FeaSANNT – an Embedded Evolutionary Feature Selection Approach for Neural Network Classifiers

Cardiff University
Manufacturing
Engineering
Centre
Cardiff (Wales),
UK

Universidade
Nova de Lisboa
Departamento
de Informática
Faculdade de
Ciências
e Tecnologia
Caparica,
Portugal

Marco Castellani
Nuno C. Marques

This paper presents FeaSANNT, an evolutionary feature selection and weight training procedure for neural network classifiers. FeaSANNT exploits the global nature of the evolutionary search to avoid sub-optimal peaks of performance. The novelty of the method lies in the implementation of the embedded approach in an evolutionary feature selection paradigm. Such choice minimises the computational overheads and eases the algorithm design. FeaSANNT is used to train a multi-layer perceptron classifier on six real-world benchmark problems. FeaSANNT achieves accurate and robust learning results with significant reduction of the feature set. Experimental comparison shows FeaSANNT outperforms three standard state-of-the-art feature selection methods.

Keywords: neural networks, evolutionary algorithms, feature selection, data mining

1 Introduction

Most of nowadays software systems and knowledge repositories have grown to a considerable level of complexity. This complexity is often of hindrance to the system maintenance, transparency, and usability. One of the most crucial problems in the management of information is the representation of the data through a set of attributes. This set should contain all the necessary information to characterise the data, while at the same time should be as coincise as possible. The exhaustiveness and the coinciseness of the data descriptors greatly affect the effectiveness and the complexity of the system that manipulates the information. A poor choice of data attributes (e.g. conflicting, incomplete data features) makes it difficult to map data relationships, and to retrieve the desired information. A set of data attributes containing redundant information will unnecessarily complicate the data representation, making it also difficult for the user to pinpoint the significant information.

This study concerns the selection of input variables for artificial neural network (ANN) classifiers [Pham & Liu 1995]. Feature selection [Blum & Langley 1997; Fogel 2000] entails the removal of conflicting, overlapping and redundant features to maximise the ANN accuracy and compactness, and minimise the cost of data acquisition.

Due to the often large set of attributes and their interactions, feature selection is difficult and time consuming. Feature selection is usually performed either via the filter approach or the wrapper approach [Blum & Langley 1997].

The filter approach selects the attributes based on desirable properties such as orthogonality and information content. The filter method is the least computational intensive. However, since the feature selection criterion ignores the inductive and representational biases of the learning system, filter algorithms are prone to unexpected failures [Fogel 2000]. The wrapper approach evaluates the solutions on the learning results of the classifier. This method involves a severe computational effort that may hinder the exploration of the search space. Greedy backward elimination and forward selection techniques are generally employed to search the solution space [Blum & Langley 1997; Cantu-Paz & Kamath 2005]. Unfortunately, such methods evaluate the features separately, not taking into account interactions amongst attributes. Since many ANN training algorithms are prone to sub-optimal convergence, wrapper procedures must also take into account inaccurate evaluations of the candidate solutions.

Evolutionary algorithms (EAs) [Portinale & Saitta 2002] are known to produce more reliable results when searching complex, multi modal and noisy surfaces. EAs [Portinale & Saitta 2002] have been widely used for feature selection [Yao 1999; Zhang et al. 2005] for ANNs, the wrapper approach being the common implementation. Despite encouraging results, the use of EAs has been limited by the lengthy training procedures of the classifiers. Lengthy evaluations of the solutions add to the slowness of convergence of EAs, making the application of this approach problematic.

Drastic reduction of computational effort is achieved by embedding the feature selection process into the training procedure of the classifier. This embedded approach [Blum & Langley 1997] avoids the computational overheads of repeating the whole training procedure for every evaluation of a solution. Embedded feature selection approaches for ANNs are to date less common and concern backward elimination or forward selection techniques [Blum & Langley 1997; Schetinin 2003; Cibas et al. 1994]. Such greedy methods, as mentioned before, are prone to sub-optimal convergence and ignore interactions amongst features.

