User Profiling in Personalization Applications through Rule Discovery and Validation

Gediminas Adomavicius
New York University
adomavic@cs.nyu.edu

Alexander Tuzhilin
New York University
atuzhili@stern.nyu.edu

Abstract

In many applications, ranging from recommender systems to one-to-one marketing to Web browsing, it is important to build personalized profiles of individual users from their transactional histories. These profiles describe individual behavior of users and can be specified with sets of rules learned from user transactional histories using various data mining techniques. Since many discovered rules can be spurious, irrelevant, or trivial, one of the main problems is how to perform post-analysis of the discovered rules, i.e., how to validate customer profiles by separating "good" rules from the "bad." This paper presents a method for validating such rules with an explicit participation of a human expert in this process.

1 Introduction

Over the last few years *personalization* or *mass customization* became an important business problem in various applications including one-to-one marketing [PR93], recommender systems [cac97], and personalized Web content presentation applications. It has been argued in [PR93] that personalization approaches provide several important advantages over more traditional segmentation methods.

One of the key technical issues in developing personalization applications is the problem of how to construct accurate and comprehensive *profiles* of individual customers that provide the most important information describing who the customers are and how they behave.

In the data mining community, the profiling problem was first studied in [FP96] within the context of fraud detection in the cellular phone industry. This was done by learning rules pertaining to individual customers from the cellular phone usage data using the rule learning system RL and then generating generalized profilers for different customer segments in order to learn general fraud conditions for various groups of customers. The problem of on-line mining of customer profiles specified with association rules was studied in [ASY98]. The body of a rule considered in [ASY98] refers to the customer demographic information, such as age and salary, and the head of a rule refers to the transactional information, such as purchasing characteristics of the customer. This approach segments customers based on their transactional characteristics and does not derive behavior of *individual* customers in one-to-one fashion [PR93]. Besides the academic community, profiling problem was also addressed in the industry by several companies, including Engage Technologies (www.engagetech.com), BroadVision (www.broadvision.com), and Open Sesame (www.bowneinternet.com/solutions/sesame.htm).

In this paper we present an approach to the profiling problem were user profiles are learned from the transactional histories using data mining methods. However, the behavioral rules learned about individual customers can be unreliable, irrelevant, or obvious. Therefore, rule validation becomes an important issue for building accurate personal profiles of the users. We address this important issue and present a solution to the validation problem in this paper.

2 Constructing User Profiles

In order to explain what user profiles are and how they can be constructed, we first focus on the data that is used for constructing these profiles.

Data Model. Various personalization applications can contain different types of data about individual users. However, in many such applications, this data can be classified into two basic types – *demographic* and *transactional*, where demographic data describes who the user *is* and transactional data describes what the user *does*. For example, in a marketing application demographic data would include name, gender, birth date, salary, etc. The transactional data would consist of records of purchases the customer made over a specific period of time. A purchase record would include such attributes as the date of purchase, product purchased, amount of money spent, use or no use of a coupon, value of a coupon if used, discount applied, etc.

Profile Model. As mentioned above, a profile is a collection of information that describes a user. We classify this information into two components – the *factual* profile and the *behavioral* profile. A factual

