Flexible Frameworks for Actionable Knowledge Discovery

Longbing Cao, *Senior Member*, *IEEE*, Yanchang Zhao, *Member*, *IEEE*, Huaifeng Zhang, *Member*, *IEEE*, Dan Luo, Chengqi Zhang, *Senior Member*, *IEEE*, and E.K. Park

Abstract—Most data mining algorithms and tools stop at the mining and delivery of patterns satisfying expected technical interestingness. There are often many patterns mined but business people either are not interested in them or do not know what follow-up actions to take to support their business decisions. This issue has seriously affected the widespread employment of advanced data mining techniques in greatly promoting enterprise operational quality and productivity. In this paper, we present a formal view of *actionable knowledge discovery* (AKD) from the system and decision-making perspectives. AKD is a *closed optimization problem-solving process* from problem definition, framework/model design to actionable pattern discovery, and is designed to deliver operable business rules that can be seamlessly associated or integrated with business processes and systems. To support such processes, we correspondingly propose, formalize, and illustrate four types of generic AKD frameworks: *Postanalysis-based AKD*, *Unified-Interestingness-based AKD*, *Combined-Mining-based AKD*, and *Multisource Combined-Mining-based AKD* (MSCM-AKD). A real-life case study of MSCM-based AKD is demonstrated to extract debt prevention patterns from social security data. Substantial experiments show that the proposed frameworks are sufficiently general, flexible, and practical to tackle many complex problems and applications by extracting actionable deliverables for instant decision making.

Index Terms—Data mining, domain-driven data mining (D^3M), actionable knowledge discovery, decision making.

1 Introduction

In general, data mining (or KDD) algorithms and tools only focus on the discovery of patterns satisfying expected technical significance. The identified patterns are then handed over to business people for further employment. Surveys of data mining for business applications following the above paradigm in various domains [10] have shown that business people cannot effectively take over and interpret the identified patterns for business use. This may result from several aspects of challenges besides the dynamic environment enclosing constraints [4]. 1) There are *often many* patterns mined but they are not informative and transparent to business people who do not know which are *truly interesting and operable* for their businesses. 2) A large proportion of the identified patterns may be either *common-sense or of no particular interest* to business needs. Business people feel confused by *why* and *how* they should care about those findings. 3) Further, business people often do not know, and are also not informed, *how to interpret* them and *what straightforward actions can be taken* on them to support business decision-making and operation.

- *L. Cao, Y. Zhao, H. Zhang, D. Luo, and C. Zhang are with the Faculty of Engineering and Information Technology, University of Technology, Sydney, PO Box 123 Broadway, New South Wales 2007, Australia. E-mail: {lbcao, yczhao, hfzhang, dluo, chengqi}@it.uts.edu.au.*
- *E.K. Park is with the Department of Computer Science Electrical Engineering, University of Missouri at Kansas City, 5100 Rockhill Road, RFH RM 560B, Kansas City, MO 64110. E-mail: ekpark@umkc.edu.*

Manuscript received 14 July 2008; revised 5 Feb. 2009; accepted 20 May 2009; published online 4 June 2009.
Recommended for acceptance by C. Ling.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-07-0357.
Digital Object Identifier no. 10.1109/TKDE.2009.143.

The above issues inform us that there is a large gap [22], [15], [14], [12] between academic deliverables and business expectations, as well as between data miners and business analysts. Therefore, it is critical to develop effective methodologies and techniques to narrow down and bridge the gap. Clearly, there is a need to develop general, effective, and practical methodologies for actionable knowledge discovery (AKD).

One essential way is to develop effective approaches for discovering patterns that not only are *of technical significance* [35], but *also satisfy business expectations* [14], and further *indicate the possible actions* that can be explicitly taken by business people [1], [6]. Therefore, we need to discover *actionable knowledge* that is much more than simply satisfying predefined technical interestingness thresholds. Such actionable knowledge is expected to be delivered in *operable* forms for transparent business interpretation and action taking.

