Feature Set Selection and Weighting for Legal Amount Recognition on Brazilian Bank Checks

Adélia C. de A. Barros[1] and *George D. C. Cavalcanti*[1,2]

[1]Center of Informatics, Federal University of Pernambuco
[2]AiLeader Technologies
Recife, PE, Brazil
{acab,gdcc}@cin.ufpe.br

Abstract

This work presents a study about feature selection and weighting for improving the recognition of handwritten words coming from Brazilian bank check lexicon. For this purpose, two global optimization methods are used: Tabu Search*(TS) and* Simulated Annealing*(SA). These methods were combined with k-NN composing two hybrid approaches for features selection and weighting: SA/k-NN and TS/k-NN. The results show that feature sets optimized by the studied models are very efficient when compared with k-NN. Both, accuracy classification and number of features in the resultant set are considered in the conclusions. Furthermore, some new structural features extracted from image upper and lower profiles are proposed.*

Keywords: legal amount recognition, feature selection and weighting, tabu search, simulated annealing

1. Introduction

Automatic check processing still is a challenging problem, specially when the legal amount (LA) recognition is the subject. This kind of system has a demand from the banking industry, because nowadays great part of the checks are manually processed [1].

The system described in this paper has been constructed to deal with the problem of legal amount post-classification. Great part of the previously proposed systems to recognize LA is based on a list of probable answers [2, 3, 4, 5]. Considering the Brazilian checks lexicon, we notice some classes share a common sub-string (suffix or prefix). On the one hand this is a meaningful discriminatory information once we can more easily classify a word as being member of some suffix or prefix group. On the other hand, the objective of a bank check recognition system is not to categorize into groups of similar words, but to find an unique answer for each LA segment.

In this work, we assume a system divided in two recognition stages: the first one groups the words based on their suffixes and prefixes. The second, called here post-classification, chooses one word from the group as the final class answer. Important works have handled with prefixes and suffixes in the recognition of handwritten words [6, 7]. This paper focuses on the second stage: techniques of feature selection and weighting are used here for improving the post-classification results.

The feature set choice is one of the earliest and most important decisions in classification tasks. At this point, defined attributes will be considered in a certain class of objects in order to make easier their recognition. However, this choice is not simple: the more complex is the classification problem, the harder it is to find a satisfactory group of discriminant features.

Furthermore, the feature size set aims at being more compact once it directly affects the classifier performance and design. Larger feature sets imply in more complex data structures and computations. Despite the supracited advantages of features set careful choice, this process is made by either through an empirical or a subjective way at times.

As one may conclude, the desired solution is the smallest feature set sufficient to preserve the quality of discriminant information between classes. In this case, we can deal with features selection as an optimization problem whose objective function could be, for instance, to improve the accuracy rate as well as to reduce the feature set.

While feature selection leads to the reduction of the number of features, feature weighting is a method that preserves the original size of the set, with the singularity of associating weight values to each feature according to its discriminatory ability.

Two well-known optimization methods could be used to find the optimal set of features: Tabu Search(TS) [8, 9] and Simulated Annealing(SA) [10]. Both methods are

[0]This work was supported in part by the Brazilian National Research Council CNPq (Proc. 478534/2006-0).

particularly efficient since they search in the space for the global minima solution avoiding local minima. TS was previously used for simultaneous feature selection and feature weighting in [11]. In the referred work, a new hybrid TS/k-Nearest Neighbor algorithm was proposed in which TS searches for solutions in the space and k-NN [12] evaluates these solutions. Despite of being an expensive solution, its results overcomes those obtained with well-known feature selection methods. Feature selection and weighting are useful for improving classification accuracy in union with optimization methods [13, 14].

This work presents a study of simultaneous feature selection and feature weighting through global optimization applied on post-classification of LA words. For this purpose, the hybrid algorithms TS/k-NN and SA/k-NN are used here to evaluate the classification accuracy rate. In a previous work [15], both methods have improved results of popular benchmarks.

Organization of this paper is as follows. Section 2 outlines the feature extraction procedures for the recognition module. Section 3 introduces the global optimization methods studied here: TS/k-NN and SA/k-NN. In Sections 4 and 5, we present the experimental methods and results of these techniques, respectively. Finally, important remarks are presented in the conclusive Section 6.