This paper presents FeaSANNT, a new EA for feature selection and ANN training. FeaSANNT is characterised by a novel approach based on the concurrent evolution of the input vector and the weights. This embedded approach reduces the EA computational overheads, freeing resources for sampling of the solution space.

Section 2 introduces the algorithm. Three state-of-the-art feature selection approaches are compared to FeaSANNT on training a multi-layer perceptron (MLP) classifier [Pham & Liu 1995] on six benchmark problems. Section 3 describes the experimental design. Section 4 presents the learning results. Conclusions are drawn in Section 5.

2 FeaSANNT: Feature Selection and ANN Training Algorithm

The Feature Selection and ANN Training (FeaSANNT) algorithm is an embedded feature selection procedure for ANNs. Compared to the customary evolutionary wrapper approach, the embedded approach removes the need to restart the ANN training procedure for every evaluation of a solution. In an embedded algorithm, the number of ANN training cycles coincides with the convergence of the evolutionary process. This removes the problem of setting a separate stopping criterion for the weight training algorithm. This problem is of major concern in wrapper algorithms, since different feature vector configurations determine different ANN learning speeds. This section presents the implementation of FeaSANNT to the evolution of MLP classifiers.

2.1 General Overview

FeaSANNT architecture is sketched in figure 1. The algorithm comprises two components, namely a feature selection module and an ANN training module. At each evolution cycle, the two modules act jointly on the current population to produce a new population of candidate solutions. Within an evolution cycle, the feature selection module and the ANN training module act sequentially due to implementation issues (one genetic operation after the other). The two modules act on different sections of the genotype, this guarantees that their operations are independent from their sequential order. Since in EAs time steps are customarily associated to evolutionary cycles, the two modules can be considered to act concurrently.

Evolution is achieved through random operations of mutation and crossover. Crossover is operated only in the feature selection module. The choice against genetic recombination of the ANN structures is motivated by the lack of clear functional units in ANNs, and by the competing convention problem [Thierens et al. 1993].

2.2 Representation Scheme

The genotype of each individual is composed of two chromosomes.

The first chromosome is a binary mask of length equal to the size of the full feature set. This chromosome defines whether an attribute is fed to the ANN input layer and is manipulated by the operators of the feature selection module.

