Journal of the Society of Korea Industrial and Systems Engineering
Vol, 29, No. 1, pp.76－86, March 2006.

역 물류 환경 인터넷 경매를 위한 요소 선택응용 추천 시스템

양재경* · 유우연**

*전북대학교 산업정보시스템공학과
**명지대학교 산업시스템공학부

Feature Selection Applied to Recommender Systems for Reverse Logistics Internet Auction

Jae-Kyung Yang* · Woo-Yeon Yu**

*Department of Industrial and Information Systems Engineering, Chonbuk National University
**Department of Industrial and Systems Engineering, Myong Ji University

다양한 데이터 마이닝 기법들의 발전과 더불어, 속성 (Feature 또는 Attribute)의 범위 (Dimension)를 줄이기 위해 많은 요소 선택 방법이 개발되었다. 이는 확장성 (Scalability)을 향상시킬 수 있고 학습 모델 (Learning Model)을 더욱 쉽게 해석할 수 있도록 한다. 이 논문에서는 네스티드 분할 (Nested Partition, 이하 NP)을 이용한 새로운 최적화 기반 속성 선택 방법을 NP 기본 구조와 다양한 실험 문제의 수치적 결과들과 함께 제시하여 어떻게 NP의 최적화 구조가 속성 선택 과정에 기여를 하고 있는지 보여준다. 그리고 이 새로운 지능적인 분할 방법이 어떻게 매우 효율적인 분할을 수행하는지를 제시한다. 이 새로운 속성 선택 방법은 필터 (Filter)방법과 래퍼 (Wrapper)방법 두 가지로 구현될 수 있다. 사례 연구로서, B2B e-비즈니스 시스템에서 효과적으로 사용될 수 있는 추천 시스템 (Recommender System)을 제안하였다. 이 추천 시스템은 분류 기법 (Classification Rule)과 제시된 NP 기반 요소 선택 방법을 사용하고 있다. 이 추천 시스템은 사용자의 인터넷 경매 참여를 추천하는데 사용되며, 이 때 제안된 요소 선택 앨고리듬은 추천 규칙들이 쉽게 이해될 수 있도록 모델을 간략화 하는데 사용된다.

Keywords : Feature Selection, Partition, Optimization, Recommendation, Internet Auction

1. Introduction

Feature selection is an important data mining problem for numerous reasons [5]. It can be used to eliminate redundant and irrelevant features from a data set, resulting in a dimensionality reduction that reduces the learning time needed for induction algorithms that are applied to the data set, and in many cases also results in better (that is, more accurate) predictive models. Careful feature selection can improve the scalability of a data mining system as the induction is usually much faster with fewer features.

The feature selection problem involves selecting a best subset of features from a finite subset and is therefore a discrete optimization problem. As such, any number of well known optimization approaches can be applied to this problem and previous work has for example used mathematical programming [2], branch-and-bound [6], genetic algorithms [11], and evolutionary search [4].

In this paper, we propose the new feature selection methodology, which applies an optimization method called

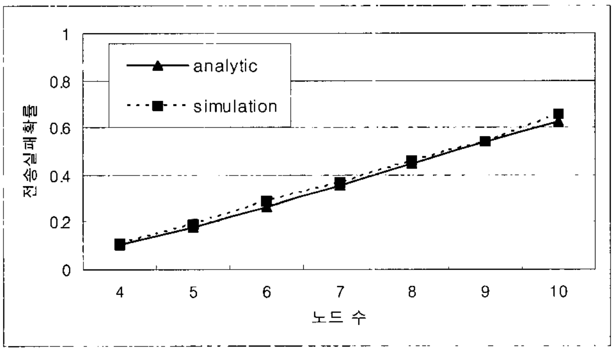
* This paper was supported by research funds of Chonbuk National University in 2005

4.4 전송경로 상 노드 수 변화에 따른 전송실패확률

전송 경로 상에 있는 노드 수 변화에 따른 전송실패확률 변화를 살펴보기 위해 다음과 같은 가정으로 실험하였다.

- 주파수 영역 반경(R) : 100m
- 노드들의 이동 속도는 동일 :
  $E[V_{i,i+1}]$= 2미터/분 $(i = 1,2,3,4)$
- 노드 간 데이터 전송 속도 :
  $E[T_{i,i+1(trans)}]$=60초 $(i = 1,2,3,4)$

<그림 8>은 전송 경로 상에 있는 노드의 수가 증가함에 따라 전송실패확률이 증가하는 것을 보여주고 있다. 노드의 수가 증가한다는 것은 데이터 전송시간의 증가와 중간 노드들의 이탈율 증가를 의미한다. 데이터 전송시간은 증가하고 노드들의 이탈할 경향이 커짐으로써 전송실패확률이 점차 증가하게 된다.



