http://dx.doi.org/10.5392/JKCA.2011.11.10.041

특징 선택에서 선택적 평가를 사용하는 개미 군집 최적화의 수렴 특성

Convergence Characteristics of Ant Colony Optimization with Selective Evaluation in Feature Selection

이진선[*], 오일석[**]
우석대학교 게임콘텐츠학과[*], 전북대학교 컴퓨터공학부/영상정보신기술연구소[**]

Jin-Seon Lee(jslee@woosuk.ac.kr)[*], Il-Seok Oh(isoh@chonbuk.ac.kr)[**]

요약

최근 특징 선택에서 개미군집 최적화를 위한 선택적 평가 기법이 제안되었다. 이 기법은 불필요하거나 가능성이 적은 후보 해를 실제 평가 과정에서 제외함으로써 계산량을 줄인다. 실험을 통해 이 기법의 우수성을 보였으나, 하나의 데이터만을 사용하였으므로 통계적으로 충분한 신뢰성을 보여주지 못한다. 이 논문의 목적은 선택적 평가 기법의 수렴 특성을 분석하고 결론의 신뢰성을 높이는 것이다. 실험을 위해 UCI 데이터베이스에서 필기, 의료, 음성에 관련된 세가지 데이터를 선택하였다. 이들의 특징 집합 크기는 256부터 617까지 분포한다. 통계적으로 안정된 데이터를 얻기 위해, 이들 각각에 대해 프로그램을 독립적으로 12번 실행하였다. 긴 시간에 걸친 수렴을 관찰하기 위해, 각각의 프로그램 실행은 72시간 동안 이루어졌다. 실험 데이터의 분석을 바탕으로, 선택적 평가 기법의 우수성에 대한 이유와 이 기법의 적용 범위에 대해 기술한다.

■ 중심어 : | 패턴인식 | 특징선택 | 수렴 | 메타휴리스틱 | 개미 군집 최적화 | 선택적 평가 |

Abstract

In feature selection, the selective evaluation scheme for Ant Colony Optimization(ACO) has recently been proposed, which reduces computational load by excluding unnecessary or less promising candidate solutions from the actual evaluation. Its superiority was supported by experimental results. However the experiment seems to be not statistically sufficient since it used only one dataset. The aim of this paper is to analyze convergence characteristics of the selective evaluation scheme and to make the conclusion more convincing. We chose three datasets related to handwriting, medical, and speech domains from UCI repository whose feature set size ranges from 256 to 617. For each of them, we executed 12 independent runs in order to obtain statistically stable data. Each run was given 72 hours to observe the long-time convergence. Based on analysis of experimental data, we describe a reason for the superiority and where the scheme can be applied.

■ keyword : | Pattern Recognition | Feature Selection | Convergence | Meta-heuristics | Ant Colony Optimization | Selective Evaluation |

I. Introduction

One of the most important tasks in various modern applications such as pattern recognition, machine learning, information retrieval, bioinformatics, and data mining is to design discriminatory features. An

* 이 논문은 2011학년도 우석대학교 교내학술연구비 지원에 의하여 연구되었음
접수번호 : #110907-003
접수일자 : 2011년 09월 07일
심사완료일 : 2011년 09월 23일
교신저자 : 이진선, e-mail : jslee@woosuk.ac.kr

effective method of detecting near-optimal feature sets is the feature selection, which reduces feature set size by removing useless, redundant, or less useful features[1]. The algorithms being used must be efficient in terms of computation time and effective for finding near-optimal solutions.

The ant colony optimization (ACO) is gaining popularity as a new approach to the feature selection [2-7]. Some literatures provided experimental results that illustrated ACO's superiority over the conventional approaches such as sequential search and the genetic algorithm[5]. The ACO me ta-heuristic has been developed as a methodology for combinatorial optimization problems[8]. In ACO, a collection of artificial ants simulate the foraging behavior of real ants which cooperate in order to carry food from a food source to their nest. The search space of a given problem is represented as a graph, and qualities of solutions are represented by pheromones put on graph edges. The pheromone is essential information over whole process. For a general understanding of ACO for subset selection problems, we refer the readers to [9].

An inherent difficulty of the feature selection is the high computational demand required when the problem size is large. Problem size depends on two factors, the number of features (D), and the number of samples in the training set (N). For example, it is common in handwritten numeral recognition for D to be several hundreds, and for N to be several ten thousands. In text IR (information retrieval) community, words appearing in documents comprise a feature vector[2]. So D is usually ten or hundred thousands. In such situations, conventional algorithms may be impractical due to computational demand.