KDD-99 San Diego CA USA

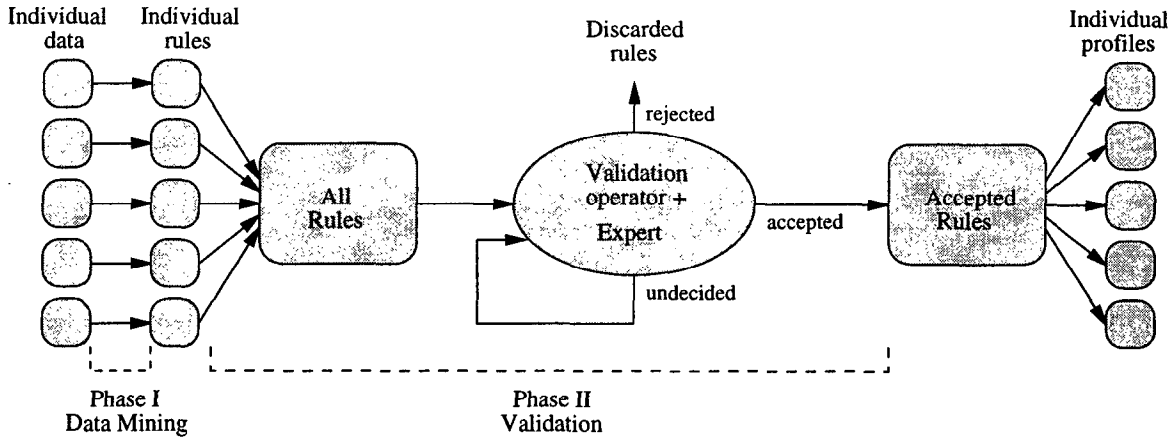

Figure 1: The expanded view of the profile building process.

profile contains specific *facts* about the user, including the demographic data. Also, a factual profile can contain facts derived from the transactional data, such as "the favorite beer of customer X is Heineken" or "the biggest purchase made by X was for $237". The construction of factual profiles is a relatively simple and well-understood problem since they can be modeled with a record in a table of a relational database.

A *behavioral* profile models the behavior of a user, and we do this using conjunctive rules, such as association [AMS+96] or classification rules [BFOS84]. An example of a "behavior" is: *"when shopping on weekends, user X usually spends more than $100 on groceries"*. The use of rules in profiles provides an intuitive, declarative and modular way to describe user behavior.

Profile Construction. Since we focus on personalization applications, rule discovery methods are applied *individually* to the transactional data of *every* user, thus, capturing truly personal behavior of users.

Such rules can be discovered using various data mining algorithms. For example, to discover association rules, we can use *Apriori* [AMS+96], and for classification rules we can use *CART* [BFOS84]. Moreover, our approach *is not limited* to any specific representation of data mining rules and their discovery methods.

One of the problems with many data mining methods is that they tend to generate a large number of patterns, and most of them, while being statistically acceptable, are trivial, spurious, or just not relevant to the application at hand [PSM94, ST96, BMUT97]. Therefore, post-analysis of discovered rules becomes an important issue, since there is a need to *validate* the discovered rules. For example, assume that a data mining method discovered the rule that, whenever customer X goes on a business trip to Los Angeles, he stays in expensive hotels there. Assume that the customer went to Los Angeles 7 times over the past 2 years and 5 out of 7 times stayed in expensive hotels. We need to validate this rule, i.e., to make sure that it really captures the behavior of customer X rather than constitutes a spurious correlation or is simply not relevant to the application at hand. As another example, consider the discovered rule saying that, whenever customer X buys tomatoes, X usually also buys soap. It is not clear if this rule captures a truly causal link between tomatoes and soap inherent to X's purchasing behavior, or it exists because X tends to do really large shoppings periodically and tends to buy *everything* together, including tomatoes and soap.

Therefore, after data mining methods discover rules describing individual user behavior in personalization applications, it is important to validate these rules.

3 Validation

A common way to perform post-analysis of discovered rules is to let a domain expert do the validation of these rules. We adopted this approach and let an expert either "accept" or "reject" the discovered rules. Then the accepted rules form the behavioral profiles of users.

One of the main issues with validating individual rules of users by a human expert is the scalability issue. In many personalization applications the number of users tends to be very large, e.g., measured in millions. If we discover a hundred rules per customer on average, then the total number of rules in such applications would be measured in hundreds of millions. Therefore, it is impossible for a human expert to validate all these rules on a one-by-one basis in such applications.

We address this problem in the paper by providing a method allowing the human expert validate *large numbers* of rules (instead of individual rules) at a time with relatively little input from the expert. This is done by introducing several *rule validation operators*. Then rule validation becomes an *iterative* process, where various operators are applied successively, thus, allowing the expert to validate many rules each time

a validation operator is applied until some termination condition is reached.

Figure 1 illustrates this process and divides the profile building activity into two phases. In Phase I, data mining phase, rules describing behaviors of individual users are generated from the users' transactional data as was described in Section 2.