2. Feature Extraction

Legal amount recognition has a small vocabulary for Brazilian checks, such as 40 words. However the major problem is given by the great variability in writing. With these constraints in mind, were selected a set of 89 features: 50 Horizontal Transitions; 36 extracted from the upper and lower profiles; and, width and height of the image, plus the ratio between them. This section details the proceedings to obtain the feature set used in this work.

2.1. Horizontal Transitions

The idea behind Horizontal Transitions procedure is to scan the image horizontally, from left to right, counting the number of 0-1 and 1-0 transitions. In this approach 0 means background while 1 means foreground pixels. Therefore, the first step is to normalize the image to a predefined height (50 pixels were used). Based on that assumption, a total of 50 features were calculated as the mean of the 0-1 and 1-0 transitions.

2.2. Upper and Lower Profiles Based Measures

After find a bounding-box (BB) of the sample image, Upper and Lower Profiles can be calculated as an one-dimensional vector. The Upper Profile estimates the distance, in pixels, from the top of the BB to the first object below it. On the other hand, the Lower Profile is given by the distance between the bottom of the BB and the first pixel that belongs to the object above it.

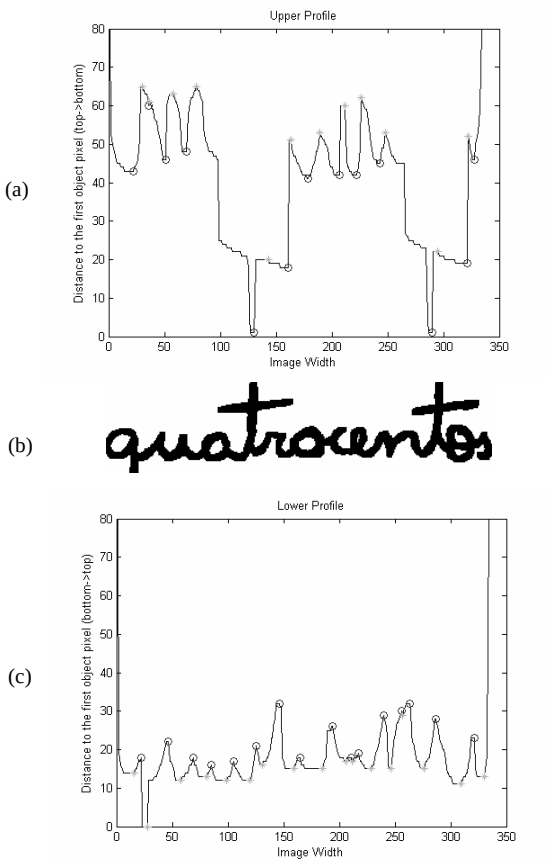

Figure 1. (a) Upper profile. (b) An image word sample . (c) Lower profile. The $\star$ means local maxima points and the $\circ$ local minima.

In Figure 1(b) a sample image is given (it is written in Portuguese (Brazil) and means "four hundred"). Its upper and lower profiles can be seen in Figures 1(a) and 1(c), respectively. From each profile, there are some especial points that are marked with "$\star$" (star) and "$\circ$" (circle), which mean the maximum (peaks) and the minimum (valleys) local of the profile, respectively. Therefore, we have two profiles to analyze: upper and lower. Each one having two set of points: maximum and minimum.

Let $x_m^l = [x_1, x_2, \ldots, x_n]$ be a profile vector, where n is the size of the vector, l can be replaced by *upper* or *lower* and m by *maximum* or *minimum*. From this vector a total of 5 (five) features are extracted, they are (adopt h as image height):

- Number of peaks (valleys) ($count(x_m^l)$);
- Maximum value normalized ($max(x_m^l)/h$);
- Minimum value normalized ($min(x_m^l)/h$);
- Average value normalized ($ave(x_m^l)/h$);

- Standard deviation ($std(x_m^l)$).

Besides these features, more 4 (four) ones per profile was obtained based on the difference of the vector x_m^l. The idea is to calculate the difference between adjacent elements of the vector. This will give us a good estimate of the proximity of the peaks (or valleys). Let dx be the difference vector given by $dx_m^l = [x(2) - x(1), x(3) - x(2), \ldots, x(n) - x(n-1)]$. The dx_m^l returns a vector one element shorter than x_m^l. The features are:

- Maximum value normalized ($max(dx_m^l)/h$);
- Minimum value normalized ($min(dx_m^l)/h$);
- Average value normalized ($ave(dx_m^l)/h$);
- Standard deviation ($std(dx_m^l)$).