It has been increasingly recognized that traditional data mining is facing crucial problems in satisfying user preferences and business needs. For example, research work has been reported on developing *actionable* interestingness [14], [1] and subjective interestingness such as profit mining [37] to extract more interesting patterns, and on enhancing the interpretation of findings through explanation [39]. However, the nature of the existing work on actionable interestingness development is mainly technical-significance-oriented, e.g., by developing *alternative* and *subjective* metrics. The critical problem to a great extent comes from the oversimplification of complex domain factors surrounding business problems, the universal focus on algorithm innovation and improvement, and the little attention taken of enhancing KDD system infrastructure to tackle organizational and social complexities in real-world applications.

Fundamental work on AKD is therefore necessary to cater for critical elements in real-world applications such as

environment, expert knowledge, and operability. This is related to, but much beyond, algorithm innovation and performance improvement. To this end, AKD must cater for domain knowledge [40] and environmental factors, balance *technical significance* and *business expectations* from both *objective* and *subjective* perspectives [14], and support automatically *converting patterns into deliverables* in business-friendly and operable forms such as actions or rules. It is expected that the AKD deliverables will be business-friendly enough for business people to interpret, validate, and action, and that they can be *seamlessly embedded* into business processes and systems. If that is the case, data mining has good potential to lead to productivity gain, smarter operation, and decision making in business intelligence. Such efforts actually aim at the KDD paradigm shift from traditionally *technical interestingness-oriented and data-centered hidden pattern mining* toward *business-use-oriented and domain-driven actionable knowledge discovery* [12].

Relevant preliminary work on AKD mainly addresses specific algorithms and tools for the filtration, summarization, and postprocessing [42] of learned rules. There is a need to develop general AKD frameworks that can cater for critical elements in the real world and can also be instantiated into various approaches for different domain problems. To the best of our knowledge, very limited research work has been reported in this regard.

This paper features the definition and development of several general AKD frameworks from the system viewpoint, which follow the methodology of Domain-Driven Data Mining (DDDM, or D^3M for short) [12], [14], [15], [6], [10]. Our focus is on introducing their concepts, principles, and processes that are new, effective to AKD, flexible, and practical. Such frameworks are necessary and useful for implementing real-world data mining processes and systems, but are often ignored in the current KDD research.

The main contributions of this work are:

1. stating the AKD problem from system and micro-economy perspectives to define fundamental concepts of actionability and actionable patterns,
2. defining knowledge actionability by highlighting both technical significance and business expectations that need to be considered, balanced, and/or aggregated in AKD,
3. proposing four general frameworks to facilitate AKD, and
4. demonstrating the effectiveness and flexibility of the proposed frameworks in tackling real-life AKD.

The main ideas of D^3M-based AKD and the four frameworks are as follows: Table 1 lists key concepts and their abbreviations used in this paper.

1. *PA-AKD*: A two-step AKD process. First, *general patterns* are mined based on technical significance; the learned patterns are then filtered and summarized in terms of business expectations, and further are converted into opertionalizable business rules for business people's use.
2. *UI-AKD*: AKD develops unified interestingness that aggregates and balances both technical significance and business expectation. The mined patterns are further converted into deliverables based on domain knowledge and semantics.

TABLE 1
Key Concepts

Notations	Explanations
AKD	Actionable Knowledge Discovery
PA-AKD	Post analysis-based AKD
UI-AKD	Unified interestingness based AKD
CM-AKD	Combined mining-based AKD
MSCM-AKD	Multi-source + combined mining-based AKD
P	$P = \{p_1, \ldots, p_u\}$ is a pattern set
$\widetilde{P}$	$\widetilde{P} = \{\tilde{p_1}, \tilde{p_2}, \cdots\}$ is an actionable pattern set
$\widetilde{R}$	$\widetilde{R} = \{\tilde{r_1}, \tilde{r_2}, \cdots\}$ is a business rule set
$Int(p)$	Pattern p's interestingness
$act(p)$	Pattern p's actionability

3. *CM-AKD*: A multistep pattern mining on the data set in terms of a certain combination strategy. The mined patterns in a step may be fed into another mining procedure to guide its feature construction and corresponding pattern mining. Individual patterns identified from each step are then merged into final deliverables based on merger strategy, domain knowledge, and/or business needs.
4. *MSCM-AKD*: Handles AKD in either multiple data sources or large quantities of data. One of the data sets is selected for mining initial patterns. Some learned patterns are then selected to guide feature construction and pattern mining on the next data set(s). The iterative mining stops when all data sets are mined, and the corresponding patterns are then merged/summarized into actionable deliverables.