Phase II constitutes the rule validation process. Rule validation, unlike rule discovery (Phase I), is not done separately for each user, but rather for *all* users at once. The reason we propose to do rule validation collectively (rather than individually) for all users is that there are usually many similar or even identical rules across different users. For example, the rule *"when buying cereal, user X also buys milk"* can be common to many users. In addition, although rules *"user X usually uses manufacturer's coupon when buying apple juice"* and *"user Y usually uses store coupon when buying orange juice"* are not identical, they are quite "similar" and can be examined by the expert together. The collective rule validation allows to deal with such common rule once, whereas when validating rules for each user separately, the expert might end up working on many identical or similar rules over and over again. Therefore, in the beginning of Phase II, rules from all the users are collected into one set. Each rule is tagged with the user ID, so that after the validation, each accepted rule could be put into the profile of that user.

After rules from all users are collected into one set, the rule validation process is performed as a second part of Phase II. This process is described in Figure 2. All rules discovered during Phase I (denoted by R_{all} in Figure 2) are considered unvalidated. The human expert selects various validation operators and applies them successively to the set of unvalidated rules. The application of each validation operator results in validation of some of the rules. In particular, some rules get accepted and some rejected (sets O_{acc} and O_{rej} in Figure 2). Then the next validation operator would be applied to the set of the remaining unvalidated rules (set R_{unv}). This validation process stops when the *TerminateValidationProcess* condition is met. There are many different ways to specify this termination condition, including the following:

- The validation process continues until some large percentage of rules (e.g., 95%) is validated. This percentage can be specified by the expert in advance.

- The validation process terminates when validation operators validate only few rules at a time, that is, when the "costs" of selecting and applying each additional validation operator exceed the "benefits" of getting few more rules validated (the law of diminishing returns).

```
Input:  Set of all discovered rules R_all.
Output: Mutually disjoint sets of rules R_acc, R_rej,
        R_unv, such that R_all = R_acc ∪ R_rej ∪ R_unv.

(1) R_unv := R_all, R_acc := ∅, R_rej := ∅.
(2) while (not TerminateValidationProcess()) begin
(3)     Expert selects a validation operator (say, O)
        from the set of available validation operators.
(4)     O is applied to R_unv.
        Result: mutually disjoint sets O_acc, O_rej,
        O_other, such that R_unv = O_acc ∪ O_rej ∪ O_other.
(5)     R_acc := R_acc ∪ O_acc, R_rej := R_rej ∪ O_rej,
        R_unv := O_other.
(6) end
```

Figure 2: Basic algorithm for the rule validation process.

After the validation process is stopped, the set of all the discovered rules (R_{all}) is split into three disjoint sets: accepted rules (R_{acc}), rejected rules (R_{rej}), and some remaining unvalidated rules (R_{unv}). At the end of Phase II all the accepted rules are put into the behavioral profiles of their respective users. This is possible, because all the rules have been tagged with the user ID in the beginning of Phase II.

4 Validation Operators

In this section we describe several validation operators that we propose to use in the rule validation process, including similarity-based grouping, template-based filtering, visualization, redundant rule elimination.

Similarity-based rule grouping. Puts "similar" rules into groups according to the expert-specified similarity criterion. As a result, the expert can inspect *groups* of rules instead of inspecting individual rules one-by-one. Validation (acceptance or rejection) of a group of rules means that all rules contained in the group are validated together as a group. To accomplish this, we have developed a method providing the expert with abilities to specify different levels of similarity of the rules. We also developed an efficient (linear running time) rule grouping algorithm, presented in [AT], which takes a set of rules and a similarity condition specified by the expert and produces groups of similar rules.

Example 1 Consider, the following "attribute structure" similarity condition, according to which all the rules having the same set of attributes (but, maybe, different attribute values and statistical parameters) are similar. Consider association rules *"(1) Product = LemonJuice ⇒ Store = RiteAid (supp=2.4%, conf=95%)"* and *"(2) Product = WheatBread ⇒ Store = GrandUnion (supp=3.5%, conf=88%)"*. With "attribute structure" similarity condition, the grouping operator would group rules 1 and 2 into the same group, because they both have the attribute structure *"Product ⇒ Store"*. Consequently, any rule that has such attribute structure would be placed into the same group as the two rules mentioned above. An example of the rule that would not be grouped into the group mentioned above would be the rule *"(3) Product = AppleJuice*

$\Rightarrow$ *CouponUsed = YES (supp=2.1%, conf=60%)*", because it has different attribute structure, i.e., *"Product $\Rightarrow$ CouponUsed"*. The "attribute structure" similarity condition provides only one example of many possible similarity conditions that can be used by this operator.