3 Global Optimization Algorithms

There are in the literature methods based on the gradient, they use this local information to move in the direction of a function minima point. Backpropagation [18] is a classical example of it. It is highly efficient to determine the appropriate direction and magnitude for moving iteratively from a coordinate in the space to a local minima.

On the other hand, global optimization algorithms search for the global minima point based on the information of the function surface as a whole [17]. Both methods *Tabu Search* and *Simulated Annealing* are considered global algorithms. They have as goal to find the function global minima point discarding the local ones when necessary.

3.1 TS/k-NN

As mentioned above, *Tabu Search* is a global search algorithm. A set of new solutions is created from the current solution and the best one (that whose cost function is the less) is always accepted as the current. The fact of always admit the new solution avoids the fall into local minima.

In order to prevent cycles in the search trajectory, the lately visited solutions are stored in a list called *tabu*. These solutions are prohibited with the intention of avoiding the algorithm to look up them again.

However, the cost of storing all the visited solutions is too high. So, only the T (size of tabu list) last solutions keep on list.

Tabu Search keep in memory the best of all visited solutions apart of being the current. So, even if it "pass" the best solution over the execution, it still will be available.

The outline is given in Algorithm 1. The 8^{st} step refers to find the best solution in the neighborhood. In this step the cost of each solution if evaluated by a k-NN classifier.

Algorithm 1 Tabu Search

1: s_0: initial solution
2: s_{best}: best solution
3: I: number of iterations
4: V: set of neighbors solutions
5: Insert s_0 in tabu list
6: **for** $i = 0$ to $i = I - 1$ **do**
7: Generate V neighbors solutions
8: Find the best $s' \in V$
9: **if** s' is not in tabu list **then**
10: $s_{i+1} \leftarrow s'$
11: Update tabu list
12: Update s_{best}
13: **end if**
14: $i = i + 1$
15: **end for**
16: Return s_{best}

3.1.1 Encoding solution

The encoding solution proposed by [11] consists of 3 parts. The first consists of weight values associated to each feature. These weights will be modified through *feature weighting*. The second part, or the binary part, consists of 0 or 1 values. And, finally, the third part consists of the *k* value extracted from *k-NN*.

3.1.2 Cost function

The cost function used in this work is the total number of correctly classified patterns as shown in Eq. 1. So, we have as objective function to maximize this number.

$$Cost = \sum_{i=1}^{n} C_i \tag{1}$$

where n is the number of classes and C_i is the number of correctly classified patterns in each class.

3.1.3 Neighborhood

There is a step in the *Tabu Search* algorithm that consists in the neighborhood creation. In this approach, a total of *M* x *N* + *P* new solutions are created at each iteration. In which *M* x *N* new solutions, or neighbors, are created by assigning *M* random weights to *N* different features. *P* new solutions are generated from turning on/off the bit of the encoding second part. In this approach, to turn a bit off means to delete a feature. Each neighbor solution is evaluated and the best one is chose to be the next current solution.

3.1.4 Termination criteria

The algorithm stops if the fixed number of iterations is reached or if after some iterations the objective function does not change.

3.1.5 Tabu rule

If a solution feature has been recently modified and it is in the tabu list, it is then prohibited. It means that, to accept an current solution, that feature could not have been modified before.

3.2 SA/k-NN

The method *Simulated Annealing* consists in, at each iteration, generate one new solution from the current. When this solution is created, its cost is evaluated to decide if it can be accepted as current. If the cost is less than the current, it is accepted. Otherwise, it can be accepted with a certain probability, known as the *Metropolis Criteria* [19]. According to this criteria, a random number δ between 0 and 1 is generated. If $\delta \leq e^{(-\Delta C/t)}$, then the solution is accepted. Where ΔC is the cost function variation and t is a parameter called temperature. Cooling schemas are responsible to define an initial temperature value, as well as, a rule to iteratively decreases this value. In this work the cooling schema adopted was the *Geometric* one [20].

The encoding solution, neighborhood generation, termination criteria and tabu rules of acceptance are the same described in the section 3.1. The algorithm outline is given in Algorithm 2.

