Experimental Comparison of Feature Subset Selection Using GA and ACO Algorithm

Keunjoon Lee[1], Jinu Joo[2], Jihoon Yang[3,⋆,⋆⋆], and Vasant Honavar[4]

[1] Kookmin Bank, Sejong Daewoo B/D
167 Naesu-Dong, Jongno-Ku, Seoul 110-070, Korea
leekjsg@hanmail.net
[2] Development Laboratory 1, Mobile Handset R&D Center
Mobile Communications Company, LG Electronics Inc.
Gasan-Dong, Gumchon-Ku, Seoul 153-801, Korea
jujoo@lge.com
[3] Department of Computer Science, Sogang University
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
yangjh@sogang.ac.kr
[4] Artificial Intelligence Research Laboratory, Department of Computer Science
Iowa State University Ames, IA 50011 USA
honavar@cs.iastate.edu

Abstract. Practical pattern classification and knowledge discovery problems require selecting a useful subset of features from a much larger set to represent the patterns to be classified. Exhaustive evaluation of possible feature subsets is usually infeasible in practice because of the large amount of computational effort required. Bio-inspired algorithms offer an attractive approach to find near-optimal solutions to such optimization problems. This paper presents an approach to feature subset selection using bio-inspired algorithms. Our experiments with several benchmark real–world pattern classification problems demonstrate the feasibility of this approach to feature subset selection in the automated design of neural networks for pattern classification and knowledge discovery.

1 Introduction

In practical pattern classification and knowledge discovery problems, many input data contain large amount of features or attributes which are mutually redundant and irrelevant with different associated measurements. Among these large number of features, selecting useful subset of features to represent the patterns that are presented to a classifier mainly affect the accuracy, time, the number of examples needed for learning a sufficiently accurate classification function, the cost of performing classification using the learned classification function, and the comprehensibility of the knowledge acquired through learning. Therefore this

⋆ This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation and by the Brain Korea 21 Project in 2006.

⋆⋆ Corresponding author.

X. Li, O.R. Zaiane, and Z. Li (Eds.): ADMA 2006, LNAI 4093, pp. 465–472, 2006.

presents us with a feature subset selection problem in pattern classification tasks. The feature subset problem is to identify and select a useful subset of features in order to use to represent patterns from a much larger set of features. Many feature subset selection methods have been introduced for automated design for pattern classifiers. We introduce a new feature subset selection approach based on bio-inspired algorithms and selected feature subsets evaluated by a neural network (DistAI). We present our experimental results from various experiments and prove our methods usability with several benchmark classification problems.

2 Feature Selection Using Bio-inspired Algorithms for Neural Network Pattern Classifiers

Among a number of bio-inspired algorithms, we consider the GA and ACO algorithm in this paper.

2.1 Genetic Algorithm

Evolutionary algorithms [1,2,3,4] include a class related randomized, population-based heuristic search techniques which include genetic algorithms [1,2], genetic programming [3], evolutionary programming [4]. They are inspired by processes that are modeled after biological evolution. The individuals represent candidate solutions to the optimization problem being solved. A wide range of genetic representations (e.g. bit vectors, LISP programs, matrices, etc.) can be used to encode the individuals depending on the space of solutions that needs to be searched. In the feature subset selection problem, each individual would represent a feature subset. It is assumed that the quality of each candidate solution (or fitness of the individual in the population) can be evaluated using a fitness function. In the feature subset selection problem, the fitness function would evaluate the selected features with respect to some criteria of interest (e.g. cost of the resulting classifier, classification accuracy of the classifier, etc.).

2.2 ACO Algorithm

The ant algorithm is a heuristic search algorithm using artificial ants known as multi-agents which run parallel when constructing feasible solutions probabilistically based on pheromone information deposited upon each plausible candidate solution or trail. The early version of the ant algorithm introduced was known as ant system (AS) [5] algorithm by Dorigo. Recently variants of ant algorithm were combined in a common frame work called ant colony optimization (ACO) meta-heuristic [6]. In this paper we have adopted the graph based ant system (GAS) [7] which has been mentioned by Gutjahr. GAS is a specific version of ACO meta-heuristic algorithm where candidate solutions can be represented in directed graphs. It is particularly successful in solving combinatorial optimization problems such as constructing paths based on direct graphs with specific starting points. GAS updates pheromone globally: pheromone trail is updated after all ants have traveled in its cycle, and provides a pheromone evaporation factor to prevent ants converging into local minima. In our ant algorithm elite policy is used for updating pheromone information on each trail. Throughout this paper algorithms that follow the ACO meta-heuristic will be called ACO algorithm.