It is important to note that evaluating the fitness of solutions is the operation that dominates the time requirements of whole ACO procedure. The other operations are negligible. To avoid the cost incurred by the fitness evaluation, some methods evaluate individual features and choose the top-ranked features[2][3][7], but this scheme does not perform well since it ignores correlations between features. A method which reduces the computational load without sacrificing quality of the solutions should be developed.

In ACO, pheromone information provides clues to actual fitness values. Using this information, we could approximate the actual fitness values of solutions with only a small amount of computation. Recently, based on this fact, the novel scheme called *selective evaluation* was proposed, which improves the convert gence and recognition performance of ACO[6]. The scheme reduces computational load by excluding unnecessary or less promising candidate solutions from the actual evaluation. Note that our scheme can be used with ACO because ACO retains valuable pheromone trail information that helps identify less promising solutions.

The superiority of the scheme was supported by experimental results. However the experiment seems to be not statistically sufficient since it used only one dataset and no convergence data was presented. The aim of this paper is to analyze convergence characteristics of the selective evaluation scheme and make the conclusion more convincing.

We chose three datasets related to handwriting, medical, and speech domains from UCI repository whose feature set size ranges from 256 to 617[11]. In order to obtain statistically stable data, we executed 12 independent runs for each of three datasets. Each run was given 72 hours to observe the long-time convergence. To cope with highly demanding computational load, we used the grid machine contructed using Beowulf clustering software. The machine has 15 nodes, each of which is equipped with dual core CPUs.

In the analysis, we presented graphs which visually comprare the convergence trends of the selective evaluation scheme and the conventional scheme. Based on the analysis of convergence data, we describe a reason for the superiority of selective evaluation scheme and where the scheme can be applied.

Section 2 explains the conventional ACO for feature selection. Section 3 describes the selective evaluation scheme. Section 4 presents experimental setup and convergence analysis results. Also it discusses why the selective evaluation is advantageous. Section 5 concludes the paper.

II. ACO for Feature Selection

Feature selection involves the selection of a subset of d features from a total of *D* features, based on a given optimization criterion. The set of *D* features are denoted by $\{x_1, x_2, \cdots, x_D\}$. X denotes the subset of selected features, and Y denotes the set of remaining features. J(X) denotes a function evaluating the performance of X. The choice of J() depends on the particular application. In our experiments, the recognition accuracy of multi-layer perceptron(MLP) was used for the values of J().

ACO uses a graph to represent the state of the solution space, where a node corresponds to a feature. The pheromone trail τ_{ij}, $1 \le i,j \le D$, is stored on the edge between nodes x_i and x_j. The outline of ACO is described by the algorithm Nonselective-ACO.

```
Algorithm Nonselective-ACO:
1. Initialize pheromone trail on every edge,
     $\tau_{ij}=\tau_{max}$, $1 \le i,j \le D$;
2. repeat {
3.   for(t = 1 to M) construct solution (subset) $X_t$
       and evaluate it;
4.   Choose the best solution, $X_{best}$, from $X_t$, $1 \le t \le M$;
5.   for(every edge between nodes in $X_{best}$)
       update pheromone trail;
6. } until (stopping-condition);
```

We use the min-max version of ACO, where τ_{min} and τ_{max} are given as user-defined parameters[9]. The value of τ_{ij} is contained with in the range $[\tau_{min}, \tau_{max}]$. In the initialization stage, all edges are set to the maximum value, τ_{max}. In the algorithm, M is the number of ants. In line 3, the t-th ant walks on the graph and generates a candidate solution (feature subset) X_t. Initially X_t is empty. The ant randomly chooses a node (feature) and adds it to X_t. Then the ant successively chooses the next node $x_k \in Y$ with the probability $p(x_k, X_t)$ of Equation (1) and adds it to X_t. The equation includes two factors, the pheromone factor τ and the heuristic factor η. Parameters α and β are weighting coefficients.

$$p(x_k, X_t) = \frac{\tau(x_k, X_t)^{\alpha} \eta(x_k, X_t)^{\beta}}{\sum_{x_j \in Y} \tau(x_j, X_t)^{\alpha} \eta(x_j, X_t)^{\beta}} \qquad (1)$$

The $\tau(x_k, X_t)$ value is computed based on the current pheromone in formation deposited on the graph. This is defined by Equation (2). [Figure 1] illustrates an example of computation of (2) where D is 5 and the subset of selected features is $X_t=\{x_1, x_4\}$.