The above frameworks have been tested in several domains, such as on social security transactional data [13], [43], [18] and exchange orderbook data [9], [8]. Substantial experiments show that these frameworks are effective and flexible for extracting actionable knowledge in complex real-world situations, and assist data mining practitioners with catering for their business requirements, needs, and decision-making actions on the findings and deliverables in the business environment.

2 Related Work

Actionable knowledge discovery is critical in promoting and releasing the productivity of data mining and knowledge discovery for smart business operations and decision making. Both SIGKDD and ICDM panelists pointed it out as one of the great challenges in developing the next-generation KDD methodologies and systems [3], [19]. In recent years, some relevant work has been emerging.

The term "actionability" measures *the ability of a pattern to suggest a user to take some concrete actions to his/her advantage in the real world*. It mainly measures the ability to suggest business decision-making actions. Existing efforts in the development of effective interestingness metrics are basically on developing and refining *objective technical* interestingness metrics ($t_o()$) [21], [25]. They aim to capture the *complexities of pattern structure and statistical significance*. Other work appreciates *subjective technical* measures ($t_s()$) [29], [31], [34], which also recognize *to what extent a pattern is of interest to particular user preferences*. For example, *probability-based belief* is used to describe user confidence of

unexpected rules [31]. There is very limited research on developing business-oriented interestingness, for instance, profit mining [37].

The main limitations for the existing work on interestingness development lie in a number of aspects. Most work is on developing alternative interest measures focusing on technical interestingness only [30]. Emerging research on general business-oriented interestingness is isolated from technical significance. A question to be asked is "what makes interesting patterns actionable in the real world?" For that, knowledge actionability needs to pay equal attention to both *technical* and *business-oriented* interestingness from both *objective* and *subjective* perspectives [14].

With regard to AKD approaches, the existing work mainly focuses on developing postanalysis techniques to filter/prune rules [27], reduce redundancy [26], and summarize learned rules [27], as well as on matching against expected patterns by similarity/difference [28]. In post analysis, a recent highlight is to *extract actions from learned rules* [38]. A typical effort on learning action rules is to split attributes into "hard/soft" [38] or "stable/flexible" [36] to extract actions that may improve the loyalty or profitability of customers. Other work is on action hierarchy [1]. Some other approaches include a combination of two or more methods, for instance, *class association rules* (or *associative classifier*) that build classifiers on association rules ($A \rightarrow C$) [23]. In [23], external databases are input into characterizing the item sets. In [32], clustering is used to reduce the number of learned association rules. Some other work is on the transformation from data mining to knowledge discovery [20], and developing a general KDD framework to fit more factors into the KDD process [39].

Regarding the existing work, we have the following comments: First, existing work often stops at pattern discovery mainly based on technical significance and interestingness. As a result, the summarized "actions" do not reflect the genuine expectations of business needs, and therefore, cannot support decision making. Second, most of the existing postanalysis and postmining focuses on association rules or their combination with some specific methods. This limits the actionability of learned actions and the generalization of proposed approaches for AKD.

To tackle the challenges in real-world KDD and bridge the gap, it is necessary to take a critical view of KDD, such as from microeconomic [24] and system perspectives, and develop workable methodologies and frameworks to support AKD. To this end, D^3M [15], [12] is proposed to involve ubiquitous intelligence in the AKD process toward the delivery of operable business rules. With the D^3M, this paper proposes four types of general frameworks that can be customized to extract actionable deliverables satisfying both technical significance and business expectations. Additionally, rather than addressing the whole KDD process as it did in some of related works, this paper only focuses on method/algorithm frameworks toward AKD.

3 A System View of Actionable Knowledge Discovery

Real-world data mining is a complex problem-solving system. From the view of systems and microeconomy, the endogenous character of AKD determines that it is an optimization problem with certain objectives under a particular environment.