2.3 DistAl: A Fast Algorithm for Constructing Neural Network Pattern Classifiers

Because feature subset selection method powered by ACO algorithm require numerous cycles of running the ACO algorithm itself and each cycle contains a lot of ants holding candidate solutions to be evaluated by training the neural network, it is not feasible to use computationally expensive iterative weight update algorithms. Consequently DistAl, offering a fast and efficient approach in training neural networks, is used for evaluating the fitness of the candidate solution. DistAl [8] is a simple and relatively fast constructive neural network learning algorithm for pattern classification. The results presented in this paper are based experiments using neural networks constructed by DistAl. The key idea behind DistAl is to add *hyperspherical* hidden neurons one at a time based on a greedy strategy which ensures that the hidden neuron correctly classifies a maximal subset of training patterns belonging to a single class. Correctly classified examples can then be eliminated from further consideration. The process terminates when the pattern set becomes empty (that is, when the network correctly classifies the entire training set). When this happens, the training set becomes linearly separable in the transformed space defined by the hidden neurons. In fact, it is possible to set the weights on the hidden to output neuron connections without going through an iterative process. It is straightforward to show that DistAl is guaranteed to converge to 100% classification accuracy on any finite training set in time that is polynomial in the number of training patterns [8]. Experiments reported in [8] show that DistAl, despite its simplicity, yields classifiers that compare quite favorably with those generated using more sophisticated (and substantially more computationally demanding) learning algorithms. This makes DistAl an attractive choice for experimenting with social intellectual approaches to feature subset selection for neural network pattern classifiers.

3 Implementation Details

In this section we explain our implementation details on GA and ACO algorithms which are utilized in our feature subset selection problem.

3.1 GA Implementation

Our GA algorithm is based on rank-based selection strategy described in Figure 1. The rank based selection strategy gives a non-zero probability of selection of each individual [9]. For more specific implementation details look at [10].

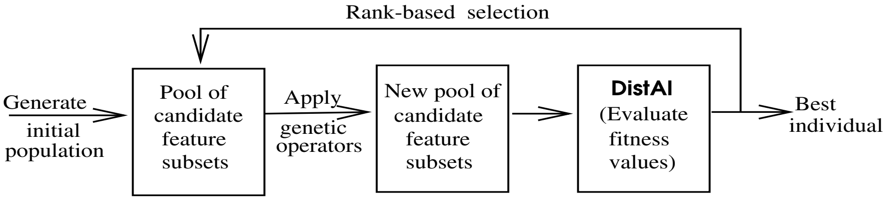
Rank-based selection
Generate initial population
Pool of candidate feature subsets
Apply genetic operators
New pool of candidate feature subsets
DistAI (Evaluate fitness values)
Best individual

Fig. 1. GADistAI: Feature subset selection using a genetic algorithm with DistAI

3.2 ACO Implementation

Our ACO algorithm is based on Gutjahr's GAS algorithm [7] with the following adjustments.

- Representation:
  Each traversed path by an ant in a cycle represents a candidate solution to the feature subset selection problem. As described in Figure 2, the selected features are represented as combination of arcs where ants have traversed through the graph. Note that every ant must visit every node in the graph no more than once and every ant starts at a specific node (first feature) and ends at a specific node (last feature) visiting every node in between with a given sequence. Every node has two arcs connected to its next visiting node, each representing either selection or exclusion of the feature it is assigned to. Therefore combining traversed arcs together gives a full representation of a candidate solution of feature selection, defined as a path, to classify the given dataset.

