

A Genetic Algorithm-based Data Mining Approach to Profiling the Adopters and Non-Adopters of E-Purchasing

Dr. Hokey Min
(Corresponding Author)
Logistics and Distribution Institute
University of Louisville
Louisville, KY 40292

Tomasz G. Smolinski and Grzegorz M. Boratyn
Logistics and Distribution Institute,
Computational Intelligence Laboratory
University of Louisville
Louisville, KY 40292

Abstract

With the proliferation of the electronic commerce e-purchasing has become a daily practice for many purchasing organizations. To embrace e-purchasing successfully, these organizations should identify determinants that are crucial for its successful adoption. In an effort to identify such determinants, including organizational readiness, information technology infrastructure, and user characteristics, we propose a novel genetic algorithm-based data mining technique. The application of the proposed data mining technique to empirical data that were collected through a mail survey proves to be useful for extracting hidden, but valuable insights into the successful implementation of e-purchasing.

1 Introduction

Electronic commerce (e-commerce) generally refers to an inter-organizational information system that is intended to facilitate business-to-business electronic communication, information exchange, and transaction support through a web of networks. E-commerce can take variety of forms such as electronic data interchange (EDI), Internet, Intranet, Extranet, online catalog ordering, and e-mail [9]. The volume of electronic commerce (e-commerce) for U.S. business transactions is expected to increase to \$327 billion by the year 2002 [11].

Despite all the potential managerial benefits that e-commerce can bring to purchasing practices (e.g. cost savings resulting from reduced paper transactions, shorter order cycle time, and enhanced opportunities for the supplier/buyer partnership through the establishment of a web of business-to-business communication networks), some firms are slow in adopting e-commerce as a purchasing tool. Therefore, there is a

need to develop a profile of adopters and non-adopters of e-purchasing and then to provide guidance for those who may consider using electronic commerce in the future.

Data mining has been used in e-commerce for some time already. It has many applications in this field such as: searching for patterns in transactional data, preparation of personalization applications, etc. [7]. However, before this kind of data analysis will be ever possible, an e-commerce system itself needs to be successfully implemented.

Data used in this research project were obtained from the survey in which the questionnaire was mailed to about 3000 randomly selected members of the National Association of Purchasing Management. The questions were related to the size of the company, purchase volume, extent of involvement in e-purchasing, importance of advantages of e-purchasing usage, severity of barriers of e-purchasing application, etc.

In previous work on the same data performed by Min and Galle [10], based on some preliminary statistical data analysis of the sample and the characteristics of the problem, a few hypotheses describing differences between adopters and non-adopters of e-purchasing were then stated. The hypotheses were verified using some traditional techniques (i.e. discriminant analysis, test of independence and cross-tabulation, chi-square test, etc.) and some conclusions were drawn. This well-known and popular approach produced some interesting insight into the domain of the problem. However, it was clearly limited to the confines of the predefined hypotheses and might have overlooked some hidden and not-so-obvious, yet important, patterns in the data. Thus, there is a need for an application of some automated and indirect data mining technique that would be able to elaborate on the previous findings and possibly discover some new relevant information.

2 Research Methodology

One of the data mining techniques that aim at the discovery of hidden knowledge is extraction of association or pseudo-association (attributes are not limited to the binary domain) rules in the form of IF-THEN statements. The theory of association rules was introduced in [1]. Since then, the methodology has been continuously improved and modified and various algorithms have been developed. Some of the modifications can be found in [2], [6]. However, very high computational complexity of the task of searching for association rules in large databases leaves the problem of the efficiency of those algorithms open. To deal with the complexity problem of various rule discovery-oriented data mining tasks in large datasets (including association rules), many artificial intelligence-related and heuristics-based solutions have been proposed recently. Application of genetic algorithms with properly defined fitness function is one of them [3], [4], [5], [8]. Most of those modern solutions apply a genetic algorithm directly to the process of a rule production by defining a fitness function in terms of the rule's support and confidence, its structure, classification error, etc. This kind of approach is obviously limited by the characteristics of the methodology of genetic algorithms (i.e. quasi-optimal solutions), but definitely tends to be more efficient. However, for really huge databases, computation of such created fitness function can also become an "expensive" task. In this paper, a slightly different approach combining the power of genetic algorithms with the simplicity of standard association rule generation algorithms will be described. This approach applies a genetic algorithm to the problem of searching for repetitive patterns hidden in data and then simply generates pseudo-association rules based on those patterns.

