# Mining Patterns That Respond to Actions*

Yuelong Jiang and Ke Wang
Simon Fraser University
{yjiang,wangk}@cs.sfu.ca

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

Ada Wai-Chee Fu
Chinese University of Hong Kong
adafu@cse.cuhk.edu.hk

## Abstract

*Data mining focuses on patterns that **summarize** the data. In this paper, we focus on mining patterns that could **change** the state by responding to opportunities of actions.*

## 1. Introduction

Data mining is about extracting interesting patterns from raw data. To quote [2], "Merely finding the patterns is not enough. You must be able to respond to the patterns, to act on them, ultimately turning the data into information, the information into action, and the action into value". Recently, Kleinberg et al. presented a microeconomic view of data mining [5] (the market segmentation): partition the customer base $C$ into $k$ parts $C_1, \cdots, C_k$ to maximize the sum of the optima $\Sigma_1^k max_{x \in D} \Sigma_{i \in C_j} c_i \cdot x$, where $c_i \cdot x$ denotes the utility of a decision $x$ on a customer $i$. Their work assumes the total knowledge about the utility of a decision on a customer (i.e., $c_i \cdot x$). In some applications, however, such total knowledge is not available, and only the partial knowledge that certain "actions" affect certain "features" may be known. The following example makes our point concrete.

**Example 1.1** Consider a miniature teaching evaluation database with two features $F_1$ and $F_2$, and one utility $U$. $F_1$ and $F_2$ represent the score on grading fairness and communication skills. $U$ represents the overall score of the instructor's evaluations. All scores are on the scale of 0,1,2 (with 0 being the worst and 2 being the best). As the user knowledge, the action $A_1$ of providing grading supervision can improve $F_1$ if $F_1 = 0$, and the action $A_2$ of sending the instructor to a communication workshop can improve $F_2$ if $F_2 = 0$. Though this knowledge does not directly hint the utility $U$, we would expect $A_1$ and $A_2$ to improve $U$ if there is a correlation between $F_i$ and $U$.

*This work was supported by a grant from NSERC.

Suppose that our goal is partitioning the following data for taking actions to maximize $U$:

$G_1 : F_1 = 0, F_2 = 0, U = 0$ (100 cases)
$G_2 : F_1 = 1, F_2 = 0, U = 1$ (100 cases)
$G_3 : F_1 = 1, F_2 = 1, U = 1$ (100 cases)
$G_4 : F_1 = 2, F_2 = 2, U = 2$ (20 cases).

The classic decision tree [7], which will partition the data according to $F_1$ to minimize the prediction error, does not facilitate our goal well: only the partition for $F_1 = 0$ correctly characterizes 100 cases for action $A_1$; the partition for $F_1 = 1$ does not characterize any action because taking the action $A_2$ means sending a half of the partition to the workshop without any improvement (since $A_2$ is applicable only if $F_2 = 0$). A better partitioning is by $F_2$ because the partition for $F_2 = 0$ correctly characterizes the 200 cases for action $A_2$. ■

This example motivates the following problem: given the historical data about the features $F_1, \cdots, F_m$ and the utility $U$, and assuming that the user knows that some actions $A_i$ affect some features $F_j$ in a "simple" way, we want to find the patterns about applying actions to boost the utility $U$. Such patterns can help to understand the action/utility relationship and can be used to recommend actions for future cases. Unlike [5], we do not assume the knowledge about the utility of actions; in fact, finding and summarizing the action/utility relationship is our goal. Rather, we assume the knowledge about how actions affect features, which is usually simple and known to the user.

Given a future case, one obvious approach is simply applying all applicable actions to the case. Unfortunately, this does not necessarily increase the utility because several actions could be conflicting with each other in affecting the utility. In addition, such a case-based approach does not produce any pattern that summarizes the action/utility relationship. The next example shows that finding such patterns requires examining both the user knowledge and the historical data.