$$\tau(x_k, X_t) = \sum_{x_i \in X_t} \tau_{ki} \qquad (2)$$

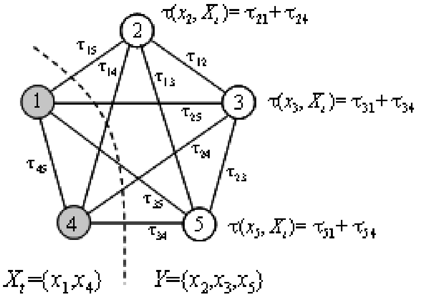

Figure 1. Computation of (2) for the situation where Xt=$\{x_1, x_4\}$

The heuristic factor $\eta(x_k, X_t)$ should be designed using a heuristic rule specific to the feature selection problem. This paper uses the null η scheme which has been proposed in [10]. It simply ignores the η factor by letting α and β to be 1 and 0, respectively. The scheme is viable since the other factor τ considers sufficiently the correlations between features.

We use elitism in which only the best solutions, called "elite" and denoted in (4) by X_{best}, of the candidate solutions are given the chance of updating the pheromone in line 5[9]. Every edges connecting a pair of nodes in the elite solution, X_{best} is given the chance of updating their pheromones. For example, when X_{best} is x_2-x_4-x_1, the τ_{24}, τ_{21}, and τ_{41} are updated. Equation (3) updates the pheromone trail for the edge connecting node x_i and x_j, where ρ is the pheromone evaporation parameter. By the min() and max() functions, the pheromone trail is kept within [τ_{min}, τ_{max}]. In Equation(4), S_{best} is the value of $J(X_{best})$, and $S_{best\_so\_far}$ is the value of the best solution found so far from the first generation.

$$\tau_{ij} = \min(\max((1-\rho)\tau_{ij} + \delta(x_i, x_j, X_{best}), \tau_{min}), \tau_{max}) \quad (3)$$

$$\delta(x_i, x_j, X_{best}) = \begin{cases} \dfrac{1}{1 + S_{best\_so\_far} - S_{best}}, & \text{if } x_i \in X_{best} \text{ and } x_j \in X_{best} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Let us analyze the computational time of the algorithm. The dominant operation is the subset evaluation that occurs in line 3. The other operations are negligible. One generation of the loop performs M subset evaluations, where M is the number of ants. So the time complexity of Nonselective-ACO is O(MT) where T is number of generations. In this situation, if we can reduce the number of subset evaluations in a reasonable way, convergence speed can be improved.

III. ACO with selective evaluation

The aim of this section is to devise an improved version of the algorithm Nonselective-ACO. Here we describe the *selective evaluation* scheme[6]. The formal description is given by the algorithm Selective-ACO. It actually evaluates only the solutions that are likely to be elite. In ACO, the pheromone trail provides valuable information that enables implementation of selective evaluation. We pre-evaluate candidate solutions and select a proportion of the top-ranked solutions for actual evaluation. The pre-evaluation is simply accomplished by adding pheromone values on the edges connecting adjacent nodes on a candidate solution. For example, when a solution is x_2-x_4-x_1, the score obtained by the pre-evaluation is $\tau_{24}+\tau_{41}$.

In early generations, the pheromone trail is unreliable. As generations proceed, the pheromone trail becomes more reliable. So starting with a high ratio of actual evaluation, we reduce the ratio gradually as time goes. Equation (5) determines the ratio for generation h. The parameter r is a decreasing factor. The parameter *lower_bound* keeps the ratio above its value.

$$q(h) = \max(r^h, lower\_bound) \quad (5)$$

```
Algorithm Selective-ACO:
1. Initialize pheromone trail on every edge;
2. $h$ = 0; // generation
3. repeat {
4.   for(t = 1 to M) construct solution $X_t$
5.   Pre-evaluate $X_t$, $1 \le t \le M$ using pheromone
     information;
6.   Sort $X_t$, $1 \le t \le M$
7.  for (t = 1 to M) if ($X_t$'s rank $\le$ M*q(h))
        evaluate $X_t$
8.  Choose the best solution, $X_{best}$, from solutions
    whose evaluation were actually performed in
    line 7;
9.  for (every edge between nodes in $X_{best}$)
        update pheromone trail;
10. h++;
11. } until (stopping-condition);
```

In the algorithm Selective_ACO, lines 5-6 perform pre-evaluation. Since M is on the order of tens, sorting is not expensive. Line 7 performs actual evaluation of solutions which passed the pre-evaluation.