Template-based rule filtering. Filters rules that match the expert-specified rule template. The expert can specify both *accepting* and *rejecting* templates. Naturally, rules that match an accepting template are accepted, rules that match a rejecting template are rejected. Rules that do not match a template remain *unvalidated*. In particular, we introduced a rule template specification language (based on the approach presented in [KMR+94]) and developed an efficient (linear running time) matching algorithm, presented in [AT], that returns the rules from a given set that match an expert-specified template.

Example 2 Consider the following rule template: *"REJECT HEAD = {Store = RiteAid}"*. This template can be interpreted as "Reject all rules that have *Store = RiteAid* in the head of the rule". Out of three rules mentioned in Example 1, only rule 1 matches the given template and, as the result, would be rejected. An example of a somewhat more complicated rule template could be: *"ACCEPT BODY $\supseteq$ {Product}; HEAD $\subset$ {DayOfWeek, Quantity}; CONF > 65%"*. This rule template can be interpreted as "Accept all rules that have attribute *Product* (possibly among other attributes) in the body of the rule, that also have either *DayOfWeek* or *Quantity* in the head of the rule, and that also have confidence more than 65%".

Redundant rule elimination. Eliminates the rules that can be derived from other, usually more general, rules and facts. In other words, this operator eliminates the rules that, by themselves, do not carry any new information about the behavior of a user. More specifically, we propose in [AT] several redundancy conditions and the algorithm, that checks rules for specified redundancy conditions, and eliminates the ones that satisfy them.

Example 3 Consider the association rule *"Product = AppleJuice $\Rightarrow$ Store = GrandUnion (2%, 100%)"*, which was discovered in the purchasing history of one particular customer X. This rule by itself might look like it really shows the specifics of X's behavior (X buys apple juice *only* at Grand Union) and, therefore, may seem logical enough to be put into the X's behavioral profile. However, assume, that it was also determined from the data that this customer does *all* of the shopping at Grand Union. Then the previous rule constitutes a special case of this finding. Obviously, keeping the fact *"the customer shops only at Grand Union"* in the X's factual profile eliminates the need to store the above mentioned rule in the X's behavioral profile.

Visualization Operators. Allow the expert to view the set of unvalidated rules or various parts of this set in different visual representations (histograms, pie charts, etc.) and can give the expert insights into what rules are acceptable and can be included in profiles.

Validation operator	Number of rules: accepted	rejected	unvalidated
1. Eliminate redund.	0	186,727	836,085
2. Filter	0	290,427	545,658
3. Filter	0	268,157	277,501
4. Filter	6,711	0	270,790
5. Filter	0	233,013	37,777
6. Group (1,046 gr.)	16,047	1,944	19,786
7. Group (6,425 gr.)	4,120	863	14,803
Final:	26,878	981,131	14,803

Figure 3: Validation process for a marketing application.

Statistical Analysis Operators. Statistical analysis operators can compute various statistical characteristics (value frequencies, attribute correlation, etc.) of unvalidated rules. This allows the expert to have many different "views" of these rules, therefore aiding the expert in the rule validation process.

Browsing Operators. As mentioned above, visualization and statistical analysis operators allow the expert to have "aggregated" views of the unvalidated rules through various visual representations and statistical characteristics. Browsing operators, on the other hand, allow the expert to inspect individual rules directly.

Browsing operators are especially useful when combined with the described above similarity-based grouping operator. Instead of browsing through individual rules and manually validating (accepting or rejecting) them on one-by-one basis, the expert can apply the grouping operator and then browse the resulting groups and manually validate a group at a time. Browsing operators can have some additional capabilities, such as being able to sort the content to be browsed in various ways. For example, it might be helpful for the expert if browsing operators could sort rules by the user ID or sort groups by their size on demand.

