Recommendation Rules — a Data Mining Tool to Enhance Business-to-Customer Communication in Web Applications

Mikołaj Morzy

Institute of Computing Science
Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland
Mikolaj.Morzy@cs.put.poznan.pl

Abstract. Contemporary information systems are facing challenging tasks involving advanced data analysis, pattern discovery, and knowledge utilization. Data mining can be successfully employed to sieve through huge amounts of raw data in search for interesting patterns. Knowledge discovered during data mining activity can be used to provide value-added services and benefits to users, customers, and organizations.
The adoption of the Web as one of the main media for business-to-customer (B2C) communication provides novel opportunities for using data mining to personalize and enhance customer interfaces. In this paper we introduce the notion of recommendation rules — a simple knowledge model that can be successfully used in the Web environment to improve the quality of B2C relationship by highly personalized communication. We present the formalism and we show how to efficiently generate recommendation rules from a large body of customer data.

1 Introduction

Rapid scientific, technological, cultural, and social development witnessed in recent years has resulted in significant increase of the volume of data gathered and processed by computer systems. Data mining, also referred to as knowledge discovery in databases, aims at the discovery of hidden and interesting patterns from large data volumes. Discovered patterns can be used, e.g., to provide additional insight into the data, to allow prediction of future events, or to assist marketing operations. The main drawback of data mining systems is the high computational cost of knowledge discovery algorithms which disqualifies many data mining methods from straight utilization in on-line Web applications.

The Web is quickly becoming an important channel for sales and customer relationship management. Several businesses, including banks, insurance companies, retail and multimedia stores, are interacting with their customers on-line. This transition to the on-line communication channel introduces new, unprecedented requirements. Among others, user expectations of response times shrink to seconds, user identification becomes difficult, security and privacy become key issues. Increasing customer satisfaction requires precise mechanisms of B2C communication. A simple approach of broadcasting all messages to all customers,

contemptuously referred to as the "spray and pray" method, is unacceptable. Messages must be highly relevant to customers with respect to customer behavior and characteristics. Assessing the relevance is difficult, because customer personal data is often limited. On the other hand, highly personalized communication is very important in marketing applications.

Let us consider an example. A bank wants to inform its customers about a new credit card. The marketing department decides that the main addressee of the message is a young person who lives in a mid-sized city and who has a medium income. Direct translation of these constraints into a database query can be error-prone. Subjective choice of attribute value thresholds (e.g., deciding that income is "medium" if it falls within the range $\langle 2000, 3000 \rangle$) can lead to false positives (addressing customers who should not be bothered with the message), or false negatives (failing to address customers who should be notified).

An attempt to solve this problem using traditional data mining techniques leads to the utilization of clustering and classification. Unfortunately, both techniques are not appropriate here. Clustering is not helpful at all, because addressees of messages do not form distinct clusters. The set of addressees is determined dynamically upon the formulation of a new message and the constraints for message delivery are vague. In other words, clusters representing addressees of messages would have to be strongly overlapping, irregular, having different shapes and sizes. Classification is not helpful either, because it is impossible to acquire high-quality training and testing sets. We propose to tackle this problem from the association rule discovery perspective. In our solution conditions that determine the relevance of a message to a given customer are not formulated arbitrarily. Rather, conditions represent frequent combinations of attribute values and can be chosen by the user from a precomputed set of possible combinations. Additionally, each customer is approximated by frequent combinations of attribute values present in customer data. In other words, every customer is mapped to a point in a multidimensional space of frequent attribute values. Messages are also mapped to the multidimensional space as subregions. The inclusion of a customer point in a given message subregion implies that the customer is an addressee of the message.

In this paper we present recommendation rules — a simple knowledge representation model that allows high personalization of B2C communication. We introduce the notion of a recommendation rule and we present an adaptation of the well-known Apriori algorithm to pre-compute frequent sets of attribute values. The results presented in this paper originate from a prototype implementation of the system developed within the Institute of Computing Science of Poznań University of Technology. The paper is organized as follows. Section 2 presents related work. In Section 3 we present basic definitions and we formally introduce the notion of a recommendation rule. Section 4 contains the description of the mining algorithms. We conclude in Section 5 with a brief summary.