<그림 8>노드 수 변화에 따른 전송실패확률

5. 결 론

본 연구에서는 MANET 환경에서 멀티캐스팅에 의한 이동 노드들의 이동성과 성능분석을 위한 전송실패확률 계산식을 설명하였고, 본 연구에서 제시한 방법에 대한 타당성 검증을 위해 모의실험과 비교 검토하였다. 실험을 통해서 노드들의 이동 속도가 크거나 데이터 전송시간이 큰 경우, 또한 소스 노드에서 목적지 노드까지 전송 경로 상에 노드의 수가 많을수록 데이터 전송실패확률이 높아진다는 것을 알 수 있었다. 그리고 라디오 주파수 영역의 크기는 전송실패확률에 대해 노드의 이동 속도와는 상대적인 영향을 주는 것을 알 수 있었다.

본 연구는 각 노드의 체재시간과 노드 간 데이터 전송시간에 대한 분포를 지수분포로 가정한 것이 현실 모형과 차이가 있을 수 있겠으나, MANET 환경의 특징인 이동성을 고려하여 멀티 홉으로 구성된 다중 경로 상에서 노드의 이동성과 전송실패확률과 같은 시스템 성능척도를 구하기 위해 수리적인 방법으로 접근하였다는데 의의가 있다.

향후 연구과제로는 체재시간 및 전송시간에 대하여 지수분포가 아닌 현실 모형에 근사한 분포를 통해 접근하는 수리적인 방법 개발과 더불어 MAC을 고려한 처리량, 지연시간 등 다양한 성능척도에 대해 연구할 수 있을 것이다.

참고문헌

[1] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communication and Mobile Computing(WCMC): Special issue on Mobile Ad Hoc Networking : Research Trends and Applications*, 2(5), pp.483-502, 2002.

[2] I. Chlamtac, M. Conti, and J. J-N. Liu, "Mobile ad hoc networking : imperatives and challenges", *Ad Hoc Networks 1*, pp.13-64, 2003.

[3] Y. Fang, "Hyper-Erllang Distribution and its Applications in Wireless and Mobile Networks", *Wireless Networks (WINET)*, 7(3), pp.211-219, May. 2001.

[4] A. Jardosh, E.B. Royer, K. Almeroth, and S. Suri, "Towards Realistic Mobility Models for Mobile Ad hoc Networks", *Mobicom 2003*, San Deigo, pp.14-19, Sep. 2003.

[5] X. Li, D.P. Agrawal, and Qing-An Zeng, "Impact of Mobility on the Performance of Mobile Ad Hoc Networks", *IEEE Wireless Telecommunications Symposium 2004*, pp.154-160, May. 2004.

[6] A.B. McDonald and T. Znati, "A Path Availability Model for Wireless Ad Hoc Networks", *WCNC99*, Vol.1, pp.35-40, 1999.

[7] J. Tian, J. Hhner, and C. Becker, "Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation", *Proceedings of the 35th Annual Simulation Symposium, in cooperation with the IEEE Computer Society and ACM*, San Diego, California, Apr. 2002.

[8] H. Xie, S. Kuek, "Priority handoff analysis", *Vehicular Technology Conference, 1993 IEEE 43rd* ,pp.855-858, May. 1993.

[9] M. Xonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns", *IEEE Jeournal on Selected Areas in Communications*, 15(7), pp.1239- 1252, Sep. 1997.

[10] Q-A. Zeng and D.P. Agrawal, "Modeling and Efficient Handing of Handoffs in Integrated Wireless Mobile Networks", *IEEE Transaction on Vehicular Technology*, 51(6), pp.1469-1478, 2002.

iteration. A sufficient condition for asymptotic convergence is that this probability of correct selection is bigger than one half, and to guarantee that a minimum probability is obtained, and we can use a two-stage sampling procedure that determines how much random sampling effort, $N(\psi,\delta)$ [7], is needed from each region to guarantee correct selection with probability ψ within an indifference zone $\delta > 0$. The two-stage sampling also allows us to further analyze the convergence of the algorithm and develop statements concerning the quality of the solution once maximum depth is reached. In particular, an expression can be derived for the probability of having found sufficiently good solution the first time maximum depth is reached:

$$\Pr\{|\ f(\mathsf{A}\ (k)) - f^*\ | \le \delta\} \ge \Psi, \qquad (2)$$

where $\delta > 0$ is an indifference zone, that is a performance value difference that is considered insignificant, and

$$\Psi = \frac{\psi^n}{(1-\psi)^n + \psi^n}, \qquad (3)$$

where ψ is the user selected minimum probability by which a correct selection is made in each iteration, and n is as before the total number of features. Sometimes it may be beneficial to stop the algorithm early, that is, we can specify a stopping depth $d_{stop}(n) \le n$, define the objective function on sets of feature subsets as

$$f(\mathsf{A}(k)) = \max_{a \in \mathsf{A}(k)} f(a), \qquad (4)$$

and equation (3) holds with Ψ replaced with

$$\Psi' = \frac{\psi^{d_{stop}(n)}}{(1-\psi)^{d_{stop}(n)} + \psi^{d_{stop}(n)}}. \qquad (5)$$

Partitioning for the feature selection problem reduces to determine an order for the features and then the subregions correspond to either including a feature or not including a feature. Thus, assuming that the current most promising region is some subset $A(k) \subset A$ of the entire feasible region, then this subset is partitioned by fixing the next feature a in the order, that is, the subsets are

$$A_1(k) = \{A \in A(k) : a \in A\} \qquad (6)$$

$$A_2(k) = \{A \in A(k) : a \notin A\} \qquad (7)$$

The surrounding region is simply $A_3(k) = A \backslash A(k)$. Each of these three regions is then sampled as discussed above and based on these samples the next most promising region is selected. In theory, the features can be selected in an arbitrary order, but an intelligent partitioning where features are ordered according to their information gain performs significantly better, and this partitioning is used in all of the numerical experiments below.