IV. Convergence characteristics and discussions

1. Experimental setup and results

All experiments were performed using the CENPARMI handwritten numeral database and two datasets from the UCI repository[11]. The chosen datasets were large in D (number of features) and/or N (number of samples). They are described in [Table 1]. MLP was used for subset evaluation. Its architecture was $(1+d)$ - $2K$ - K where $d = |X|$ and K denotes the number of classes. The MLPs were trained using an error back-propagation algorithm and tested using the test set[12]. For Arrhythmia, cross-validation was used since separate training and test sets were not available. Test accuracy was used to evaluate the fitness of solutions. Accuracy was calculated as a ratio of the number of correctly recognized samples to the total number of samples. No rejection was allowed. For every dataset, we used $d=D/4$. So for example, d=64 features were selected from D=256 features for Numerals.

The ACOs used the following parameters.

Nonselective-ACO: population size = 30, α = 1,
β = 0, ρ (evaporation rate) = 0.06,
$[\tau_{min}, \tau_{max}]$ = [0.2,20.0]
Selective-ACO: the same as above, r = 0.98,
lower-bound = 0.3

To observe the convergence characteristics of the ACO, we allowed a long operating time, 72 hours for the *stopping-condition* of the algorithms. So totally the computation time is 3 datasets*12 runs*72 hours*2 schemes=5184 hours(216 days). To cope with this high computational load, we used the grid machine with 15 nodes, each of which is equipped with dual core CPUs.

[Table 2] compares the recognition accuracies for Nonselective-ACO and Selective-ACO. For an objective comparison, we ran ACO 12 times independently, and the average, minimum, and maximum were recorded in [Table 2]. [Table 2] illustrates that selective ACO produced better solutions than nonselective ACO in terms of both the average and maximum accuracies. As an example, for Isolet, selective ACO increased the average and maximum accuracies by 0.08% and 0.13%, respectively. Though the gap is small, we believe that it is meaningful because the data was obtained from 12 independent runs. For Arrhythmia, the selective ACO increased the average and maximum accuracies by 0.57% and 0.66%, respectively.

For a finer analysis of the algorithm behaviors, [Figure 2][Figure 3][Figure 4] show the convergence characteristics of the three datasets. Note that time scale of the x-axis is not linear. Since solutions improved more dynamically during the early stage, we set the time scale of the early and late stages as fine and coarse, respectively. Let us pay attention to average curves. For Numerals, the selective ACO continued to lead nonselective ACO after 10 hours by about 0.2%. For Arrhythmia and Isolet, the average trend was similar.

Table 1. Dataset used in experiments

Datasets	Area	Number of features (D, d)	Number of classes	Number of samples (train, test)
Numerals	Hand writing	(256, 64)	10	(4000, 2000)
Arrhythmia	Medical	(279, 70)	16	452*
Isolet	Speech	(617, 154)	26	(6238, 1559)

* Arrhythmia has no separate test set, so cross-validation was used.

Table 2. Comparison of accuracies (%)

Datasets	nonselective ACO*	selective ACO*
Numerals	97.30 (97.20, 97.40)	97.48 (97.35, 97.75)
Arrhythmia	70.19 (69.91, 71.02)	70.76 (70.13, 71.68)
Isolet	94.99 (94.80, 95.25)	95.07 (94.80, 95.38)