2.1 Assumptions

The main goal of this project was to identify the profiles of e-purchasing adopters. It was also very important to determine the most significant factors in terms of the firm's perceived importance of managerial benefits as well as the obstacles related to the possible implementation of an e-purchasing system. The methodology described in this paper, allows for an introduction of some automation into such an analysis. The whole knowledge that can be discovered with this approach is hidden in data and can be extracted virtually with no assumptions. The only requirement that must be fulfilled prior to the investigation of this kind of relations in the data is to divide the attribute space

into two disjoint subspaces representing *premise* and *consequent* parts of the decision rules. Obviously, in most cases, this kind of division is determined already at the time of preparation of the survey itself, since the authors of such a survey know the target of their investigation as well as the domain of the findings in advance. What is usually unknown at that stage is what questions in the survey are directly related to the given problem, to what extent answers to those questions affect the results, and what the intermediate relations and dependencies between some parts of the survey are.

2.2 Searching for patterns

The theory of genetic algorithms is based on the process of natural selection, according to which the nature aims at the creation of organisms that will be adjusted to the surrounding environment in the best possible way. Basically, genetic algorithms can be used for solving problems for which it is possible to construct an objective function to estimate how a given representative (*solution*) fits the considered environment (*problem*). In genetic algorithms, possible solutions to the analyzed problem are encoded into so-called *chromosomes*. Chromosomes consist of *genes* that represent a solution numerically. Possible values that can be assigned to a particular gene are determined by its *allele* (*domain*). By iterative application of genetic operators (*crossover*, *mutation*, and *selection*) to a whole population of such chromosomes, genetic algorithms create solutions that are usually not perfect but quite satisfactory in terms of the fitness function optimization.

In this project, the chromosomes that are being produced and modified along the process of evolution (a sequence of generations) represent patterns covering records in the data set. Each one of them has the length of the number of attributes describing the data (61 attributes in this case). The alleles of the chromosome are constrained by the domains of those attributes (i.e. depending on the number of possible answers to a question). Hence, the alleles of such a chromosome consist of all values valid for the attribute corresponding to a given gene and a "don't care" asterisk, which means that this attribute is not important, and will not be used for generation of a rule. The "don't care" asterisk also represents a missing value from the data set (i.e. no response at all). An example of such a chromosome is shown in Figure 1.

Each of such patterns has a possible coverage in the data (*support*), which is given by the number of records matching the pattern. For the example shown

```
***1*****5*1*****3*****
```

Figure 1: Example of a chromosome (set positions - genes no. 4, 11, 13, 22).

```
t=0;
P(t) := InitializePopulation( no_of_attributes,
                             attributes_domains );
while ( t < max_number_of_generations ) do
  EvaluateFitness( P(t), dataset );
  t := t+1;
  P(t) := Select( P(t-1) );
  Crossover( P(t) );
  Mutate( P(t) );
end while;
```

Figure 2: Pseudo-code for the general scheme of the genetic algorithm used in the project.

in Figure 1 it will be the number of all records containing given values at fourth, eleventh, thirteenth, and twenty second positions, no matter what are all the other values. Obviously, we are mostly interested in patterns that have relatively high support and this will be the main feature of the fitness function used for this algorithm. The minimal, desired level of support in data can be specified before the execution of the genetic algorithm, so that all the patterns with less coverage will not be included in the result at all. Pseudo-code for the general scheme of the genetic algorithm is shown in Figure 2.

Although the support of a pattern is a basic feature of the fitness function implemented in the algorithm, it cannot be its ultimate characteristic. The number of “set” positions (other than “don’t care” asterisks) is also very important. For example, a pattern consisting only of asterisks will gain support of 100% of the data records, but it has no meaning in terms of knowledge discovery. The structure of the IF-THEN rules generated afterwards is also very important, and it should be kept in mind that the patterns must contain at least two “set” attribute values in order to stand as a basis for any useful association rule.