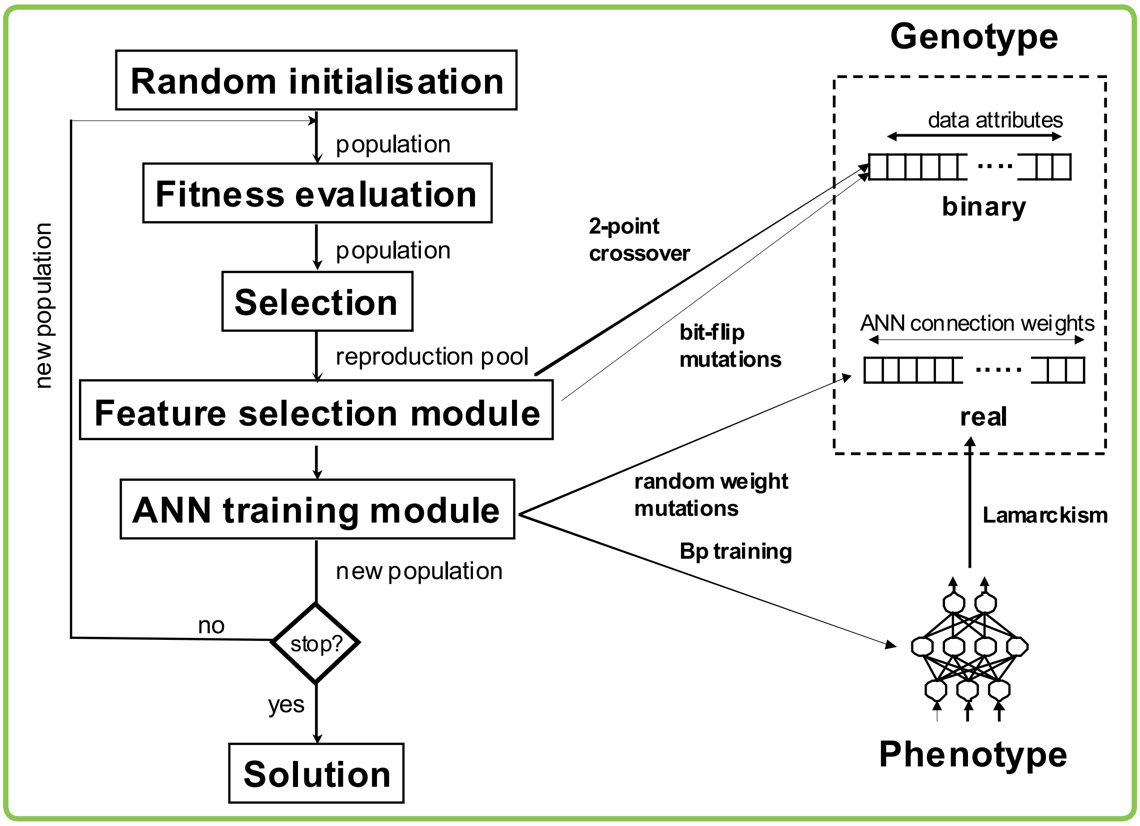

Figure 1: FeaSANNT Architecture

The second chromosome is a real-valued vector that encodes the ANN weights. This chromosome is manipulated by the operators of the weight optimisation module.

2.3 Fitness Evaluation Procedure

The fitness of an individual is given by the following measures:

$\text{fitness}_j = \{\text{accuracy}_j, \text{size}_j\}$

where accuracy_j is the classification accuracy of the j^{th} member of the population on the training set, and size_j is the size of the feature vector used by the j^{th} member of the population.

2.4 Selection Scheme

Fitness ranking [Portinale & Saitta 2002] is used to select individuals for reproduction. Solutions are ranked according to their accuracy (i.e. descending order of fitness measure accuracy_j). To encourage the creation of compact solutions, if two individuals have equal accuracy, the solution using the smallest feature set is ranked first (i.e. ascending order of fitness measure size_j).

2.5 Genetic Operators of Feature Selection Module

The feature selection module manipulates the binary input mask via the two customary genetic operators of bit-flip mutation [Portinale & Saitta 2002] and two-point crossover [Portinale & Saitta 2002].

2.6 Genetic Operators of Neural Network Weight Optimisation Module

Evolution is achieved via two operators, namely mutation and the Backpropagation (BP) [Pham & Liu 1995] rule.

Genetic mutations slightly modify the weights of each node of a solution. For each weight, the perturbation is randomly sampled with uniform probability from an interval of pre-defined width.

The BP rule [Pham & Liu 1995] is used as a deterministic mutation operator to speed up the learning process. This operator changes the weights of the decoded individuals to reduce the classification error. The changes are stored into the genotype (Lamarckian learning) [Aboitiz 1992]. If selected, a solution undergoes one cycle of BP learning over the whole training set. Because BP is computational expensive, the operator is used with a moderate rate of occurrence.

2.7 Termination Condition

The evolutionary procedure is repeated until a pre-defined number of iterations has elapsed. The fittest solution of the last generation is picked as the final solution.

3 Experimental Design

FeaSANNT is tested on six benchmark classification problems. For comparison purposes, four control algorithms are tested on the same benchmark problems. The four control algorithms include well-established and effective examples of state-of-the-art feature selection and ANN training procedures.

3.1 Data Sets

Six real-world numerical data sets are chosen from the UCI Machine Learning Repository [UCI Machine Learning Repository]. Their features are listed in table 1.