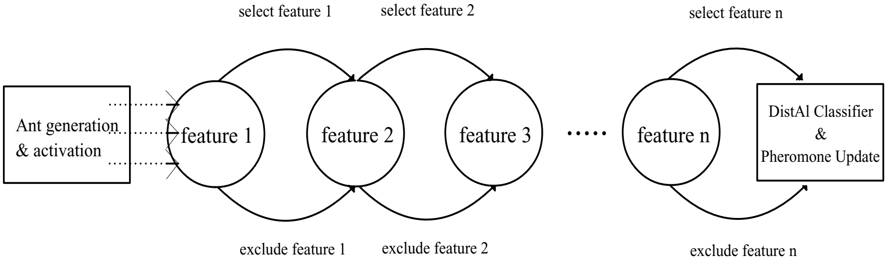

Fig. 2. ACODistAI: Feature subset selection using ACO algorithm with DistAI

- Definition of pheromone update rule with *elite policy*:
  Each node has two choices of arc leading to the next neighboring node. That is, if $\forall i, j \in V$ then $\forall (i,j) \in E$ is $(i,j) = (i,j)^+ \cup (i,j)^-$ where V is the set of nodes in the graph, j the next visiting node from i, E the set of arcs in the graph, and $(i,j)^+, (i,j)^-$ are selection and exclusion arcs from node i to j respectively. Therefore the initial pheromone on each trail

is, $\tau_{ij} = 0.5 = 1/$(number of arcs possible to traverse from node i). Unlike GAS algorithm, we introduce an *elite policy* to guide our pheromone update on each path. Pheromone updates occur on paths that outperform the best path in the previous cycle. In other words, paths that perform better than the previous best are the only paths considered to deposit more pheromone on the trail. The partial pheromone deposited on each arc by each ant is,

$$\Delta\tau_{ij}^{s} = \begin{cases} \mu(p_m^s) & \text{if } \mu_{m-1}^{*} \leq \mu(p_m^s) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where μ_{m-1}^{*} is the best performance measurement performed at cycle $m-1$, and p_m^s is the path built by ant s at cycle m. From (1) the total pheromone deposited on each arc when an elite path is found is $\Delta\tau_{ij} = \frac{1}{C}\sum_{s=1}^{S} \Delta\tau_{ij}^{s}$,

where C is the normalization factor defined as $C = \sum_{(i,j)} \sum_{s=1}^{S} \Delta\tau_{ij}^{s}$. Therefore the pheromone update rule is $\tau_{ij}(m+1) = (1-\rho)\tau_{ij}(m) + \rho\Delta\tau_{ij}$,
where ρ is the evaporation factor and m is the number of cycles performed so far. On the contrary, if an elite has been found on the m_{th} cycle then the pheromone update rule is $\tau_{ij}(m+1) = \tau_{ij}(m)$.

- Definition of transition probability:
  From the pheromone update rule introduced above, the transition probability is estimated as,

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{i,k\in A}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}} \tag{2}$$

  where η is the heuristic value, and α, β are parameters.
- Setting of user-controlled parameters:
  Iteration of ACODistAl performed : 5; Number of ant : 50; Number of cycle : 20; Evaporation factor ρ : 0.3; Heuristic value η : 1; Transition probability parameter : $\alpha = 1.0$, $\beta = 1.0$.

4 Experiments

4.1 Description of Datasets

The experiments reported here used a wide range of real-world datasets from the machine learning data repository at the University of California at Irvine [11] as well as a carefully constructed artificial dataset (3-bit parity) to explore the feasibility of using bio-inspired algorithms for feature subset selection for neural network classifiers [1]. The feature subset selection using DistAl is also applied to document classification problem for journal paper abstracts. For more details on datasets see [10].

[1] [http://www.ics.uci.edu/ mlearn/MLRepository.html]

Table 1. Datasets used in the experiments. *Size* is the number of patterns in the dataset, *Features* is the number of input features, and *Class* is the number of output classes.