**Example 1.2** Consider a customer database in the long distance call application:

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE

$$D(CustID, Married, Rate, \cdots, U).$$

$U$ represents the profit generated on a customer and has the domain $\{Low, High\}$. $Rate$ represents the rate charged to a customer, with $Normal$ denoting the normal rate and $Special$ denoting the promotional rate. Suppose that we observe the following two populations in the data

$$P : Rate = Normal, Married = No \rightarrow U = Low,$$
$$P' : Rate = Special, Married = No \rightarrow U = High.$$

Intuitively, the customers in $P'$ generate a higher profit by making more calls at the lower rate. Comparing these two populations reveals that the telephone company can increase the profit by offering the special rate to the customers in $P$. The increase is expected because a higher profit was observed on the customers (i.e., $P'$) who shared similar characteristics (i.e., unmarried) but were offered the special rate. ■

If certain features are correlated with the utility in some population (i.e., $Rate$ is correlated with $U$ for unmarried customers), a change in those features implies a certain change in the utility. Now if some actions can influence those features, the influence will cascade to the utility through the correlation. We are interested in summarizing those actions and populations where such cascaded influences increase the utility. In this paper, we formulate a new action mining problem, called *AFU (Action-Feature-Utility) mining*, to capture this notion of actionability.

Some previous work has been done on action mining. [6] defines actionability in terms of restoring deviations back to the norm in the specific context of healthcare application. A domain-independent approach based on action hierarchy and pattern templates was presented in [1]. [8] presented a profit-motivated (as opposed to hit-motivated) recommender approach. [9] requires the user to provide a "cost" for changing a feature value. Obtaining such cost will be the bottleneck in practice. For example, it is difficult for the user to determine the cost required for converting a customer from the attrition state to the loyal state, or to tell the cost required to score higher on the teaching evaluation question "Explain difficult concepts effectively". Our approach models the typically available action/feature knowledge, not the cost, therefore, is more scalable in practice.

## 2. Overview

Consider a data set $D(F_1, \cdots, F_m, U)$, where $F_i$ are *features* and $U$ is the *utility* to be maximized. $F_i$ and $U$ have a *ranked domain*, consisting of a small number of linearly ordered scales, represented by the first few integers 0,1,... For example, donation scale, skill level, letter grade, performance evaluation, all can be abstracted into a ranked domain. Following [5], we shall be calling elements of $D$ "customers". Available is a set of *actions* $A_1, \cdots, A_n$. An *action model*, to be defined formally in Section 3, specifies how an action $A_i$ increases or decreases the current value of a feature $F_j$.

Suppose that, given a set of customers $D_i$, we can determine a set of actions $AS_i$, or an *actionset*, for $D_i$. Further, suppose that there is a method that, given $AS_i$ and $D_i$, estimates the *actionability* of increasing the utility $U$ by taking the actions $AS_i$ on $|D_i|$ *future* customers randomly drawn from the same distribution as $D_i$. Let $PAct_i$ denote this actionability. For example, $PAct_i$ in Example 1.2 is the estimated profit increase after offering the special rate to $|P|$ *future* customers matching the description of the population $P$.

**Definition 2.1** In the *Action-Feature-Utility (AFU) mining* problem, we want to find a partition of the data, $\{D_1, \cdots, D_q\}$, that maximizes the aggregated actionability:

$$\Sigma_i PAct_i. \quad (1)$$

A solution consists of $AS_i, D_i$ and the description of $D_i$, $1 \leq i \leq q$. ■

Given a future customer matching the description of $D_i$, we can recommend the actionset $AS_i$ to the customer. The recommendation maximizes the utility over the underlying distribution of $D_i$. For a large $D$, the AFU mining is a very hard problem, since it requires optimization over all partitions of $D$, therefore, computation exponential in the size of $D$. In fact, this problem seems to be harder than the market segmentation in [5] in that the utility of actions and the number of partitions are not specified, and the maximization is on future customers. Below, we focus on the action model, the actionability and an approximate solution.