Algorithm 2 Simulated Annealing

```
1: s_0: initial solution
2: I: number of iterations
3: for i = 0 to i = I − 1 do
4:    Generate solution s'
5:    if Cost(s') ≤ Cost(s_i)) then
6:       s_{i+1} ← s'
7:    else
8:       if random ≥ e^{(−[Cost(s')−Cost(s_i)]/t_{i+1})} then
9:          s_{i+1} ← s'
10:      end if
11:   end if
12:   i = i + 1
13:   Update temperature
14: end for
15: Return s_i
```

4 Database and Experiments

Three methods were compared in the following experiments, they are:

- **k-NN:** well-known classifier which uses Euclidean Distance as similarity measure.
- **SA/k-NN:** hybrid method constituted by k-NN and *Simulated Annealing* described in Section 3.2.
- **TS/k-NN:** the technique *Tabu Search* described in Section 3.1 combined with k-NN.

The experimental results were performed over the Legal Amount vocabulary from Brazilian bank checks. Figure 2 shows samples of the 40 words that constitutes the LA grammar. As mentioned before, the variability in writing is a notorious problem in this kind of application. There were used a total of 300 samples of each word to perform the experiments.

Data were randomly divided into training set and test set. The training set consists of 70% of the total number of prototypes and the test set consists in 30% of the data. Houldout is used in the training phase. To compute the results, 5 runs of each algorithm were simulated and their classification accuracy were used to obtain the final average rate results.

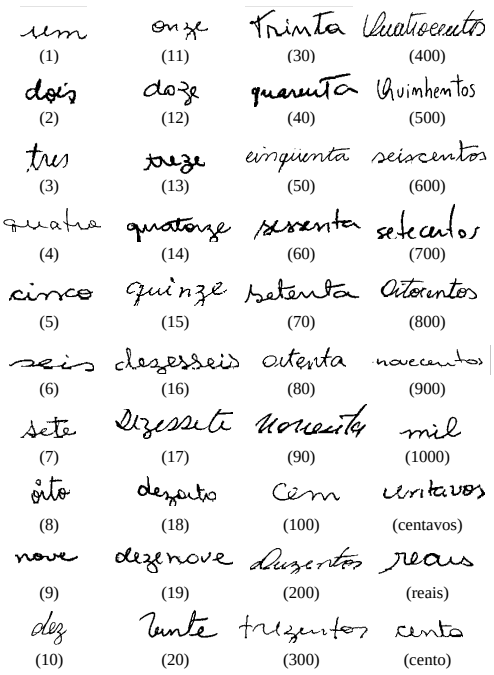

Figure 2. Samples of legal amount database (the numbers in the parentheses represents the value of the written word).

There were run 5 simulations of the k-NN method varying the number of k, for each database. The k values experimented are in the interval from 1 to 5. The one which has presented best results was adopted. The final rate is the average of results collected in each iteration. In this work, k-NN iterations number (I) was set to 100.

The method TS/k-NN has its parameters described here as well. The number of iterations is 100 for all datasets. In addition to it, were performed 5 trial runs, in other words, each run has the average result of 100 it-

erations and the final result is the average of the 5 trials.

One important parameter of *Tabu Search* is the length T of tabu list. Some criterion are suggested in the literature in order to help in the choice of T. The tabu list size was calculated, in this work, by:

$$T = ceil(\sqrt{F}) \tag{2}$$

where T and F are, respectively, the size of tabu list and the number of features.

Other parameters introduced by the model proposed in [11] are *M*, *N* and *P*. Just to review, *M* x *N* + *P* neighbors are generated at each TS/k-NN iteration. Where *M* are the number of patterns, *N* the number of feature weights to be modified and, finally, *P* is the number of solutions generated by turning on/off some features. These parameters values were set according to [11]. In that work, the value of *P* was obtained through Eq. (2), *M* was set to 10 and *N* set to 2.

The same values of *M*, *N* and *P* used in TS/k-NN were used in the SA/k-NN model. In the *Simulated Annealing* rule, we have seen that even if a new solution has inferior performance than the current solution, it could be accept to avoid local minima. Follows the acceptance criterion: one random value is tested to be less than a prior defined probability, if the test succeeds, the new solution replaces the current one. Otherwise, it is discarded. In this work, the prior defined probability was determined by:

$$p = \frac{exp(-(E_n - E_a))}{t} \tag{3}$$

where p is the probability, t is the temperature, E_a is the classification error of the current solution and E_n the error of the new solution.

The choice of a cooling schema is another configuration required by the SA method. A cooling schema has to define an initial temperature as well as the rule for updating its value. One common cooling schema is the *Geometric* one [20], where the new temperature is computed by the product of the current temperature with a reduction factor. In this study, the initial temperature was set to 1 and the reduction factor set to 0.9. The temperature is updated following this schema at every 10-multiple iteration.