Given $d_{stop}(n)$, δ, Ψ and an order $a_{[1]}$, $a_{[2]}$, , $a_{[n]}$ of features
Initialize A(0) $\leftarrow$ A, k $\leftarrow$ 0, $A^* = \{\}$ and $f^* = \infty$
loop
  $A_1(k) \leftarrow \{A \in \mathrm{A}(k): a_{d(k)} \in A\}$,
  $A_2(k) \leftarrow \{A \in \mathrm{A}(k): a_{d(k)} \notin A\}$,
  $A_3(k) \leftarrow A \setminus \mathrm{A}(k)$,
  for every set $A_j(k)$
    $A^j_{best}(k) \leftarrow \{\}$, $f^j_{best}(k) \leftarrow \infty$, $i \leftarrow 1$
    loop
      $A_{ji}(k) \leftarrow$ Randomly select a feature subset
      if $f_{ji}(k) < f^j_{best}(k)$ then
        $f^j_{best}(k) \leftarrow f_{ji}(k)$, $A^j_{best}(k) \leftarrow A_{ji}(k)$,
        $i \leftarrow i+1$
    until enough feature subset samples given
    δ and Ψ
    if = 3 then A(k + 1) $\leftarrow$ A(k - 1)
    else A(k + 1) $\leftarrow$ $A_{j^*}(k)$
  end
until $d(\mathrm{A}(k)) = d_{stop}(n)$

<Figure 2> NP feature selection pseudocode

A complete description of the NPF is shown in Figure 2. Note that it uses a fixed number of n_0 samples to evaluate each region, starts with the set A of all possible feature subsets as the most promising region, and terminates when the depth of the most promising region has reached maximum, that is, it is a singleton. We also let A^* be the best feature subset found and f^* be the corresponding performance value, which is calculated according to equation (1) above.

3. Evaluation of the NP-Wrapper and NP-Filter

In this section we present numerical results for tests of the NPW and NPF when used to precede the Naïve Bayes algorithm. The code is written in Java using the *Weka* machine learning software library [10] for implementation of the learning algorithms themselves. We used five data sets from the *UCI Repository of machine learning databases* [1]. The characteristics of these data sets are shown in Table 1, from which we note that the sizes range from 148 to 3196 instances and from 9 to 69 features. As both the NPF and NPW are randomized algorithms, we run five replications for each experiment and report both the average and the standard error.

<Table 1> Characteristics of the tested data

Data Set	Instances	Features
lymph	148	18
vote	435	16
audiology	226	69
cancer	286	9
kr-vs-kp	3196	36

3.1 Value of Feature Selection

Our first set of experiments addresses the effectiveness of feature selection using the NPF and NPW for the selected data sets. As noted before, the Naïve Bayes is used to induce classification models with the selected features. We measure the effectiveness along two dimensions. First we consider the accuracy of the models induced after feature selection compared to the corresponding models without feature selection, and second we consider how many features are eliminated, that is, how much smaller the models become when feature selection is employed.

<Table 2> Accuracy of Naive Bayes with and without feature selection.

Data Set	NFS		NPF		NPW	
	Accu.	Size	Accuracy	Size	Accuracy	Size
lymph	85.1	18	85.4±1.0	10.6±2.1	86.2±0.8	9.2±0.8
vote	90.1	16	93.2±1.0	6.8±1.1	95.8±0.4	3.0±1.4
audiology	71.2	69	71.2±1.5	27.4±3.2	75.0±2.3	23.0±3.9
cancer	73.4	9	73.8±0.4	5.8±0.8	75.7±0.2	3.6±0.9
kr-vs-kp	88.0	36	90.8±2.1	11.6±1.5	94.4±0.3	14.2±3.8

The results for the Naïve Bayes classification method are shown in Table 2. Columns 2-3 show the results for no feature selection (NFS), columns 4-5 show results for the NPF and finally columns 6-7 for the NPW. First looking at the accuracy we note that it actually improves or is no worse when we use feature selection, and the models where classification is preceded by a NPW have the highest accuracy. Indeed, there is an average of 1.5% improvement in accuracy when we precede a Naïve Bayes method with NPF, and 4.7% when it is preceded with a NPW. Such improvement in accuracy may or may not occur when feature selection is employed. In particular, the performance of Naïve Bayes is known to be degraded by redundant features and it appears that those are effectively eliminated by both feature selection algorithms. What we do, however, always expect from a feature selection procedure is a significant reduction in the number of features, resulting in simpler and easier to explain models. Table 2 demonstrates this reduction. For example, when the NPF is used, the 69 features of the 'audiology' data are reduced to an average of 27.4 features, and when the NPW is used, they are reduced to an average 23.0 features. This is a significant simplification of the models. Across all the data set there is an average 52% reduction in number of features when we use the NPF and an average of 64% fewer features when we use the NPW. We note that the NPW performs better on both the accuracy and simplicity measures.