Each algorithm uses the training set for learning and the test set for the final estimation of the learning accuracy. To reduce the danger of over fitting, the order of presentation of the training samples is randomly reshuffled for every learning cycle.

3.2 Full Feature Set

In the first experiment, the ANN is trained using the full set of attributes. The results of the learning trials are used as a baseline to assess the efficacy of the feature selection algorithms.

For each classification problem, the MLP is trained using the BP rule with momentum term. The learning procedure is run for a fixed number of iterations. This algorithm will be called henceforth basicBP.

3.3 Feature Selection through Filter Approach

In the second experiment, the ANN is trained using a reduced set of attributes that is generated by a feature extraction algorithm based on the filter approach.

Table 1:
Data Sets

	Source Size	Features	Classes	Training	Set
Ionosphere	UCI ML Repository	351	33	2	200 – fixed
LandSat	UCI ML Repository	6435	36	6	4435 – fixed
Musk	UCI ML Repository	6598	166	2	80 % – random
Segmentation	UCI ML Repository	2310	19	7	80 % – random
Vehicle	UCI ML Repository	846	18	4	80 % – random
Vowel	UCI ML Repository	990	10	11	528 – fixed

Feature reduction is performed following principal components analysis (PCA) [Mardia et al. 1979] of the attribute set. Even though, strictly speaking, PCA is not a feature selection method but a feature extraction method [Cantu-Paz & Kamath 2005], PCA is a good term of comparison since it is one of the most effective and well understood filter-based feature reduction techniques.

PCA transforms the original vector space of (possibly) correlated variables into an equivalent space of uncorrelated variables (the principal components). Since most of the data variance can be accounted for by a small number of principal components, effective reduction of the input vector is achieved by setting a heuristic criterion to cut off the least discriminant components.

For each classification problem, the basicBP algorithm is used for training the MLP using the reduced vector of principal components.

3.4 Feature Selection through relevance-based Embedded Approach

In the third experiment, the Optimal Cell Damage (OCD) [Cibas et al. 1994] algorithm is used. OCD is an embedded feature selection

Multi-Layer Perceptron Settings	Ionosphere	LandSat	Musk	Segmentation	Vehicle	Vowel
Input nodes	33	36	166	19	18	10
Output nodes	2	6	2	7	4	11
Hidden layers	1	1	1	1	1	1
Hidden nodes	2	30	20	30	30	40
Act. function of hidden nodes	Hyper-tangent					
Act. function of output nodes	Sigmoidal					

Learning Algorithms Settings	BasicBP	OCD	Wrapper	FeaSANNT
Learning coefficient	0.1	0.1	n.a.	n.a.
Momentum term	0.01	0.01	n.a.	n.a.
Init. range for MLP weights	[-0.05, 0.05]	[-0.05, 0.05]	n.a.	[-0.05, 0.05]
Training subset (% training data)	n.a.	80 %	80 % **	n.a.
Validation subset (% training data)	n.a.	20 %	20 % **	n.a.
Trials	10	10	10	10
Generations	*	*	75	*
Population size	n.a.	n.a.	30	100
Feature mask crossover rate	n.a.	n.a.	1.0	1.0
MLP weights mutation rate	n.a.	n.a.	n.a.	0.25
Amplitude MLP weights mut.	n.a.	n.a.	n.a.	0.2
BP mutation rate	n.a.	n.a.	n.a.	0.6
Feature mask mutation rate	n.a.	n.a.	0.2	0.05
Masked features at start	n.a.	n.a.	10 % of total	10 % of total
Saliency threshold	n.a.	*	n.a.	n.a.

* (depending upon data set); ** (BP training of MLP); n.a. (not applicable)

Table 2: Parameter Setting of Multi-Layer Perceptron and Learning Algorithms

method that uses an attribute relevance criterion called "feature saliency". The OCD algorithm is described in figure 2.