5 Results

The idea is to expose the accuracy rate of classes that are commonly misplaced. A source of error cames from words that share the same prefix or the same suffix. Another source comes from similar words derived from the same kernel, not necessarily a suffix or prefix. For example we have the words: “dois”, “doze” and “duzentos”, which are represented by the classes “2-12-200” in Table 1. Observing these classes (see samples in Figure 3), we notice they do not share a suffix, but they have many structural characteristics in common. It appears as a confusion source in the recognition system.

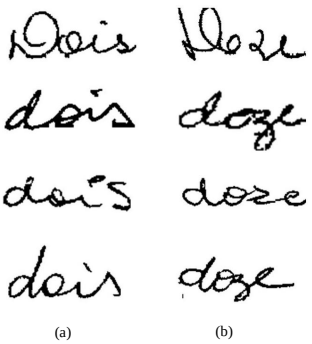

Figure 3. (a) Examples of “dois” (2). (b) Examples of “doze” (12).

Table 1 shows the accuracy rate in % for the algorithms k-NN, SA/k-NN and TS/k-NN when using *Euclidean Distance*. Notice that the results of the SA/k-NN outperforms the simple k-NN results to all comparison groups presented. Furthermore, it is important to observe that for some groups of classes the SA/k-NN performance is near or better than TS/k-NN, for example the group “4-14-40-400”. This observation let us to consider SA/k-NN as an interesting cheaper alternative to TS/k-NN.

Another point to observe is the resultant number of features. The k-NN does not reduces the features number, so we have to consider the total number (89) in this case.

According to the table, the usage of hybrid SA/k-NN and TS/k-NN always reduced significantly the number of features. It means, SA/k-NN have reduced the feature set in the interval between 1.12% and 20.22% while TS/k-NN had the reduction interval between 11.23% and 25.84%.

6 Conclusion

In this paper, two hybrid optimization algorithms are used to improve post-classification results of handwritten words from the Brazilian bank checks lexicon: Simulated Annealing/k-Nearest Neighbor and Tabu Search/k-Nearest Neighbor. A post-classification procedure is required because many of the legal amount recognizers give as answer a list of probable classes. An efficient way to answer the correct class between the n-top ones makes the system faster and more reliable. SA/k-NN and TS/k-NN methods are proposed to simultaneous feature selection and weighting. These methods outperformed the results get with simple k-NN for all class groups analyzed. It includes both: the smaller number of features in the resultant subset and the classification accuracy.

Another contribution is the detailed feature extraction

Table 1. Accuracy classification rate $\bar{\mathbf{x}}$ (in %), number of neighbors **k** and resultant feature set size **f**

	k-NN		SA/k-NN			TS/k-NN		
Similar Classes	$\bar{\mathbf{x}}$	**k**	$\bar{\mathbf{x}}$	**k**	**f**	$\bar{\mathbf{x}}$	**k**	**f**
2-12-200	92.23 ± 0.66	5	95.78 ± 0.68	4	83	96.30 ± 0.59	8	66
3-13-30-300	85.40 ± 0.65	3	92.28 ± 0.62	9	87	93.22 ± 0.68	3	72
4-14-40-400	72.55 ± 0.75	5	83.44 ± 0.36	9	83	83.05 ± 0.73	9	75
5-50	99.59 ± 0.24	3	99.89 ± 0.12	5	71	100.00 ± 0.00	5	66
15-500	96.56 ± 0.61	1	99.67 ± 0.25	1	78	100.00 ± 0.00	1	71
6-16-60-600	73.56 ± 0.75	5	79.56 ± 0.66	5	88	79.78 ± 0.53	3	70
7-17-70-700	75.67 ± 1.09	4	82.22 ± 0.88	7	86	84.61 ± 0.79	7	79
8-18-80-800	84.08 ± 0.96	1	89.61 ± 0.47	1	86	89.89 ± 0.71	9	66
9-19-90-900	79.14 ± 0.80	1	83.56 ± 0.20	3	87	84.67 ± 0.41	9	72
20-30-40-50-60-70-80-90	46.32 ± 1.09	1	54.75 ± 1.20	7	82	55.67 ± 0.68	7	71
200-300-400-500-600-700-800-900	45.08 ± 1.64	1	53.67 ± 1.05	1	86	55.25 ± 0.60	5	67

described in Section 2. After many experiments, over a Brazilian LA database, the feature set reaches very satisfactory results. We consider important as a future work to compare this extraction with other popular methods.