Obviously, not all the chromosomes will have a physical coverage in the data. Some of them (especially those with relatively large number of “set” positions) might not have a support at all, however some parts of them (subsets of values) still can be very useful and after an application of some genetic operators (i.e. crossover) may produce a desired result. It is crucial then to appropriately treat all those chromosomes and assign them some credit in terms of the fitness

```
FitnessEvaluation( chromosome )
  set_positions :=
    CountSetPositions( chromosome );
  if ( HasSupportinData( chromosome ) ) then
    support := CalculateSupport( chromosome );
    fitness := support * set_positions;
  else
    partial_support :=
      CalculatePartialSupport( chromosome );
    fitness := partial_support * threshold_support;
  end if;
  FitnessEvaluation := fitness;
end FitnessEvaluation;
```

```
CalculatePartialSupport( chromosome )
  for each record in dataset
    matching :=
      CountGenesMatchingRecord( record );
    matching_ratio :=
      matching / length_of_chromosome;
    partial_support :=
      partial_support + matching_ratio;
  end for;
  CalculatePartialSupport :=
    partial_support / no_of_records;
end CalculatePartialSupport;
```

Figure 3: Pseudo-code for the fitness function evaluation used in the project.

function even though they do not have support in the data as a whole.

All these aspects have to be precisely encoded and implemented into the algorithm and all the chromosomes (potential solutions) should be awarded or punished according to the criteria stated above during the process of evolution. Thus the fitness function can be completely defined as a multi-layer estimation of the fitness of the chromosomes in terms of their partial support in the data at first, and then total coverage of the data weighted by the number of “set” positions. More detailed description of the algorithm for the fitness function evaluation is presented by the pseudo-code in Figure 3.

Another very important feature of the proposed genetic algorithm is a multi-point crossover option. In many experiments on different types of data, this approach was found to be much more effective with respect to both the number of discovered patterns, and the time of convergence. The number of crossover points for a given process of evolution, as well as the crossover positions in chromosomes are selected randomly based on some predefined constraints (i.e. max-

imal number of crossover points).

As an outcome of several evolutions modeled by this genetic algorithm, a set of data patterns was created. Those patterns, along with the information about the level of their support, were then used as an input to the second algorithm that generated pseudo-association rules.

2.3 Rule generation

Association rules expose the existence of relations between attributes in data (binary domain - exists vs. does not exist) while pseudo-association rules discover relations between values of those attributes (domains of the attributes themselves). Basically, pseudo-association rules are simple IF-THEN statements:

“If the set of attributes X , included in the premise part of the rule, has some values, described by a set of values $V(X)$, then the set of attributes Y , included in the consequent part, tends to have values, described by another set of values $V(Y)$ ”.

In our case, since we aim at the derivation of rules of type:

“If a given company’s profile, in a sense of attributes X , is described by the values $V(X)$, then the likelihood of the company being involved in adoption of particular e-commerce solution(s), described by a set of attributes Y , is determined by the level of importance of the managerial benefits it looks for and concerns with some obstacles $V(Y)$ ”

and:

“If a given company perceives managerial benefits X important to the extent of $V(X)$, then its likelihood of adoption of a particular e-commerce solution(s) Y is determined by the level of concern with some obstacles $V(Y)$ ”.

this approach seems to be ideal.

Association and pseudo-association rules are characterized by their support in data (number of cases that a given rule applies to, i.e. how “popular” the rule is) and confidence (ratio of the support of the rule to the number of cases that contain its premise part, i.e. how sure one can be that judging on the basis of the values from the premise part of the rule the rule will be true).

An algorithm of extraction of association rules usually consists of two parts: searching for patterns hidden in data (in this project achieved by application of the genetic algorithm) and generation of rules based on those patterns.

The idea of the rule generation is relatively simple. For a given partition of the attributes into two disjoint subspaces representing *premise* and *consequent*

parts of the expected rules, the algorithm takes the set of all the “non-asterisk” values of a particular pattern. Based on those values, it creates all possible pairs (P, C) , where P and C are non-empty, disjoint subsets of that set, such that the attributes included in P and C are subsets of *premise* and *consequent* respectively. This yields all the possible candidates for rules in the form of IF P THEN C , based on this particular pattern. After a validation of those candidates in terms of the expected levels of support and confidence, the rules are selected.