The classifier is initially trained using the BP rule. The duration of the BP procedure is determined using the early stopping criterion. That is, the training set is divided into a learning subset and a validation subset. The classifier is trained on the former and the learning accuracy is monitored on the latter. When the classification accuracy on the validation subset stops improving, learning is terminated.

Once the MLP is trained, the saliency of each feature is estimated through the Hessian matrix of the mean-square-error on the learning set. Variables whose saliency is below an empirically fixed threshold are eliminated. Using the remaining attributes, the ANN mean-square-error is again minimised via a new BP phase. ANN training cycles and feature selection cycles are repeated until no further variable elimination is possible without deterioration of the ANN accuracy.

3.5 Feature Selection through Evolutionary Wrapper Approach

In the fourth experiment, a standard genetic wrapper procedure is used. The algorithm uses the feature selection module of FeaSANNT to search the attribute space, and the BP rule with momentum term for training the classifier. A flowchart of the wrapper algorithm is given in figure 3.

The genetic wrapper algorithm keeps the structure of FeaSANNT. Since evolution concerns now only the feature space, individuals are characterised by the sole binary string defining the input mask. This genotype is manipulated by the operators of the feature selection module. Each generation, the solutions are evaluated using the procedure described in subsection 2.3. That is, for each solution an MLP is randomly initialised and trained using the reduced feature vector defined by the input mask.

Since different input vector configurations often determine different ANN learning speeds, the duration of the BP procedure is determined for every solution using the early stopping criterion. At every evolutionary cycle, the learning and the validation subsets are randomly re-initialised, and all the solutions are trained and evaluated on the same learning and validation subsets.

The pool of reproducing individuals is selected via the method described in subsection 2.4. The offspring population fully replaces the parent population (generational replacement).

Ionosphere	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	94.2	95.6	0.4	94.6	0.02	92.0	0.64	91.7	0.65
Std_Dva	2.3	0.6		1.4		1.7		2.1	
LandSat	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	89.3	89.7	0.27	87.1	**6.75**	85.5	**12.42**	86.0	**16.72**
Std_Dva	0.6	0.5		0.7		0.9		0.6	
Musk	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	98.5	99.2	1.84	98.7	0.11	97.4	1.67	98.4	0.05
Std_Dva	0.4	0.3		0.4		0.7		0.3	
Segmentation	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	97.0	97.2	0.05	93.1	**5.37**	94.4	**5.09**	94.4	**8.43**
Std_Dva	0.6	0.5		1.6		1.0		0.6	
Vehicle	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	81.9	82.7	0.04	61.1	**23.49**	77.7	1.22	75.2	3.90
Std_Dva	2.5	2.9		3.5		2.8		2.3	
Vowel	FeaSANNT	Basic BP	F-Test	PCA + BP	F-Test	OCD	F-Test	Wrapper	F-Test
Accuracy	54.3	55.2	0.08	55.63	0.17	53.55	0.03	52.7	0.11
Std_Dva	2.8	1.6		1.78		2.65		3.9	