3.2 Importance of Intelligent Partitioning

We can ask how much of these previous results is due to the generic NP framework itself and how much can be contributed to the intelligent partitioning scheme developed in this paper. To address this, we compare NP algorithms using the intelligent partitioning to NP algorithms using all other possible ways of partitioning. Since a partition is defined by the order in which features are either included or not, this implies considering all possible orders of the features. In particular, we use a complete enumeration of all partitions to find the best and worst one, and compare those to the intelligent partitioning. We first draw a sample of 7 features and then apply the NP algorithms. Note that we still have to evaluate 5040 different partitions. To assure the sampling does not introduce a bias we repeat the process five times, each time drawing an independent sample of 7 features. For those experiments we restrict ourselves to the 'vote' and 'cancer' data sets and we only consider the NPW with Naïve Bayes classification. Results for other configurations are similar and are omitted here for brevity.

<Table 3> Accuracy of intelligent partitioning in NPW using Naive Bayes.

Data Set		Accuracy		
		Intelligent	Best	Worst
vote	1	90.8±0.3	91.0	86.4
	2	95.9±0.0	95.9	94.3
	3	89.0±0.0	89.0	87.4
	4	90.0±0.3	90.1	85.1
	5	95.6±0.0	95.6	92.0
cancer	1	75.9±0.0	75.9	72.7
	2	75.7±0.0	75.9	72.7
	3	75.8±0.2	75.9	73.1
	4	72.8±0.3	73.1	70.6
	5	75.9±0.0	75.9	72.0

<Table 4> Speed of intelligent partitioning in NPW using Naive Bayes.

Data Set		Computation Time		
		Intelligent	Slowest	Fastest
vote	1	3812	20370	2814
	2	3515	19408	2153
	3	3371	35080	2784
	4	3690	18046	2774
	5	3433	17375	3094
cancer	1	2740	14050	1382
	2	2624	16013	1372
	3	2664	20390	1342
	4	4969	25226	1362
	5	2642	6920	1372

In Table 3 the prediction accuracy of the models using intelligent partition, and the best and worst partition found using enumeration are reported. We note that the accuracy found using the intelligent partition is very close to the optimal. In particular, for half of the problems the intelligent partitioning always results in the same accuracy as the optimal partition, and for the other half the performance is within one standard deviation. On the other hand, we note that partitioning poorly results in feature subsets that have significantly lower accuracy but even for the worst possible partition the NP method is still able to obtain fairly high quality subsets.

We also compare the computational time used by the NPW if different partitioning schemes are used, These results are reported in Table 4 and we see that again using the intelligent partitioning results in performance that is fairly close to the optimal, although this time there is more different than with respect to accuracy. Thus, we can conclude that the NPF is capable of compensating fairly well for poor partitions in terms of obtaining accurate models, but this occurs at the expense of using very long computation time. The intuitive reason for this is that any NP algorithm can compensate for mistakes, that moves in the wrong direction, by backtracking, but frequent backtracking is time consuming and will slow the search significantly. We conclude that a good partition is important with respect to both obtaining high accuracy models and in the time it takes to find the appropriate feature subsets, and of the two the latter is by far the most significant.

3.4 Comparison with GA Feature Selection

Genetic algorithms (GA) are similar to the NP method in that they use a randomized search strategy to explore the set of alternatives, in this case all possible subsets of features. They have also been shown to perform well for the feature selection [11]. Genetic algorithms thus provide a useful benchmark for comparing the performance of the new methodology.

These GA comparisons use Naive Bayes as the classifier. The maximum depth for the NPF is taken to be the minimum depth, and as before 5 replications are run for each algorithm. The NP-based algorithms and the GA algorithms are allowed to run for the same amount of time and the GA parameters were selected for best overall performance. Thus, NP and GA are compared only in terms of accuracy and the size of the selected subsets.

The first experiments compare the NPF with GA search that uses the same evaluation criterion, that is, a corresponding GA-Filter (GAF). The results are reported in Table 5. The accuracy of the sets obtained by the two algorithms appears to be quite similar. Although the average accuracy obtained by NPF is strictly better for all five of the test sets, the difference is only statistically significant for the 'vote' data. The difference in performance may be explained by the fact that the NPF tends to select slightly larger feature subsets for all but one of the data sets.

<Table 5> Comparison of NPF and GAF.

Data Set	NPF		GAF	
	Accuracy	Size	Accuracy	Size
lymph	85.7±0.7	11.8±1.6	84.3±1.3	9.8±1.1
vote	99.0±1.4	5.4±2.5	94.3±1.3	3.8±1.3
audiology	70.5±3.1	11.6±1.8	69.6±0.9	9.0±2.3
cancer	73.7±0.4	4.6±0.6	73.6±0.2	5.6±1.1
kr-vs-kp	90.7±1.3	7.6±0.9	89.8±1.4	4.8±1.3

A similar comparison with the NPW and a corresponding GA-Wrapper (GAW) is reported in Table 11, and the results are similar to the filter results. The NPW has higher accuracy for three out of the five data sets, the GAW is better for one, and the two are tied for the 'cancer' data set. However, none of these differences are statistically significant.

