according to the equation:

$$k(u, v) = (\gamma * u^T * v + coef0)^{degree} \tag{1}$$

The parameters u and v from the kernel function are two samples from the training set passed to the kernel. An extended description about the meaning of the kernel function and how they are used can be found in [4]. The classifiers were optimized according to the average per-class accuracy. This was preferred over the general accuracy because of the great difference in the dataset share between the classes, which, however, does not represent a greater significance of one class compared to the other. Predicting all observations as being of the more popular class could result in a good general accuracy although no discretization of the classes were being made. On the other hand, a too high average per-class accuracy could also be a sign of over-fitting. SVM performs around 10% better than ClusWiSARD at this measure despite losing in general accuracy, showing a bit more discretization power, at expense of time.

Figure 3a shows the general accuracy results of the tested options. Figure 3b logarithmically compares the time spent by both models to train with the data from the three previous months.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.



(a) General Accuracy (b) Training time

6 Conclusion

The problem of classification applied to credit analysis is an interesting application for machine learning. In the case of this work, the data preprocessing was an important step in the classification of the presented data, resulting in improvement for both mechanisms, though there was an added binarization step for ClusWiSARD.

Both ClusWiSARD and SVM presented close and accurate results in this batch learning setting but ClusWiSARD also demonstrated great advantage in training time. Moreover, ClusWiSARD operates in the same way for an online learning setting, therefore being more flexible. Further work should develop a proper benchmark for comparison.

In conclusion, ClusWiSARD appears to be a promising model for large and complex amounts of data with efficient training and classification time.

References

[1] Tarja Joro and Paul Na. Derivatives and credit risk: Credit risk modeling for catastrophic events. WSC '02. Winter Simulation Conference, 2002.

[2] Hsin-Min Lu, Feng-Tse Tsai, Hsinchun Chen, Mao-Wei Hung, and Shu-Hsing Li. Credit rating change modeling using news and financial ratios. *ACM Trans. Manage. Inf. Syst.*, 3(3):14:1–14:30, October 2012.

[3] Igor Aleksander, W. V. Thomas, and P. A. Bowden. WISARD: A radical step foward in image recognition. *Sensor Review*, 4:120–124, July 1984.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] Bruno P. A. Grieco, Priscila M. V. Lima, Massimo De Gregorio, and Felipe M. G. França. Producing pattern examples from "mental" images. *Neurocomputing*, March 2010.

[6] Gail Carpenter and Stephen Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, pages 54–115, 1987.

[7] NeuroTech S.A. (Brazil). CI algorithms competition dataset, 2013. http://brics-cci.org/ci-algorithms-competition-ciac/.

[8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.