Table 3: Experimental Results – Learning Accuracy

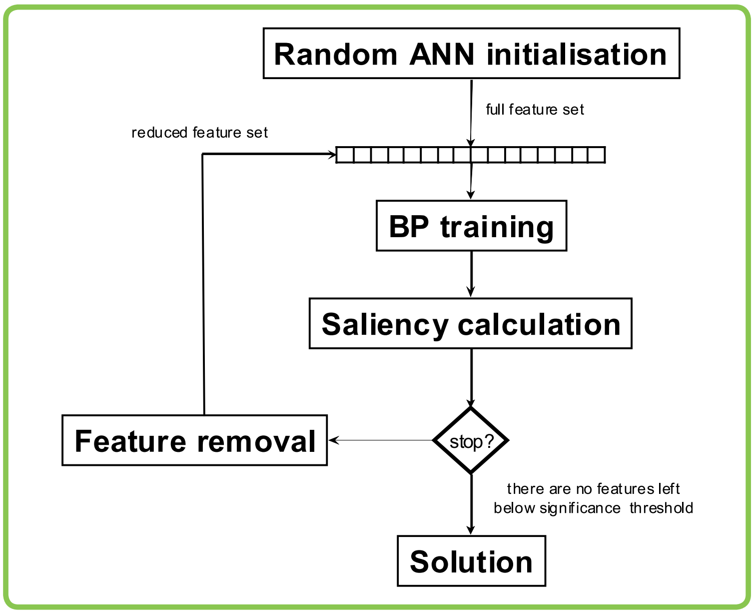

Figure 2: OCD Architecture

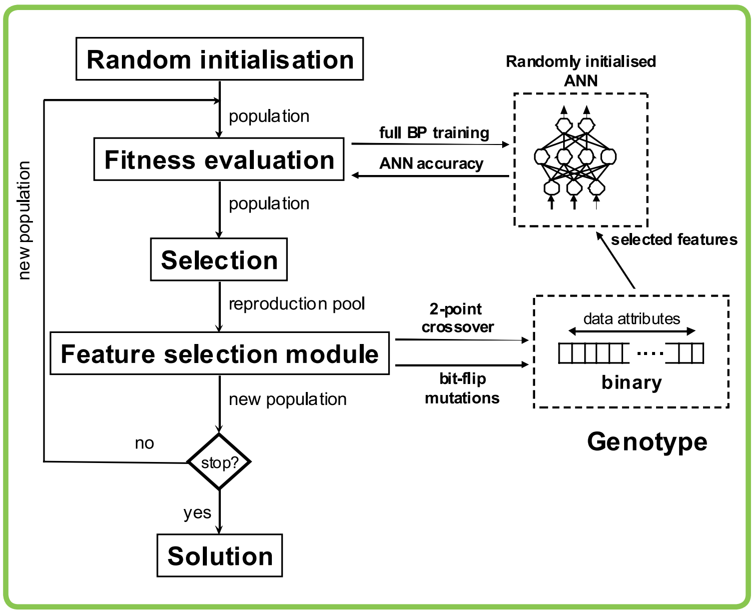

Figure 3: Genetic Wrapper Algorithm Architecture

The genetic wrapper procedure is run for a pre-defined number of iterations. At the end of the last generation, the population is evaluated and the fittest individual is picked as the final solution.

4 Experimental Results

This section presents the experimental settings and the results obtained by FeaSANNT and the four control algorithms on the six benchmark problems.

4.1 Experimental settings

Input data are normalised according to the mean-variance procedure. In the learning trials involving the three largest data bases, a sampling procedure is used to reduce the computational overheads of the algorithms. The sampling method randomly selects a subset of the training data from each class of patterns of a database, at each learning cycle, and uses this sample instead of the full training set. The size of the subset is 10 % of the training examples for the Musk database, 20 % of the training examples for the LandSat database, and 50 % of the training examples for the Image Segmentation database.

Principal components are calculated using the correlation matrix of the input variables. The heu-

	FeaSANNT	Full Set	PCA + BP	OCD	Wrapper
Ionosphere	11.9	33.0	19.0	9.4	22.3
LandSat	24.4	36.0	4.0	20.9	26.0
Musk	58.2	166.0	26.0	57.1	140.3
Segmentation	9.7	19.0	9.0	4.6	12.9
Vehicle	16.5	18.0	5.0	11.8	13.9
Vowel	8.7	10.0	7.0	8.8	8.4

Table 4: Experimental Results – Feature Selection

OCD	Ionosphere	LandSat	Musk	Segmentation	Vehicle	Vowel
Saliency threshold	0.50	0.05	0.05	0.40	0.30	0.10

Table 5: Variation of OCD Saliency Threshold

ristic variable selection criterion retains only that number of first principal components that is sufficient to account for at least 90 % of the total data variance.