<Table 6> Comparison of NPW and GAW.

Data Set	NPW		GA Filter	
	Accuracy	Size	Accuracy	Size
lymph	86.0±1.8	9.4±0.9	84.9±1.6	12.0±0.7
vote	94.3±0.5	5.8±0.5	95.1±0.8	5.6±1.5
audiology	73.5±2.5	14.0±4.3	72.0±1.7	38.0±9.3
cancer	74.3±0.9	4.2±1.6	74.3±0.6	5.0±1.4
kr-vs-kp	94.2±0.1	7.4±1.7	92.4±0.7	19.3±3.1

We conclude that the new methodology is a promising alternative and is certainly competitive to other methods such as GA that produce high quality feature subsets. However, we do not expect the NP-based methods to outperform GA for every data set.

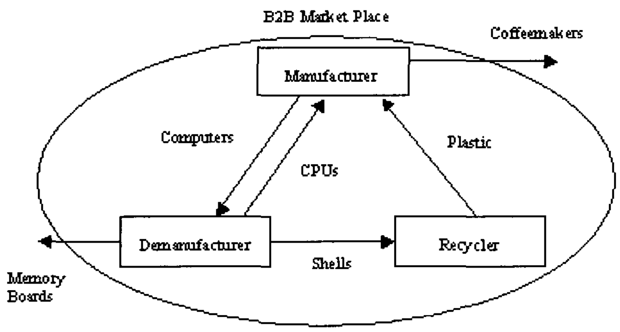
4. Case Study-Recommender Systems

In e-commerce applications, recommender systems intelligently suggest products to users as well as provide information that helps users make a decision which products to be of interest, based on rules or knowledge extracted from data storing observed user behavior and experience profiles. The success of such recommender systems depends on how good quality of recommendations can be made to users. It is very important to recommend the correct products users since such a recommendation lead to have users positively respond.

In this section, the recommender system for an Internet auction system [8] to facilitate reverse logistics is provided using classification rules. The Internet auction system is intended to bring together a fragmented market of manufacturers, recyclers, demanufacturers and others interested in the take-back and reuse of obsolete recyclable products. To support such an auction system, the recommender system is proposed that among other functions recommends to a user if they should participate in a particular auction. However, it is equally important that a user understands what goes into such recommendation and thus, feature selection is used to build a simple recommendation model that can be easily explained. This is a critical aspect due to the fragmentation of the market, which implies that many potential users are unfamiliar with each other and the range of available products. The new feature selection algorithm applying a nested partition method is used for reducing the number of features.

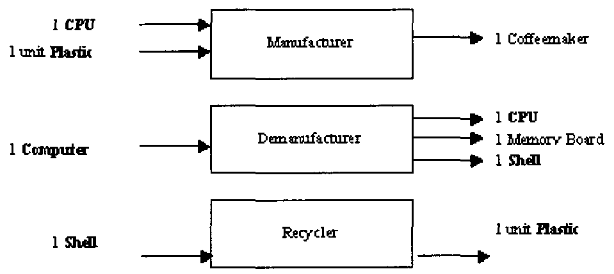
4.1 Recommender System for Reverse Logistics Internet Auction

This auction system simulates the growing industry of electronics recycling. The reverse logistics or returned products from consumers can be regarded as a movement process of various types of products or raw materials back from consumers for any reasons. The Internet auction system facilitates the process of reverse logistics for recycling. Three types of participants, three manufacturers, three demanufacturers, and three recyclers, participate an auction for 6 rounds. In each round, participants can bid price and quantity of items to sell or to buy [8].



<Figure 3> Relationships among participants in the online market place.

We constructed a data set from a database gathered in that prototyped auction system, namely auction participation data set. Based on the data set, we construct recommender systems that consist of classification and feature selection.



<Figure 4> Relationships among products traded online (bold) and sold externally.

As seen in Figure 3 and Figure 4, Each manufacturer sells computers, and produces and sells coffee makers outside the auction, each of which consists of 1 CPU and 1 unit of Plastic that can be purchased from the auction. Demanufacturers buy computers from manufacturers and disassemble into CPUs, memory boards, and shells. Demanufacturers can sell memory boards outside the auction. Recyclers buy shells from demanufacturers and transforms them into plastics. However, the participants can participate in any auction they want.

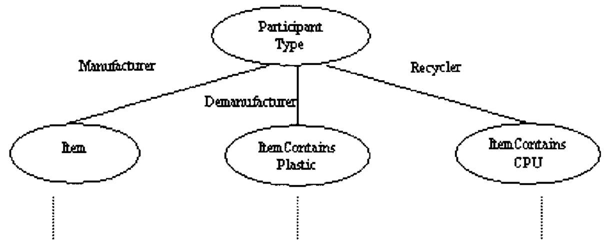
4.2 Recommendation for Auction Participation

In order to figure out which auctions can attract participants, the data set below is constructed by consisting of 31 features and 1 class feature. Those features state behaviors of each participant in each auction round for a specific item as well as characteristics of the auction itself and the class feature in the data is YES/NO depending on if the user participated in a particular auction or not. The detailed description of which is shown in Table 7. The data set contains 2592 instances, each of which states an example of an auction including historical and current data of each participant. Using classification, specifically C4.5, we construct recommender system to generate concrete rules that can figure out which auctions have users interested. We expect to create rules that may catch out some implicit facts, not just obviously simple ones.