2 Related Work

The problem of mining association rules was first introduced in [AIS93]. In [AS94] Agrawal et al. proposed the Apriori algorithm that quickly became the seed for several other frequent itemset mining algorithms. The original formulation of the association rule mining problem was generalized into the problem of mining quantitative association rules in [SA96].

Tightly coupled with quantitative association rules are clustered association rules first presented by Lent et al. in [LSW97]. The idea behind clustering was to combine quantitative association rules for which rule antecedents or consequents corresponded to adjacent ranges of attribute values. Another similar problem was the problem of finding profile association rules, first presented by Aggarwal et al. in [ASY98]. An exhaustive study of the subject can be found in [ASY02]. Profile association rules correlate patterns discovered in user demographics data with buying patterns exhibited by users.

3 Definitions

Given a set of attributes $A = \{A_1, A_2, \ldots, A_n\}$. Let $dom\,(A_i)$ denote the domain of the attribute A_i. Let the database D consist of a relation R with the schema $R = (A_1, A_2, \ldots, A_n)$. For each tuple $r \in R$ let $r\,(A_j) = a_j$ denote the value of the attribute A_j in tuple r. The support of the value a_j of the attribute A_j is the ratio of the tuples $r \in R$ having $r\,(A_j) = a_j$ to the number of tuples in R. Given a user-defined minimum support threshold denoted as *minsup*. The value a_j of the attribute A_j is called *frequent*, if $support_R\,(A_j, a_j) \geq minsup$. The support of the set of values $\{a_j, \ldots, a_m\}$ of the attributes $A_j, \ldots, A_m$ is the ratio of the tuples $r \in R$ having the values of the attributes $A_j, \ldots, A_m$ equal to $a_j, \ldots, a_m$, respectively, to the number of tuples in R. The set of values $\{a_j, \ldots, a_m\}$ of the attributes $A_j, \ldots, A_m$ is *frequent*, if $support_R\,(A_j, a_j, \ldots, A_m, a_m) \geq minsup$. We say that a set of attribute values $\{a_j, \ldots, a_m\}$ *satisfies* the tuple r if $\forall k \in \langle j, \ldots, m\rangle : r\,(A_k) = a_k$. Let L denote the collection of all frequent sets of attribute values appearing in the relation R. Given a set of messages $M = \{m_1, m_2, \ldots, m_p\}$, where each message m_i is a string of characters. A *recommendation rule* is an implication of the form: $a_j \wedge a_k \wedge \ldots \wedge a_m \rightarrow k_i$, where $a_j \in dom\,(A_j) \wedge a_k \in dom\,(A_k) \wedge \ldots \wedge a_m \in dom\,(A_m) \wedge \exists l_q \in L : \{a_j, a_k, \ldots, a_m\} \subseteq l_q$. The left-hand side of the rule is called the antecedent and the right-hand side of the rule is called the consequent.

Each tuple $r \in R$ can be approximated using frequent sets of attribute values appearing in the tuple. The processing of recommendation rules consists in finding, for a given tuple r, all recommendation rules which apply to r, i.e., in finding all recommendation rules having the antecedent satisfying the tuple r. It is worthwhile noticing that this formulation of data mining task is fundamentally different from previous approaches. Previous approaches concentrated on efficient discovery of associations between attribute values. Our approach takes the opposite direction, i.e., the knowledge is known *a priori* (we assume that the

user has a vague notion of constraints that should be satisfied in order to send a message to a given customer), but the formulation of the knowledge is difficult. Therefore, we propose to reverse the process. Instead of formulating constraints and looking for customers satisfying those constraints, we begin with the in-depth analysis of the customer data and we derive frequent sets of attribute values to serve as descriptors for large user communities. Next, we force the user to formulate the constraints for message delivery only in terms of frequent sets of attribute values discovered during the first step. In this way, the user can not introduce arbitrary conditions that cut across communities of similar customers and the constraints for message delivery become "natural" in the sense that they represent natural clustering of attribute values present in customer data.