The learning algorithms and the topology of the MLPs are optimised according to trial and error. Due to the lengthy training of the classifier, the duration of the genetic wrapper algorithm is limited to a small population and a reduced number of generations. To sustain the exploration of the search space, the wrapper algorithm uses a high mutation rate.

Table 2 reports the parameter settings. Once the parameters of an algorithm are optimised, they are kept unchanged for all the experiments. The only exceptions are the duration of the algorithms, and the OCD feature saliency threshold that require a specific setting for each benchmark problem.

Experimental evidence shows that the performance of FeaSANNT is robust to reasonable variations of the search parameters. Learning times ranged from a few minutes for the smaller data sets to several hours for the larger data sets on a PentiumIII 1 GHz processor with 512 MB of RAM.

For each benchmark problem, FeaSANNT and the four control algorithms are run ten times with different random initialisations. The learning results are estimated as the average values of the ten independent learning trials.

4.2 Learning Results

For each algorithm, table 3 reports the mean and the standard deviation of the ANN accuracy over the ten learning trials for each classification task. The statistical significance of the differences between the accuracy results obtained by FeaSANNT and the results obtained by the control algorithms is evaluated through ANOVA F-tests. Table 3 includes the results of the ANOVA tests and the critical value for a 5 % alpha level of significance. Results above the critical value are reported in bold.

For each benchmark problem, table 4 reports the average number of selected features.

Experimental evidence shows that FeaSANNT reduces considerably the input features in most of the data sets. No statistically significant differences are found between the accuracy results obtained by FeaSANNT and the results obtained by the basicBP algorithm on the full feature set. Overall, the tests prove the ability of FeaSANNT of rejecting a large number of features without affecting the MLP accuracy.

4.3 Comparison with Control Algorithms

The lack of correlation between the feature selection criterion and the learning procedure of the classifier makes the feature vectors produced by filter algorithms not always optimal for learning. This problem is confirmed by the poor learning results obtained by the basicBP algorithm when using the PCA-generated feature subsets on the LandSat, Image Segmentation and Vehicle data sets.

Standard wrapper and embedded approaches select features according to the learning results of the classifier. Due to their greedy feature selection approaches, standard wrapper and embedded algorithms are prone to sub-optimal convergence, and they can not capture interactions between attributes. These shortcomings are confirmed by the poor learning results obtained by the OCD algorithm on the LandSat and Image Segmentation classification problems. Moreover, standard approaches require often time-consuming tuning of their learning parameters. Table 5 shows the wide variation over the six data sets of the optimal value for the OCD feature saliency threshold.

EAs are able to capture interactions between variables, and can escape local subpeaks of accuracy. Unfortunately, the customary wrapper approach to evolutionary feature selection entails a high computational cost that hinders the extent of the evolutionary search. This problem accounts for the mediocre results obtained by the genetic wrapper algorithm. Examination of the evolution curves shows that the wrapper algorithm is often terminated before it converges to a solution. Nonetheless, due to the computational complexity of the fitness evaluation procedure, the genetic wrapper algorithm requires much longer execution times than FeaSANNT and the other algorithms.

Despite the removal of an often large number of features, none of the feature selection algorithms improves the accuracies obtained using the full attribute set. This result suggests the presence of superfluous attributes, rather than features that negatively affect the separability of the classes. Nonetheless, feature selection is of primary importance since it reduces the cost of collecting and pre-processing the data.

5 Conclusions

This paper presented FeaSANNT, an evolutionary feature selection procedure for ANNs. FeaSANNT exploits the global nature of the evolutionary search to escape sub-optimal peaks of performance. The novelty of the method lies in the implementation of the embedded approach in an evolutionary feature selection paradigm. Experimental evidence proves the ability of FeaSANNT of rejecting an often large number of features without affecting the ANN accuracy.