<Table 7> Data set for auction participation.

Type of Feature	Feature
Product Description	Item
	ItemContainsCPU
	ItemContainsShell
	ItemContainsPlastic
Round	Round
Information on Last Transaction to Occur	PriceForLastComputerAuction
	QuantityForLastComputerAuction
	PriceForLastCPUAuction
	QuantityForLastCPUAuction
	PriceForLastShellAuction
	QuantityForLastShellAuction
	PriceForLastPlasticAuction
	QuantityForLastPlasticAuction
Information on Last Transaction with Participants Involvement	TimeOfParticipantsLastComputerAuction
	PriceForParticipantsLastComputerAuction
	QuantityForParticipantsLastComputerAuction
	CategoryOfParticipantsLastComputerAuction
	TimeOfParticipantsLastCPUAuction
	PriceForParticipantsLastCPUAuction
	QuantityForParticipantsLastCPUAuction
	CategoryOfParticipantsLastCPUAuction
	TimeOfParticipantsLastShellAuction
	PriceForParticipantsLastShellAuction
	QuantityForParticipantsLastShellAuction
	CategoryOfParticipantsLastShellAuction
	TimeOfParticipantsLastPlasticAuction
	PriceForParticipantsLastPlasticAuction
	QuantityForParticipantsLastPlasticAuction
	CategoryOfParticipantsLastPlasticAuction
Participant	ParticipantName
	ParticipantType
Class	ParticpateInAuction

Figure 5 shows top nodes of the decision tree. The top node, Participant Type, is branched into Manufacturer, Demanufacturer and Recycler.



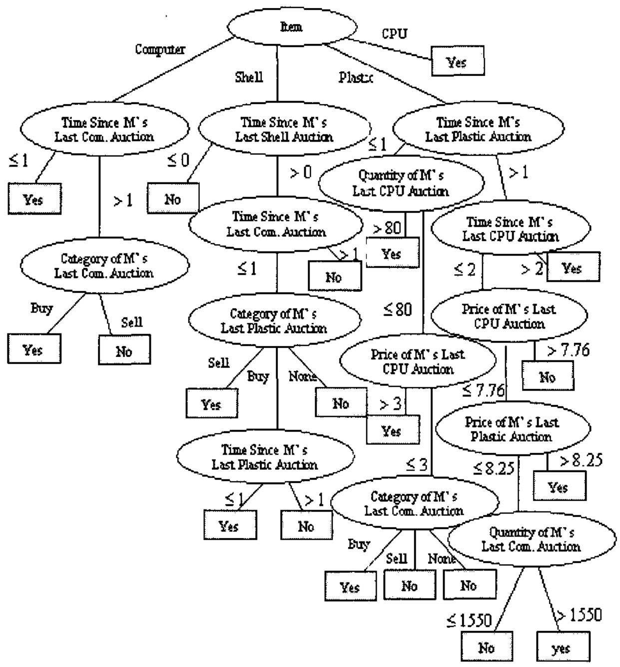
<Figure 5> Root of decision tree.

Since feature selection is not applied to generate this tree, the size is too big to be shown in one simple tree. Next several figures represent the decision tree for auction participation.

First let's consider auction behaviors of manufacturers and make some interesting rules (See Figure 6). Simply we can figure out several rules or conditions from the tree that are used to recommend the auction to manufacturers as shown in following examples.

CPU auction recommendation:

- If the item is CPU, then we *recommend* the auction.



<Figure 6> Decision tree of manufacturers' auction recommendation.

The first rule about the CPU auction is of course obvious as stated previously and does not provide any meaningful context. Computer and Shell auctions for manufacturers are more complicated than that of CPU.

Computer auction recommendation:

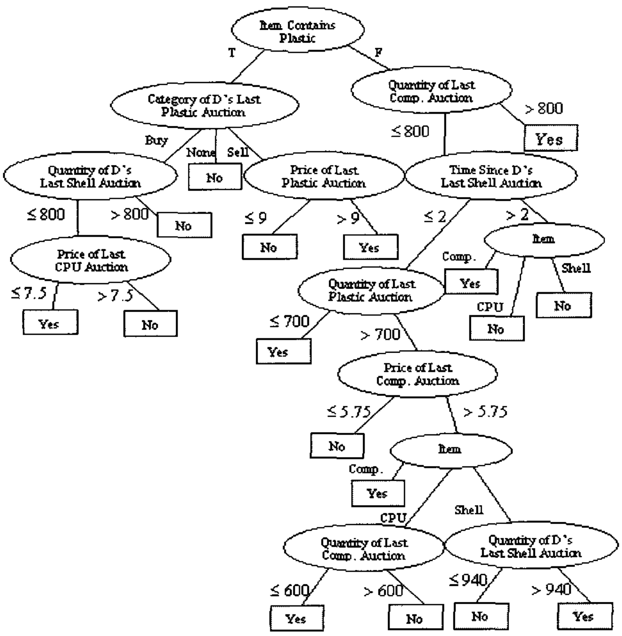
- If the time since the manufacturer last participated in a computer auction is less than or equal to 1 round, then *recommend*the auction.
- Else if the time since the manufacturer last participated in a computer auction is greater than 1 round and the manufacturer bought computers in the last auction, then *recommend* the auction.
- Else if the manufacturer sold computers, then *do not recommend.*