Dataset	*Size*	*Features*	*Feature Type*	*Class*
3-bit parity problem (**3P**)	100	13	numeric	2
audiology database (**Audiology**)	200	69	nominal	24
pittsburgh bridges (**Bridges**)	105	11	numeric, nominal	6
breast cancer (**Cancer**)	699	9	numeric	2
credit screening (**CRX**)	690	15	numeric, nominal	2
flag database (**Flag**)	194	28	numeric, nominal	8
heart disease (**Heart**)	270	13	numeric, nominal	2
heart disease [Cleveland](**HeartCle**)	303	13	numeric, nominal	2
heart disease [Hungarian](**HeartHun**)	294	13	numeric, nominal	2
heart disease [Long Beach](**HeartLB**)	200	13	numeric, nominal	2
heart disease [Swiss](**HeartSwi**)	123	13	numeric, nominal	2
hepatitis domain (**Hepatitis**)	155	19	numeric, nominal	2
horse colic (**Horse**)	300	22	numeric, nominal	2
ionosphere structure (**Ionosphere**)	351	34	numeric	2
pima indians diabetes (**Pima**)	768	8	numeric	2
DNA sequences (**Promoters**)	106	57	nominal	2
sonar classifiction (**Sonar**)	208	60	numeric	2
large soybean (**Soybean**)	307	35	nominal	19
vehicle silhouettes (**Vehicle**)	846	18	numeric	4
house votes (**Votes**)	435	16	nominal	2
vowel recognition (**Vowel**)	528	10	numeric	11
wine recognition (**Wine**)	178	13	numeric	3
zoo database (**Zoo**)	101	16	numeric, nominal	7
paper abstracts 1 (**Abstract1**)	100	790	numeric	2
paper abstracts 2 (**Abstract2**)	100	790	numeric	2

4.2 Experimental Results

The experiment explored the performance of ACODistAI, comparing it with GA-based approaches for feature subset selection. The parameter setting described in Section 3 was chosen for fair comparison of ACODistAI with GADistAI.

Fitness evaluation was obtained by averaging the observed fitness value for 10 different partitions of the data into training and test sets. The final results are estimated by averages over 5 independent runs of the algorithm which are shown in Table 2. The entries in the tables give the means and standard deviations in the form *mean ± standard deviation*. The results of Table 2 show that, in most of the datasets ACODistAI and GADistAI perform better than the original DistAI with full feature sets. Datasets with similar accuracy among the three algorithm show that ACODistAI and GADistAI can perform with high accuracy with almost half the features used to classify the dataset. For example nearly 90 $\sim$ 95% accuracies were yielded in **Cancer**, **HeartSwi**, **Promoters**, **Votes** and **Abstract1** datasets, where ACODistAI and GADistAI classified each dataset with almost half the features used in DistAI. Contrary to the fact that

Table 2. Comparison of neural network pattern classifiers constructed by DistAl using the entire set of features with the best network constructed by GADistAl and ACODistAl using fitness estimates based on 10-fold cross-validation.

	DistAl		GADistAl		ACODistAl	
Dataset	*Features*	*Accuracy*	*Features*	*Accuracy*	*Features*	*Accuracy*
3P	13	79.0±12.2	4.8 ± 0.7	100.0 ± 0.0	10.8 ± 0.4	100 ± 0.0
Audiology	69	66.0±9.7	37.2 ± 1.8	72.6 ± 2.8	31.2 ± 2.5	68.2 ± 2.4
Bridges	11	63.0 ± 7.8	4.9 ± 0.6	56.9 ± 7.6	5.8 ± 1.5	67.6 ± 1.8
Cancer	9	97.8 ± 1.2	6.0 ± 1.1	98.0 ± 0.3	5.4 ± 0.8	97.7 ± 0.1
CRX	15	87.7 ± 3.3	7.4 ± 2.6	87.7 ± 0.4	6.8 ± 1.8	89.6 ± 0.2
Flag	28	65.8 ± 9.5	14.2 ± 2.8	63.9 ±6.1	14.2 ± 2.3	68.3 ± 0.5
Heart	13	86.7 ± 7.6	7.6 ± 0.8	85.5 ±0.7	9.4 ± 1.2	87.2 ± 0.4
HeartCle	13	85.3 ± 2.7	8.4 ± 0.8	86.9 ±0.6	12.6 ± 0.5	84.1 ± 0.5
HeartHun	13	85.9 ± 6.3	7.4 ± 1.4	85.4 ±1.3	6.8 ± 1.8	88.4 ± 0.2
HeartLB	13	80.0 ± 7.4	7.6 ± 1.0	79.8 ±1.9	7.8 ± 2.2	82.3 ± 0.2
HeartSwi	13	94.2 ± 3.8	7.4 ± 1.7	95.3 ±1.1	6.2 ± 2.1	95.8 ± 0.0
Hepatitis	19	84.7 ± 9.5	10.2 ± 1.6	85.2 ±2.9	17 ± 0.0	84.1 ± 0.8
Horse	22	86.0 ± 3.6	9.6 ± 2.7	83.2 ±1.6	11.4 ± 2.4	85.5 ± 1.1
Ionosphere	34	94.3 ± 5.0	16.6 ± 3.0	94.5 ± 0.8	17.4 ± 1.6	95.4 ± 0.8
Pima	8	76.3 ± 5.1	4.0 ± 1.7	73.1 ±3.1	3.8 ± 1.0	77.6 ± 0.0
Promoters	57	88.0 ± 7.5	30.6 ± 2.1	89.8 ±1.7	30.6 ± 4.7	94.3 ± 0.9
Sonar	60	83.0 ± 7.8	32.2 ± 2.2	84.0 ±1.6	31 ± 4.1	79.6 ± 1.0
Soybean	35	81.0 ± 5.6	21.0 ± 1.4	83.1 ± 1.1	18.2 ± 2.8	43.4 ± 2.9
Vehicle	18	65.4 ± 3.5	9.4 ± 2.1	50.1 ± 7.9	9.4 ± 1.6	68.8 ± 0.7
Votes	16	96.1 ± 1.5	8.2 ± 1.5	97.0 ±0.7	8.6 ± 2.1	97.2 ± 0.2
Vowel	10	69.8 ± 6.4	6.8 ± 1.2	70.2 ±1.6	4.2 ± 1.6	49.0 ± 0.4
Wine	13	97.1 ± 4.0	8.2 ± 1.2	96.7 ±0.7	5.4 ± 0.5	95.1 ± 0.5
Zoo	16	96.0 ± 4.9	8.8 ± 1.6	96.8 ±2.0	9.4 ± 0.8	95.6 ± 0.5
Abstract1	790	89.0±9.4	402.2 ± 14.2	89.2 ± 1.0	387.2 ± 10.4	90.0 ± 1.1
Abstract2	790	84.0±12.0	389.8 ± 5.2	84.0 ± 1.1	401.0 ± 9.1	88.4 ± 0.5