## 3. The Action Model

### 3.1. Influence Matrix

The action model, denoted $\{M_{ij}\}$, specifies the influence of an action $A_i$ on a feature $F_j$, where $1 \leq i \leq n$ and $1 \leq j \leq m$. $M_{ij}$, called the *influence range* of $A_i$ on $F_j$, has the form $[a, b]$, where

1. $a = b = -$, if $A_i$ has no influence on $F_j$.
2. $a < b$, if $A_i$ increases the current value $c$ in $[a, b]$ for $F_j$ to some target value in the *destination range* $[c, b]$.
3. $a > b$, if $A_i$ decreases the current value $c$ in $[a, b]$ for $F_j$ to some target in the *destination range* $[b, c]$.

The *NIL action* has no influence on any feature. $A_i$ changes the current value $c$ to the trivial destination range $[c, c]$ if $A_i$ has no influence on $F_j$, or $c$ is not in the influence range $[a, b]$, or $A_i$ is the NIL action.

As an example, suppose that the user knows that attending the communication workshop increases any communication skill level $f$ in the range $[1, 5]$ to some level in $[f, 5]$. This knowledge can be represented by the entry $M_{ij} = [1, 5]$ for the action and the feature, with the following implications. (1) The action has no influence on the instructors whose communication skill level is less than 1 or more than 5. (2) The new level after the action is in the range $[f, 5]$, but the exact level is not specified, as usually the case. (3) We make no prior assumption about the distribution of this value in $[f, 5]$. As we will see shortly, the data itself will provide this information.

The above model implicitly assumes that the influence of an action on a feature is independent of the state of other features and the influence of other actions. We make this assumption because the human user tends to work better by dealing with one thing at a time. In practice, the user knowledge is largely imprecise and approximating due to the human bottleneck, and capturing complete user knowledge is neither desirable nor tractable. The naive Bayes classifier [3] is an example of making an independence assumption and achieving good performance results.

### 3.2. Actionsets

Under the independence assumption, applying an actionset will change the current feature value to a range given by the union of the destination range of applying each action separately.

**Definition 3.1** Consider an actionset $AS$ and a feature value $F_j = f_j$. Let $[l_{ij}, h_{ij}]$ be the destination range of applying $A_i \in AS$ to $F_j = f_j$. The *destination range* of applying $AS$ to $F_j = f_j$ is $[l_j, h_j]$, where $l_j = \min(l_{ij})$ and $h_j = \max(h_{ij})$. ■

In other words, $DS_{1,k} = ([l_1, h_1], \ldots, [l_k, h_k])$ describes the feature ranges after applying $AS$ to any customer matching $\hat{f} = (f_1, \ldots, f_k)$. If $AS$ contains no action, we assume that it contains the NIL action so that $AS$ is non-empty.

**Example 3.1** Refer to the influence matrix in Table 1. Assume that all features have the domain 0-6. Consider the actionset $\{A_1, A_2\}$ and the vector $(F_1 = 3, F_2 = 3)$. $A_1$ increases $F_1 = 3$ to any value in $[3, 5]$ and increases $F_2 = 3$ to any value in $[3, 6]$; $A_2$ decreases $F_1 = 3$ to any value in $[2, 3]$ and has no effect on $F_2 = 3$. Therefore, the destination space of applying $\{A_1, A_2\}$ to $(F_1 = 3, F_2 = 3)$ is $([2, 5], [3, 6])$. Note that $A_1$ and $A_2$ have conflicting influence on $F_2$. ■

| | $F_1$ | $F_2$ | $F_3$ |
|---|---|---|---|
| $A_1$ | [1,5] | [1,6] | [2,4] |
| $A_2$ | [4,2] | [2,0] | [2,5] |
| $A_3$ | [0,5] | [0,5] | [2,6] |
| $A_4$ | [-,-] | [4,0] | [5,0] |

**Table 1. Influence matrix**

### 3.3. Actionability of an Actionset

So far, we considered how actions may affect features. Ultimately, we are interested in their actionability in terms of increasing the utility $U$. Consider the population $P \subseteq D$ described by the vector $\hat{f} = (f_1, \ldots, f_k)$. From Definition 3.1, an actionset $AS$ will change any customer $c$ in $P$ to the destination space $DS_{1,k}$. To estimate the actionability of this change, we need to estimate the new utility of $c$ after the change.