It may be interpreted as follows; if the manufacturer participated in the computer auction right before the current round or has never been participated, the manufacturer want to sell computers, which is natural. Otherwise if the manufacturer did not participate in the last round of a computer auction and has bought computers previously, the manufacturer wants to participate in the current auction. That implies that the manufacturer bought computers to earn more money that can be used to buy CPU and Plastic in the last auction, or the manufacturer want to sell computers that have left up to the current round. That is not surprising in that everyone wants to maximize a profit. Thus, even though manufacturers sold all computers they had, they would want to make more coffee makers to be sold continuously, which means they act as a broker. Thinking collectively, we recommend a computer auction to manufacturers inferred as they have unsold computers. For the other auctions, the system can generate recommendation rules that can be interpreted in the same manner above.

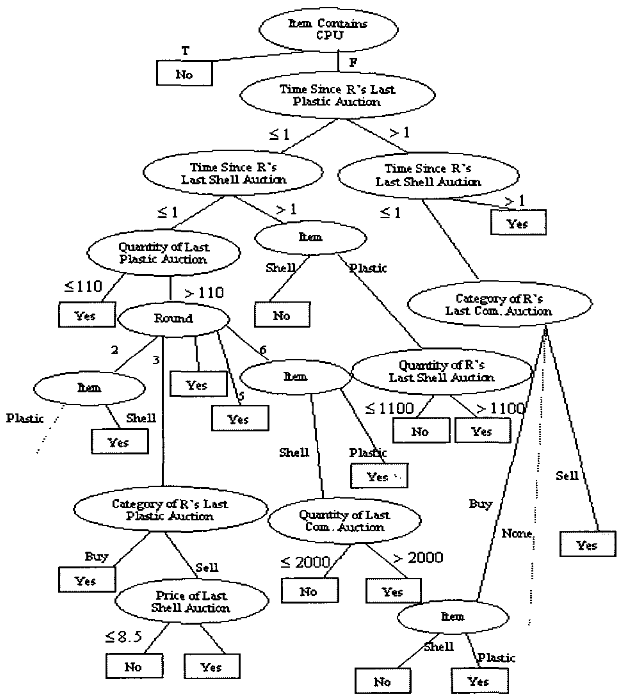
Now let's look at the decision tree for demanufacturers (See Figure 7). The top node of demanufaturers splits according as an item is plastic or not. Intuitively if the item is plastic, the demanufacturer wants to act as a broker since a plastic is not a part of memory boards. If the item is plastic, the recommendation rules for demanufacturers are as follows;

- If the demanufacturer sold plastics last with higher price, then *recommend* the auction.
- Else if the demanufacturer bought plastics last and traded large amount of shells last, then do *not recommend* the auction.

- Else if the demanufacturer traded small amount of shells last and CPUs were traded with low price last, then *recommend* the auction.
- Else, do *not recommend* the auction.



<Figure 7> Decision tree of demanufacturers' auction recommendation

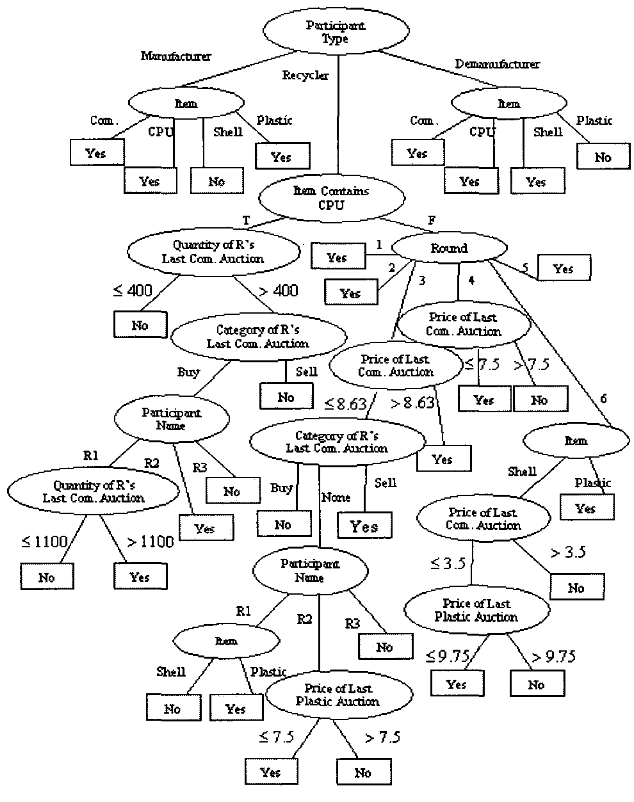


<Figure 8> Decision tree of recyclers' auction recommendation.