most of the datasets yield similar performances between ACODistAl and GADistAl, some datasets like **Heart**, **HeartHun**, and **Promoters** showed specifically higher accuracies in ACODistAl compared to the other methods. However, the performance of ACODistAl is much worse in **Soybean** and **Vowel** datasets. We surmise that our current implementation of ACO is not appropriate for those particular problems.

5 Summary and Discussion

GADistAl and ACODistAl are methods to feature subset selection for neural network pattern classifiers. In this paper a fast inter-pattern distance-based constructive neural network algorithm, DistAl, is employed to evaluate the fitness of candidate feature subsets in the ACO algorithm. The performance of ACODistAl was comparable to the GA based approach (GADistAl), both of which outperformed DistAl significantly. The results presented in this paper indicate that

ACO algorithms offer an attractive approach to solving the feature subset selection problem in inductive learning of pattern classifiers in general, and neural network pattern classifiers in particular.

Some directions for future research include: Extensive experiments on alternative datasets including documents and journals; Extensive experimental (and wherever feasible, theoretical) comparison of the performance of the proposed approach with that of other bio-inspired algorithm–based and conventional methods for feature subset selection; More principled design of multi-objective fitness functions for feature subset selection using domain knowledge as well as mathematically well-founded tools of multi-attribute utility theory [12].

References

1. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, New York (1989)
2. Holland, J.: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, MA (1992)
3. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA (1992)
4. Fogel, D.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway, NJ (1995)
5. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: An autocatalytic optimizing process (1991)
6. Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, London (1999) 11–32
7. Gutjahr, W.J.: A graph-based ant system and its convergence. Future Gener. Comput. Syst. **16**(9) (2000) 873–888
8. Yang, J., Parekh, R., Honavar, V.: Distal: An inter-pattern distance-based constructive learning algorithm. In: Proceedings of the International Joint Conference on Neural Networks, Anchorage, Alaska (1998) 2208 – 2213
9. Mitchell, M.: An Introduction to Genetic algorithms. MIT Press, Cambridge, MA (1996)
10. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. In Motoda, Liu, eds.: Feature Extraction, Construction and Selection - A Data Mining Perspective. Kluwer Academic Publishers (1998) 117–136
11. Murphy, P., Aha, D.: Uci repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA (1994)
12. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York (1976)