5 Implementation of a Validation System and a Case Study

We implemented the methods presented in Sections 3 and 4 in the 1:1Pro system.[1] The 1:1Pro system takes as inputs the demographic and transactional data, and generates a set of validated rules capturing personal behaviors of users following the approach presented in this paper. The 1:1Pro system can use various relational DBMSs to store user data and various data mining tools for discovering individual users' rules.

The 1:1Pro system was tested on a real life marketing application that analyzes customer reactions to various types of promotions, including advertisements, coupons, and various kinds of discounts. The applica-

[1] 1:1Pro stands for One-to-One Profiling System.

tion included data on 1,903 households that purchased different types of beverages over a period of one year. The data set contained 353,421 purchasing transactions (on average 186 records per household) characterized by 21 attributes, such as customer ID, date and time of purchase, type of product, and coupon usage. The data mining module of the 1:1Pro system executed Apriori-like rule discovery algorithm on the individual household data for *each* of the 1,903 households and generated 1,022,812 association rules in total, on average about 537 rules per household.

Since we were familiar with this application, we performed the role of experts and validated the 1,022,812 discovered rules ourselves using the sequence of validation operators presented in Figure 3. Along with the type of each operator that was used in this application, Figure 3 also lists the numbers of rules that were accepted, rejected, and remained unvalidated by that particular validation operator. For example, the filtering operator 4 from Figure 3 accepted rules that state direct relationship between kinds of products purchased and various promotions, i.e., rules that have product information (possibly among other attributes) in the body and promotion-related information (discount, sale, coupon used, or advertisement seen) in the head. Using this particular operator we were able to validate (in this case accept) 6,711 rules. As another example, consider the grouping operator (operator 6 in Figure 3), which grouped the 37,777 unvalidated rules into 1,046 groups, where the biggest group contained 2,364 rules and the smallest group had just 1 rule in it. We inspected 50 biggest groups and were able to validate 38 of them (31 accepted and 7 rejected), which brought the unvalidated rule count down to 19,786.

After applying seven validation operators (listed in Figure 3), we managed to validate all but 14,803 rules (out of 1,022,812). We stopped the validation process at this point because of the diminishing returns. This validation, including expert and computing time, took about 1.5 hours, during which we validated 98.5% of the initially discovered rules. The total number of accepted and rejected rules constituted 2.6% and 95.9% respectively of the initially discovered rules. The total number of rules accepted and put into profiles was 26,878 (on average, about 14 rules per household).

We would like to reiterate that the validation process that we just described is *subjective*, that is, different domain experts can use the tools provided by 1:1Pro in various ways and produce different validation results.

6 Conclusions

In this paper, we addressed the problem of constructing accurate behavioral profiles of individual users by using data mining methods to generate these profiles and validating them in the post-processing stage using similarity-based grouping, template-based filtering, redundant rule elimination, browsing, visualization, and other operators. We also tested our profiling system on a real-life marketing application. Our system managed to validate 98.5% from the total number of 1,022,812 discovered rules. This demonstrates that our approach can validate significant percentages of rules for medium-scale personalization applications. Since our algorithms are linear in the number of customers and generated rules, we expect that the validation process should be able to scale up well.

Acknowledgments. We would like to thank Sergei Levin for implementing a part of the 1:1Pro system and the Center for Advanced Technologies at NYU for supporting this work.

References

[AMS+96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, ch. 12. AAAI Press, 1996.

[ASY98] C. C. Aggarwal, Z. Sun, and P. S. Yu. Online generation of profile association rules. In *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998.

[AT] G. Adomavicius and A. Tuzhilin. Efficient rule post-analysis methods for personalization applications. In preparation.

[BFOS84] L. Breiman, J. H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth Publishers, 1984.

[BMUT97] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD Conference*, 1997.

[cac97] *Communications of the ACM*, 40(3):88–89, 1997. Special issue on Recommender Systems.

[FP96] T. Fawcett and F. Provost. Combining data mining and machine learning for efficient user profiling. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, August 1996.

[KMR+94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the Third International Conference on Information and Knowledge Management*, December 1994.

[PR93] D. Peppers and M. Rogers. *The One-to-One Future.* Doubleday, 1993.

[PSM94] G. Piatetsky-Shapiro and C. J. Matheus. The interestingness of deviations. In *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*, 1994.

[ST96] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE TKDE*, 8(6), 1996.