Since demanufacturers behave as a broker, they would be interested in auctions that they can make much profit, which is directly reflected in the first rule. The rest of rules imply that if they did not earn money from shell and CPU auctions, they would turn their interest to the plastic auctions to make money through the broker's behaviors.

For recyclers, what kind of an auction should be recommended? In a word, recyclers do not act behaviors of a broker in that the first node splits depending on whether or not the item contains CPU. The items that contain CPU are computer and CPU itself. Thus from the tree, it is easily found that the auction for items containing CPU is not recommended to recyclers, which implies that recyclers do not participate in auctions as a broker only.

We investigated recommendation rules based on decision tree induced classification algorithm, C4.5, using a full size of data set with 32 features and 2592 instances.



<Figure 9> Decision tree reduced by feature selection

Generally, most learning algorithms are not scalable with the increasing number of features. Thus it is critical to reduce the feature dimension for faster induction of classification. Even though the number, 32, in this prototyped data set is quite manageable, real data sets would

have huge amount of data. In order to reduce the feature dimension, the NPW is used with C4.5. Hence, we have much smaller size of data set with 10 selected features but even accuracy improvement, 84.7% versus 85.5%.
The selected 10 features are as follows;

- Item
- ItemContainsCPU
- ItemContainsShell
- Round
- PriceForLastComputerAuction
- PriceForLastPlasticAuction
- QuantityForParticipantsLastComputerAuction
- CategoryOfParticipantsLastComputerAuction
- ParticipantName
- ParticipantType

The induced decision tree based on the reduced data set is shown in Figure 9. Comparing time to build the model, 1.94 seconds for full data set is significantly reduced to 0.64 seconds for reduced data set. Of course, the size of the tree also becomes much smaller, which means that the small size of decision tree can be more easily interpreted and provide clear recommendation.

5. Conclusion

We have developed a new optimization based feature selection methods that can be implemented as both a filter and a wrapper. The methods provide significant contributions in that the NP based feature selection algorithm has an optimization framework even presenting a scalable structure and can effectively be used to create learning models that are easily interpreted as a preprocessing step prior applying learning algorithms. The major contribution of this paper is that the new approach with an optimization framework can guarantee an optimal solution given a certain distance of the optimum with a given probability after a finite time stopping criterion is satisfied. The numerical results show that the new method performs quite well on several comparison test problems. Through numerical results, we showed why intelligent partitioning is important with respect to accuracy and speed and that using random sampling on instances can be a potential way for handling large number of instances in the NPF. As a case study of the NP feature selection method, we constructed recommender systems for auction participation and auction bids using a classification rule with their interpretations. It would be meaningful that this research provided how feature selection and learning algorithm can make contribution on an online auction system. However since the recommendation rules were derived using data gathered from a prototyped system having some limitation, for example, sealed bid double auction system and limited number of rounds, products and users. Thus, it is hard to apply these results to a real auction system by the stated limits, which could be further investigated for applying the system into a real situation as future research.

References

[1] Blake, C.L. and Merz, C.J., 1998, *UCI Repository of machine learning databases* <http://www.ics.uci.edu/mlearn/MLRepository.html>, University of California, Irvine, CA (Date Accessed: October 31, 2003).

[2] Bradley, P.S., Mangasarian, O.L., and Street, W.N., 1998, "Feature selection via mathematical programming", *INFORMS Journal on Computing*, 10(2), 209-217.

[3] Hall, M.A., 1998, "Correlation-based feature selection for discrete and numeric class machine learning", in *Proceedings of the Seventeenth International Conference on Machine Learning*, Stanford University, CA. Morgan Kaufmann.

[4] Kim, Y.S., Street, W.N, and Menczer, F., 2000, "Feature selection in unsupervised learning via evolutionary search", in *Proceedings of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*

[5] Liu, H. and Motoda, H., 1998, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Kluwer Academic Publishers.

[6] Narendra, P.M., and Fukunaga, K., 1977, "A branch and bound algorithm for feature subset selection", *IEEE Transactions on Computers*, 26(9), 917-922.

[7] Olafsson, S. and Yang, J., 2005, "Intelligent partitioning for feature selection", *INFORMS Journal on Computing*, (in print).

[8] Ryan, S.M., Min, K.J. and Olafsson, S., 2001, "Experimental study of scalability enhancements for reverse logistics E-Commerce", in Prabu, Kumara and Kamath (eds.) *Scalable Enterprise System An I-ntroduction to Recent Advances.*

[9] Shi, L. and Olafsson, S., 2000, "Nested partitions meth-

od for global optimization", *Operations Research*, 48, 390-407.

[10] Witten, I. and Frank, E., 2000, "Data mining: practical machine learning tools and techniques with Java implementations", Morgan Kaufmann

[11] Yang, J. and Honavar, V., 1998, "Feature subset selection using a genetic algorithm" In H. Motada and H. Liu (eds), *Feature Selection, Construction, and Subset Selection: A Data Mining Perspective*, Kluwer, New York.