Both support and confidence parameters are standard features of association rules mining theory. They allow the user to set a desired level of usefulness for the discovered knowledge. By increasing and decreasing them different sets of rules can be found (i.e. those with small support but strong confidence vs. the ones with higher support but less credibility). This also allows the user to examine the data from different points of view.

2.4 Algorithm complexity

Apart from the execution of the standard genetic algorithms operators, the comparison of the chromosome to the data records is the most critical part of the algorithm. Effectiveness of this portion of the algorithm strongly depends on four parameters: the number of records in the database (N), the number of attributes in the database (A), the size of the population (P), and the number of generations (G) within a given evolution process. In each generation, the algorithm scans the data only once and matches all the chromosomes to the records that are being scanned. The size of the population is usually much smaller than the size of the data sample, so it is much more efficient to perform several scans on the population, than to go through the data many times (the data is usually stored on a disk while the population resides in the computers memory). Since the matching is nothing more but a value-to-gene comparison of each chromosome in the population against each record in the database, the total number of such comparisons (C) is given by:

$$C = N * A * P * G.$$

As for the second part of the methodology presented in this paper - rule generation, its efficiency obviously depends on the number of the attributes included in a given pattern and on the constraints concerning the rule structure in terms of its premise and consequent parts (provided by the user).

The total complexity of this approach might seem to be quite high at first. One must keep in mind that due to the fact that the search space is radically lim-

ited, in practice, the algorithm works very effectively. Because of the emphasis on the fast discovery of patterns by the genetic algorithm in the first part, the rule generation is rather fast. What is even more important, it is very flexible and allows the user to generate different set of rules (i.e. different points of view) on the basis of the same set of patterns.

3 Results

3.1 Preliminary statistical data analysis

The first step of the data exploration in this project was a basic statistical analysis of the answer frequencies. On the basis of this stage of the investigation some conclusions were drawn. Those conclusions were quite interesting but at the same time very general. However, one must not forget that such a preliminary insight into the data domain based on the frequency analysis is always very useful in terms of any further analysis. It may indicate some points of interest that might be embraced by the knowledge hidden in the data and, what is even more important from the point of view of the data mining approach described in this paper, it determines the levels of support and confidence of potentially discovered patterns and rules (the levels of support and confidence of the rules are obviously directly related to the frequency of occurrence of the attributes' values included in those rules). Accordingly, the results of this preliminary statistical data analysis were carefully analyzed and taken into account while determining the genetic algorithm's parameters.

3.2 Searching for patterns

In order to increase the variety of patterns, the genetic algorithm was launched on several computers simultaneously. Because of the randomness aspects of genetic algorithms, depending on the starting population of chromosomes and the direction of the search that a particular evolution process had taken, the results differed from one another in terms of attribute selection (i.e. attributes used in particular patterns). However, some of those results were duplicated, and this had to be considered prior to the step of rule generation.

Because of the relatively small size of the data sample as well as the conclusions derived from the preliminary statistical data analysis (about small diversity of the data), support threshold of desired patterns was lowered to 3 - 5%. Obviously, having such a small

database of examples we usually cannot expect particular patterns to occur very frequently (it usually takes quite large amount of data to allow for some patterns to dominate the sample), however we should be able to identify them even at a lower rate of occurrence. This level of support threshold might seem to be extremely low, but it is valid because rules based on patterns with relatively small support in data still may have quite large level of confidence.

As a result of several evolutions (each consisting of comparable number of generations) of the genetic algorithm, 1564 patterns were found in the database. Each of those patterns had at least two "set" values for the corresponding attributes.

3.3 Rule generation

Patterns discovered and prepared in the previous step were then used as the basis for rules generation. At this level, the sets of attributes were divided into premise and consequent parts of the rules adequately to the problem specification. The input patterns were also checked for overlapping, and those that were covered by another pattern were removed from consideration. After this verification, 633 effective patterns were preserved. On the basis of this final set of patterns, a number of rules of given support and confidence was generated.