The performance of FeaSANNT is compared to the performance of a well-known filter approach (PCA), a popular embedded approach (OCD), and a standard genetic wrapper algorithm. None of the control algorithms attains the consistency of learning results attained by FeaSANNT. PCA and OCD often select smaller feature sets than FeaSANNT, but the smaller attribute sets not always allow acceptable learning results. The genetic wrapper algorithm achieves sub-standard results in terms of both classification accuracy and feature rejection.

Compared to the customary evolutionary wrapper approach, the proposed embedded approach simplifies the algorithm design since it removes the need of setting the stopping criterion for the ANN learning algorithm.

Thanks to the robustness of the evolutionary search, FeaSANNT solves all the learning tasks using a unique parameter setting. This robustness compares favourably with the sensitivity of many standard feature selection algorithms to critical learning parameters. An example of the latter case is the variation of the optimal OCD feature saliency threshold over the six benchmark problems.

6 Acknowledgements

The attendance to this conference was sponsored by the EC FP6 Network of Excellence on Innovative Production Machines and Systems (I*PROMS).

7 References

Aboitiz, F.: Mechanisms of adaptive evolution – Darwinism and Lamarckism restated. Medical Hypotheses, 38 (3), pp. 194–202, 1992

Blum, A.; Langley, P.: Selection of Relevant Features and Examples in Machine Learning, Artificial Intelligence, 97, pp. 245–271, 1997

Cantu-Paz; Kamath, C.: An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. IEEE Trans. on Syst., Man, and Cyb., part B, 35 (5), pp. 915–927, 2005

Cibas, T.; Fogelman Soulie, F.; Gallinari, P.; Raudys, S.: Variable Selection with Optimal Cell Damage. Proceedings International Conference on Artificial Neural Networks, Sorrento, I, (1), pp. 727–730, 1994

Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 2nd ed., IEEE Press, New York, 2000

Mardia, V.; Kent, J.T.; Bibby, J.M.: Multivariate Analysis. Academic Press, London, 1979

Pham, D.T.; Liu, X.: Neural Networks for Identification. Prediction and Control, Springer-Verlag Ltd., London, UK, 1995

Portinale, L.; Saitta, L.: Feature Selection. Deliverable, D14.1 IST Project MiningMart, IST-11993, 2002

Schetinin, V.: A Learning Algorithm for Evolving Cascade Neural Networks. Neural Processing Letters 17, pp. 21–31, 2003

Thierens, D.; Suykens, J.; Vanderwalle, J.; De Moor, B.: Genetic Weight Optimisation of a Feedforward Neural Network Controller. In Albrecht, R.F.; Reeves, C.R.; Steele, N.C. (editors): Artificial Neural Networks and Genetic Algorithms, Springler-Verlag, Wien, A, pp. 658–663, 1993

UCI Machine Learning Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html

Yao, X.: Evolving Artificial Neural Networks. Proceedings IEEE, 87(9), pp. 1423–1447, 1999

Zhang, P.; Verma, B.; Kumar, K.: Neural vs. Statistical Classifier in Conjunction with Genetic Algorithm Based Feature Selection. Pattern Recognition Letters, 26, pp. 909–919, 2005

Marco Castellani
castellanim@Cardiff.ac.uk

M. Castellani obtained his Ph.D. in 2000 from Cardiff University. In 2001 and 2002, he worked for a private company in Germany. He was at Universidade Nova de Lisboa between 2002 and 2005. He is currently at the Manufacturing Engineering Centre of Cardiff University. His expertise includes Machine Learning, Intelligent Control, Pattern Recognition, and Machine Vision.

N.C. Marques
nmm@di.fct.unl.pt

N.C. Marques obtained his Ph.D. in Computer Science in january 2000 at the Faculdade de Ciências e Tecnologia of Universidade Nova de Lisboa (FCT-UNL). Since january 2000, he has been member of the Centre for Artificial Intelligence of FCT-UNL, where he currently works as an assistant professor. His research interests include machine learning, remote sensing, and data mining.