Our approach is examining those customers that fall into the destination space $DS_{1,k}$, i.e., $\{c' \mid c' \in D \wedge c' \in DS_{1,k}\}$, and using their utility to estimate the new utility of $c$. Intuitively, such customers serve as a "role model" for how $c$ would perform after taking the actions $AS$. We call this set of customers the *role model* of $P$ and denote it by $rm(P)$. The actionability of $AS$ on a customer $c$ in $P$ then is measured by $agg(rm(P)) - c.U$, where $agg(rm(P))$ denotes some aggregate, such as average, maximum or minimum, of the utility of the customers in $rm(P)$ and $c.U$ denotes the utility of $c$. The actionability of $AS$ on $P$ is the sum of the actionability of $AS$ over the customers in $P$.

**Definition 3.2** Given a population $P \subseteq D$ described by a vector $\hat{f}$, and an actionset $AS$, the *actionability* of $AS$ on $P$ is defined as:

$$Act(P, AS, \hat{f}) = \Sigma_{c \in P}(agg(rm(P)) - c.U). \blacksquare$$

Note that $Act(P, AS, \hat{f})$ refers to observed customers in the given data $D$, whereas $PAct$ in Equation (1) refers to future customers.

**Example 3.2** Consider the following data set $D$ and the influence matrix in Table 1.

$$c_1 : F_1 = 0, F_2 = 0, F_3 = 1, U = 0$$
$$c_2 : F_1 = 2, F_2 = 1, F_3 = 1, U = 1$$
$$c_3 : F_1 = 3, F_2 = 2, F_3 = 2, U = 2$$
$$c_4 : F_1 = 3, F_2 = 3, F_3 = 2, U = 3$$
$$c_5 : F_1 = 3, F_2 = 3, F_3 = 3, U = 5$$
$$c_6 : F_1 = 4, F_2 = 4, F_3 = 4, U = 6$$
$$c_7 : F_1 = 5, F_2 = 5, F_3 = 4, U = 6$$
$$c_8 : F_1 = 5, F_2 = 6, F_3 = 6, U = 4$$

The population described by $(F_1 = 3, F_2 = 3)$ is $P = \{c_4, c_5\}$. The destination space of applying $\{A_1, A_2\}$ to $P$ is $([2,5],[3,6])$, and the role model of $P$ is $rm(P) = \{c_4, c_5, c_6, c_7, c_8\}$. If $agg$ is the average function, $agg(rm(P)) = 4.8$. The actionability of applying $\{A_1, A_2\}$ to $P$ is $(4.8-3)+(4.8-5) = 1.6$. ■

## 4. An Approximate Solution

We present an approximate solution to the AFU mining by iteratively partitioning the data to maximize the actionability in Equation (1). Initially, we have a single node representing the whole data $D$. At each step, we partition a leaf node according to a feature selected by some selection criterion, i.e., partitioning the customers at the parent node according to the values of the feature. This tree growth continues until for any leaf node either there is no available feature or the selection criterion becomes non-positive.

Let us consider the selection criterion of features. Consider partitioning a leaf node $N_p$ into the child nodes $N_i$, $1 \leq i \leq u$, according to the values $f_{k1}, \cdots, f_{ku}$ of some feature $F_k$. At the node $N_p$, we have the vector $\hat{f}_p = (f_1, \ldots, f_{k-1})$, the actionset $AS_p$, the population $P_p$ described by $\hat{f}_p$, the destination space $DS_{1,k-1}$, and the role model $rm_p$ of $P_p$. $\hat{f}_p$ corresponds to the branching conditions on the path from the root to $N_p$. Similarly, at each child node $N_i$, we have the vector $\hat{f}_i = (f_1, \ldots, f_{k-1}, f_{ki})$, the actionset $AS_i$, the population $P_i$ described by $\hat{f}_i$, the destination space $DS_{1,k}$, and the role model $rm_i$ of $P_i$.