Furthermore, we worked with very common problems in the literature (feature selection and feature weighting) in order to contribute still more with this research area.

Finally, the good results presented by the hybrid global optimization algorithms prove still more their efficiency in feature selection problems. In previous works [15, 11], they have already succeed in this task. This fact instigates deeper studies concerning this subject.

References

[1] C. Zanchettin, G. D. C. Cavalcanti, R. C. Doria, E. F. A. Silva, J. C. B. Rabelo and B. L. D. Bezerra, *A Neural Architecture to Identify Courtesy Amount Delimiters*. International Joint Conference on Neural Networks, pp. 5849–5856, 2006.

[2] C. O. Freitas, A. E. Yacoubi, F. Bortolozzi and R. Sabourin. *Brazilian bank check handwritten legal amount recognition.*. Brazilian Symposium on Computer Graphics and Image Processing, pp. 97-104, 2000.

[3] N. Gorski, V. Anisimov, E. Augustin, O. Baret and S. Maximovu. *Industrial bank check processing: the A2iA CheckReader.*. International Journal of Document Analysis and Recognition, (3):196-206, 2001.

[4] K. K. Kim, J.-H. Kim, Y. K. Chung and C. Y. Suen. *Legal amount recognition based on the segmentation hypotheses for bank check processing.*. International Conference on Document Analysis and Recognition, pp. 964–967, 2001.

[5] H. Tang, E. Augustin, C. Y. Suen, O. Baret and M. Cheriet. *Recognition of unconstrained legal amounts handwritten on chinese bank checks.* International Conference on Pattern Recognition, vol. 2, pp. 610-613, 2004.

[6] M. N. Kapp, C. Freitas and R. Sabourin; *Methodology for the Design of NN-based Month-Word Recognizers Written on Brazilian Bank Checks*. Image and Vision Computing, vol. 25, pp. 40–49, 2007.

[7] C. Freitas, F. Bortolozzi and R. Sabourin, *Handwritten isolated word recognition: an approach based on Mutual Information for feature set validation*. International Conference on Document Analysis and Recognition, pp. 665–669, 2001.

[8] F. Glover, *Future paths for integer programming and links to artificial intelligence*. Computers and Operation Research, vol. 13, pp. 533–549, 1986.

[9] P. Hansen, *The steepest ascent mildest descent heuristic for combinatorial programming.* Conference on Numerical Methods in Combinatorial Optimisation, 1986.

[10] S. Kirkpatrick, C. D. Gelatt Jr and M. P. Vecchi, *Optimization by simulated annealing.* Science, vol. 220, pp. 671–680, 1983.

[11] M. A. Tahir, A. Bouridane and F. Korugollu, *Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier.* Pattern Recognition Letters, vol. 28, pp. 438–446, 2007.

[12] T. M. Cover and P. E. Hart, *Nearest Neighbor Pattern Classification.* IEEE Transactions on Information Theory, vol. IT-13, no. 1, pp. 21–27, 1967.

[13] M. L. Raymer et al., *Dimensionality reduction using Genetic algorithms.* IEEE Transactions on Evolutionary Computation, vol. 4(2), pp. 164–171, 2000.

[14] H. Zhang and G. Sun, *Features Selection using Tabu Search method,* Pattern Recognition, vol. 35, pp. 701–711, 2002.

[15] A. C. A. Barros and G. D. C. Cavalcanti, *Combining Global Optimization Algorithms with a Simple Adaptive Distance for Feature Selection and Weighting*. IEEE International Joint Conference on Neural Networks, Hong Kong, 2008.

[16] J. Wang, P. Neskovic and L. N. Cooper, *Improving nearest neighbor rule with a simple adaptive distance measure.* Pattern Recognition Letters, vol. 28, pp. 207–213, 2007.

[17] X. Yao, *Evolving artificial neural netwoks*. Proceedings of the IEEE, vol. 87, pp. 1423–1447, 1999.

[18] A. K. Jain, J. Mao and K. M. Mohiuddin, *Artificial Neural Networks: A Tutorial.* IEEE Computer, pp. 31–44, 1996.

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, *Equation of state calculations by fast computing machines*. Journal of Chemical Physics, vol. 21, no. 6, pp. 1087–1092, 1953.

[20] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques*, Springer, 2000.