* Average (minimum, maximum) of 12 independent runs

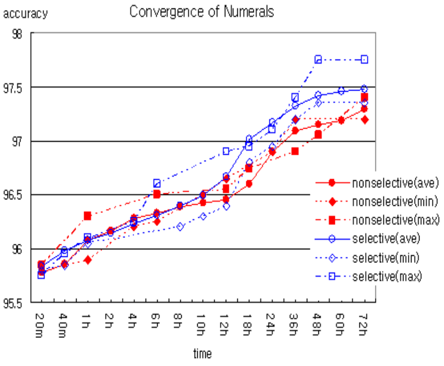

Figure 2. Convergence of Numerals

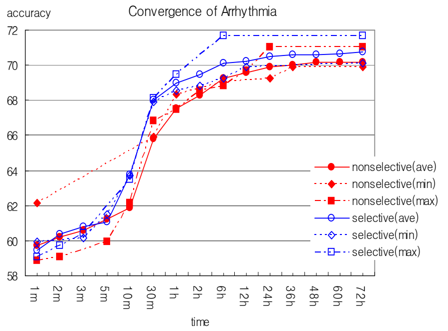

Figure 3. Convergence of Arrhythmia

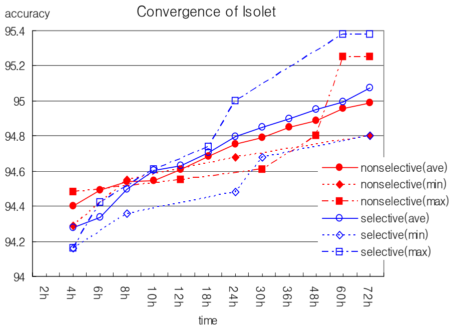

Figure 4. Convergence of Isolet

2. Discussions

Why is the selective evaluation scheme advantageous? The scheme can use more generations than the non-selective version under the same timing budget because it evaluates less solutions per generation. This gives the scheme more chance to find out better solutions. As another reason, we argue that the selective evaluation has a stronger *exploration* force since it sometimes happens to miss the true best solutions in the pre-evaluation stage. In that case, the chosen solution which is not elite may perturb the graph state t o explore the search space more broadly. Putting these two arguments in one statement, we conjecture that the selective evaluation scheme performs better by searching the space more broadly for a longertime.

Is the selective evaluation scheme applicable to other problems rather than the feature selection? The scheme is generic since it can be embedded in any ACO procedure. In terms of efficacy, It is useful only for the problems which the fitness evaluation of solutions is demanding. Typical example is the feature selection which this paper uses as test-bed problem. Other examples include prototype selection problem which tries to find out the optimal subsets of prototypes for the k-nearest neighbor classifier, and neural network optimization which attempts to find out the optimal network architecture. On the contrary, traveling salesperson problem benefits nothing from the selective evaluation scheme because fitness evaluation is a computation of path length which is a trivial calculation.

V. Conclusions

The aim of this paper was to analyze convergence characteristics of the selective evaluation scheme and make the conclusion more convincing. Experiments using datasets from diverse application areas and statistically stable observations was designed and performed. The analysis made the conclusions more convincing. The scheme will be useful for the problems and situations where the time budget is limited or ACO is impractical due to a large problem size.

We have presented an intuitive explanation why the scheme is advantageous. In the future, it would be worthwhile developing a mathematically rigorous theory which explains the reason.

References

[1] J. Kittler, "Feature selection and extraction," in *Handbook of Pattern Recognition and Image Processing*, Academic Press (Edited by T.Y .Young and K.S.Fu), pp.59-83, 1986.

[2] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," Expert Systems with Applica ti ons, Vol.36, pp.6843-6853, 2009.

[3] A. Al-Ani, "Feature subset selection using ant colony optimization," International Journal of Computational Intelligence, Vol.2, No.1, pp.53-58, 2006.

[4] M. E. Basiri and S. Nemati, "A novel hybrid ACO-GA algorithm for text feature selection," IEEE Congresson Evolutionary Computation, pp.2561-2568, 2009.

[5] K. J. Lee, J. Joo, J. Yang, and V. Honavar, "Experimental comparison of feature subset selection using GA and ACO algorithm," Lecture Notes in Computer Science (Advanced Data Mining and Applications), Vol.4093, pp.465-472, 2006.

[6] 오일석, 이진선, "패턴 인식에서 특징 선택을 위한 개미 군락 최적화," 한국콘텐츠학회논문지, 제10권, 제5호, pp.1-9, 2010.

[7] S. M. Vieira, J. M. C. Sousa, and T. A. Runkler, "Fuzzy classification in ant feature selection," IEEE International Conference on Fuzzy Syst ems, pp.1763-1769, 2008.

[8] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Computational Int elligence Magazine, Vol.1, No.4, pp.28-39, 2006.

[9] C. Solnon and D. Bridge, "An ant colony optimization meta-heuristic for subset selection problems," in *System Engineering using Parti cle Swarm Optimization* (Edited by Nadia Nedjah and Luiza Mourelle), Nova Science publisher, pp.7-29, 2006.

[10] I. S. Oh and J. S. Lee, "Ant colony optimization with null heuristic factor for feature selection," Proceedings of IEEE TENCON, 2009.

[11] A. Asuncion and D. J. Newman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2007.

[12] 오일석, *패턴인식*, 교보문고, 2008.

저자소개

이 진 선(Jin-Seon Lee) 정회원

- 1985년 : 전북대학교 전산통계학과(이학사)
- 1995년 : 전북대학교 전자계산기공학과 박사
- 1995년 3월 ~ 현재 : 우석대학교 게임콘텐츠학과 교수

<관심분야> : 멀티미디어, 패턴인식

오 일 석(Il-Seok Oh) 정회원

- 1984년 : 서울대학교 컴퓨터공학과(공학사)
- 1992년 : KAIST 전산학과 박사
- 1992년 9월 ~ 현재 : 전북대학교 컴퓨터공학부 교수

<관심분야> : 컴퓨터비젼, 패턴인식