4 Mining Algorithms

Require: D, $minsup$, P
  $Normalize(D, P)$;
  $Discretize(D, P)$;
  $RemoveCorrelatedAttributes(D, P)$;
  L_1 = set of frequent attribute values;
  $L = Apriori(D, L_1, minsup)$;
  for all tuples $t \in D$ **do**
    $L_t = subset\,(L, t)$;
    for all sets $l \in L_t$ **do**
      $\langle t.t\_id, l.s\_id \rangle \rightarrow metadata$
    end for
  end for

Require: L, M
  $rhs = \{m_i : m_i \in M\}$;
  $lhs = \emptyset$;
  $L_{LHS} = L$;
  while ($notFinished$) **do**
    $lhs = \{l : l \in L_{LHS}\}$;
    $L_{LHS} = L_{LHS} \setminus \{l : l \in L_{LHS} \wedge l \not\supseteq lhs\}$;
    $notFinished \leftarrow user\ input$
  end while

Fig. 1. Algorithms for generation of frequent sets and recommendation rules

Figure 1 presents the algorithms for generating frequent sets and recommendation rules. This algorithm is a minor modification of the Apriori algorithm [AS94]. Let D denote the database of customer data. Let $M = \{m_1, \ldots, m_k\}$ denote the set of user-defined messages. Finally, let P denote the set of user-defined preferences (e.g., the correlation factor for pruning correlated attributes, parameters for attribute normalization and discretization, etc.). The first algorithm begins by performing necessary data preprocessing, such as normalization of numerical attributes and discretization of numerical attributes into discrete bins. Next, the algorithm generates all frequent sets of attribute values using the Apriori technique. In the last step, all tuples describing customers are verified for the containment of frequent sets of attribute values. For every customer tuple the information about all frequent sets of attribute values contained in the given tuple is added to the metadata. This step is necessary for achieving a satisfying performance during runtime. Let *lhs* and *rhs* denote the left-hand side and the right-hand side of the generated recommendation rule, respectively. The user first selects the messages to be communicated to customers and adds them to the right-hand side of the rule. Next, the user adds conditions to the left-hand side of the rule. Conditions are represented by frequent sets of attribute

values. In each iteration the user is free to choose from the collection of available frequent sets of attribute values, where the collection consists of all supersets of frequent sets already chosen for the left-hand side of the rule. The rationale behind this is that it guarantees that every message will be communicated to at least *minsup* fraction of customers, since every left-hand side must necessarily belong to the collection of frequent sets of attribute values. Specialization of a given recommendation rules continues until the user is satisfied with the joint condition.

5 Summary

In this paper we have presented the idea of recommendation rules. It is a knowledge representation model that allows organizations to personalize messages addressed to their customers, avoiding the arbitrary choice of message delivery criteria. In addition to the formulation of the problem, we have presented an algorithm for efficient definition of recommendation rules. Our prototype implementation proves that this idea is applicable in real-world applications. The integration of data mining techniques with Web applications is a very promising research area. Recommendation rules and the prototype presented in this paper provide an interesting step into this domain.

References

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[ASY98] Charu C. Aggarwal, Zheng Sun, and Philip S. Yu. Online algorithms for finding profile association rules. In *Proc. of the seventh international conference on Information and knowledge management*, pages 86–95. ACM Press, 1998.

[ASY02] Charu C. Aggarwal, Zheng Sun, and Philip S. Yu. Fast algorithms for online generation of profile association rules. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1017–1028, 2002.

[LSW97] Brian Lent, Arun N. Swami, and Jennifer Widom. Clustering association rules. In Alex Gray and Per-Åke Larson, editors, *Proceedings of the Thirteenth International Conference on Data Engineering, April 7-11, 1997 Birmingham U.K*, pages 220–231. IEEE Computer Society, 1997.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996*, pages 1–12. ACM Press, 1996.