Only, on the level of 10% of minimal support and 75% of minimal confidence, more than 100 interesting rules were found. Some of those rules simply validated the conclusions drawn from the preliminary statistical data analysis and the previous work, but some of them produced much more precise description of adopters' and non-adopters' profiles. A few examples of those rules are presented below:

RULE 1:

IF a company uses EDI system, THEN the company is willing to help suppliers to establish an electronic commerce network WITH support of 32% and confidence of 79%.

RULE 2:

IF a company uses electronic commerce in purchase orders frequently, THEN the reduction of transaction time is extremely important for them WITH support of 21% and confidence of 81%.

4 Summary

All of the generated rules were definitely reasonable, however not all of them were so obvious and could not be easily anticipated. Some of them simply confirmed the conclusions drawn from the statistical data analysis while others produced an interesting and novel insight into the problem of profiling adopters and non-adopters of e-purchasing.

On the basis of these results it can be stated that this approach is appropriate and useful for the discovery of the profile description hidden in data. It creates an environment for automatic exploration of survey data, as well as knowledge discovery based on modern, artificial intelligence-oriented methodology, which is quite novel approach in the field. This methodology, as it was shown above, can generate interesting rules even from a quite small set of data and provided with a larger database could only improve its performance.

Similar approaches (based on genetic algorithm-driven search for association rules), had been applied to various data mining problems before, however the complexity of the fitness functions used in those projects was usually very high. This approach, on the other hand, due to the decomposition of the task into two separate phases (i.e. searching for patterns and rule generation) not only effectively puts the limits on the search space, but also allows for a flexible manipulation of the rule structure in terms of its premise and consequent parts and thus makes it possible to analyze the discovered knowledge from different perspectives at a virtually no additional cost.

References

- [1] Agrawal R., Imielinski T., Swami A., Mining Association Rules Between Sets of Items in Large Databases *Proceedings of the ACM SIGMOD '93*, pp. 207-216. Washington, D.C., May 1993.
- [2] Agrawal R., Srikant R., Fast Algorithms for Mining Association Rules, *Proceedings of the 20th Int'l Conf. On Very Large Databases (VLDB '94)*, Santiago, Chile, September 1994.
- [3] Fidelis M.V., Lopes H.S., Freitas A.A., Discovering Comprehensible Classification Rules with a genetic algorithm, *Proceedings of the Congress on Evolutionary Computation 2000 (CEC-2000)*, pp. 805-810, La Jolla, California, July 2000.
- [4] Flockhart I. W., Radcliffe N. J., A Genetic Algorithm-based Approach to Data Mining, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 299-302, Portland, Oregon, 1996.
- [5] Freitas A.A., A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pp. 96-101, San Francisco, California, July 1997.
- [6] Hipp J., Guntzer U., Nakhaeizadeh G., Mining Association Rules: Deriving a Superior Algorithm by Analyzing Today's Approaches, *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery*, Lyon, France, September 2000.
- [7] Kohavi R., Provost F., Applications of Data Mining to Electronic Commerce, *Data Mining and Knowledge Discovery - International Journal*, Special Issue on E-Commerce and Data Mining, Kluwer Academic Publishers, Boston, 2001.
- [8] Liu J.J., Kwok J.T., An Extended Genetic Rule Induction Algorithm, *Proceedings of the Congress on Evolutionary Computation (CEC)*, pp. 458-463, La Jolla, California, July 2000.
- [9] Min H., Galle W.P., Electronic Commerce Usage in Business-to-Business Purchasing, *International Journal of Operations and Production Management*, Vol. 19, No. 9, pp. 909-921, 1999.
- [10] Min H., Galle W.P., E-Purchasing: Profiles of Adopters and Non-Adopters, *Logistics and Distribution Institute Working Paper Series*, University of Louisville, Louisville, Kentucky, 2001.
- [11] Radstaak B.G., Ketelaar M.H., Worldwide Logistics: The Future of Supply Chain Services, *Holland International Distribution Council*, Hague, The Netherlands.