The usefulness of $F_k$ at $N_p$ is measured by the promise of increasing the actionability $\Sigma PAct_i$ in Equation (1) after partitioning $P_p$ into $P_i$ and applying $AS_i$ to $P_i$. $PAct_i$ is the actionability of $AS_i$ on $|P_i|$ *future* customers drawn from the same distribution as $P_i$, where $P_i$ is a set of observed customers and $AS_i$ is the set of actions applicable to $P_i$. Since future customers are unknown, we can only heuristically maximize $\Sigma PAct_i$ using $Act(P_p, AS_p, \hat{f}_p)$ and $Act(P_i, AS_i, \hat{f}_i)$ (Definition 3.2), or simply $Act(N_p)$ and $Act(N_i)$, defined on the observed customers at $N_p$ and $N_i$. The usefulness of $F_k$ can then be measured by the increase of $Act$ after partitioning the data using $F_k$, called "actionability gain" below.

**Definition 4.1** The *actionability gain* of $F_k$ at $N_p$, denoted $ActGain(F_k, N_p)$, is defined as

$$\sum_{i=1}^{u} Act(N_i) - Act(N_p). \blacksquare$$

To partition the population $P_p$ at the node $N_p$, we select the feature $F_k$, among all available features, that maximizes $ActGain$. To compute $ActGain$, we need to determine the actionset $AS_i$ at a child node $N_i$ so that $Act(N_i)$ can be computed. $AS_i$ is a set of actions that are "applicable" at $N_i$. The exact definition of $AS_i$ is a bit technical and is given in the full report [4]. Based on $AS_i$, we can compute $DS_{1,k}$, as in Definition 3.1, and the role model $rm_i$ of $P_i$ at $N_i$.

The above criterion is based on observed customers (i.e., $Act$), therefore, may not maximize the actionability on future customers due to potential over-fitting. To address this issue, we need to prune the fully expanded tree to maximize the *projected actionability* over the *entire population* at a node $N$ (i.e., $PAct(N)$). $PAct(N)$ can be estimated using a modification of the *pessimistic error estimation* adopted in [7]. The pruning process then is similar to the bottom-up pruning of the decision tree [7]: at each node we check whether pruning the subtree can increase the projected actionability, if yes, prune the subtree immediately, and if not, keep the subtree. Our empirical evaluation shows that this method generates a higher utility than ignoring the opportunities of actions. The details are reported in [4].

## 5. Conclusion

Actionability mining is an important but under-studied topic. A key issue is modeling the user knowledge in a scalable manner. Previous works assume the availability of the cost for actionability. Obtaining such cost will be the bottleneck in practice. A contribution of our work is the introduction of "actions" as a domain-independent and scalable way to capture a certain type of user knowledge. We also presented a concrete framework for incorporating such knowledge to obtain utility function.

## References


[1] G. Adomavicius and A. Tuzhilin. Discovery of actionable patterns in databases: The action hierarchy approach. In *KDD*, pages 111–114, 1997.

[2] C. Derman. *Finite State Markov Decision Process*. Academic Press, New York, 1970.

[3] R. Duda and P. Hart. Pattern classification and scence analysis. In *Wiley*, 1983.

[4] Y. Jiang, K. Wang, A. Tuzhilin, and A. Fu. Mining patterns that respond to actions. *Technical Report, School of Computing Science, Simon Fraser University*, 2005.

[5] J. Kleinberg, C. Papadimitriou, and P. Raghavan. A microeconomic view of data mining. *Journal of Knowledge Discovery and Data Mining*, 2:311–324, 1998.

[6] G. Piatesky-Shapiro and C. J. Matheus. The interestingness of deviations. In *AAAI-94 Workshop on KDD*.

[7] J. Quinlan. C4.5: Programs for machine learning. In *Morgan Kaufmann*, 1993.

[8] K. Wang, S. Zhou, and J. Han. Profit mining: From patterns to actions. In *EDBT*, pages 70–87, 2002.

[9] Q. Yang and J. Yin. Postprocessing decision trees to extract actionable knowledge. In *ICDM*, 2003.