Let DB be a database related to business problems Ψ, $X = \{x_1, x_2, \ldots, x_L\}$ be the set of items in DB, where x_l ($l = 1, \ldots, L$) be an item set, and the number of attributes in DB be S. Suppose $E = \{e_1, e_2, \ldots, e_K\}$ denotes the environment set, where e_k represents a particular environment setting for AKD. Further, let $M = \{m_1, m_2, \ldots, m_N\}$ be the data mining method set, where m_n ($n = 1, \ldots, N$) is a method. For method m_n, suppose its identified pattern set $P^{m_n} = \{p_1^{m_n}, p_2^{m_n}, \ldots, p_U^{m_n}\}$ includes all patterns discovered in DB, where $p_u^{m_n}$ ($u = 1, \ldots, U$) is a pattern discovered by m_n.

In the real world, data mining is a problem-solving process (R) from business problems Ψ (with problem status τ) to problem-solving solutions Φ:

$$R : \Psi(\tau_1) \rightarrow \Phi(\tau_2). \tag{1}$$

From the modeling perspective, such an AKD-based problem-solving process is a state transformation from the source data $DB(\Psi \rightarrow DB)$ to the resulting pattern set $P(\Phi \rightarrow P)$.

Definition 1 (Actionable patterns). *Let*

$$\widetilde{P}^{m_n} = \{\tilde{p_1}^{m_n}, \tilde{p_2}^{m_n}, \ldots, \tilde{p_U}^{m_n}\}$$

be an Actionable Pattern Set *mined by method m_n for the given problem Ψ (its data DB), in which $\tilde{p_u}^{m_n}$ is* actionable *for the problem solving if it satisfies the following conditions:*

a. *$t_i(\tilde{p_u}) \geq t_{i,0}$; "$\geq$" indicates the pattern $\tilde{p_u}$ can beat technical interestingness t_i with threshold $t_{i,0}$;*
b. *$b_i(\tilde{p_u}) \geq b_{i,0}$; "$\geq$" indicates the pattern $\tilde{p_u}$ can beat business interestingness b_i with threshold $b_{i,0}$;*
c.

$$R : \tau_1 \overset{A(\tilde{p_u}^{m_n})}{\longrightarrow} \tau_2;$$

the pattern can support business problem solving by taking action A, and correspondingly, transform the problem status from the initially nonoptimal status τ_1 to the greatly improved τ_2.

Definition 2 (Actionable knowledge discovery). *AKD is an iterative optimization process toward the actionable pattern set $\widetilde{P}$, considering surrounding business environment and problem states.*

$$\begin{aligned} AKD^{e,\tau,m\in M} : DB &\overset{e,\tau,m_n}{\longrightarrow} P^{m_n} \\ &\longrightarrow O_{p\in P}^{e,\tau,m\in M} Int(p) \\ &\longrightarrow \widetilde{P}, \end{aligned} \tag{2}$$

where $P = P^{m_1} U P^{m_2}, \ldots, U P^{m_n}$, $Int(.)$ is the interestingness evaluation function, and $O(.)$ is the optimization function to extract those $\tilde{p} \in \widetilde{P}$ when $Int(\tilde{p})$ can beat a given benchmark.

For a pattern p, $Int(p)$ can be further measured in terms of *technical interestingness* ($t_i(p)$) and *business interestingness* ($b_i(p)$) [14].

$$Int(p) = I(t_i(p), b_i(p)), \tag{3}$$

where $I(.)$ "aggregates" the contributions of all particular aspects of interestingness. Further, $Int(p)$ can be described in terms of *objective* (o) and *subjective* (s) factors from both *technical* (t) and *business* (b) perspectives.

$$\begin{aligned} Int(p) &= I(t_o(p), t_s(p), b_o(p), b_s(p)) \\ &\rightarrow t_o(x,p) \wedge t_s(x,p) \wedge b_o(x,p) \wedge b_s(x,p), \end{aligned} \tag{4}$$

where $t_o()$ is objective technical interestingness, $t_s()$ is subjective technical interestingness, $b_o()$ is objective business interestingness, and $b_s()$ is subjective business interestingness, and $I \rightarrow \text{`}\wedge\text{'}$ indicates the "aggregation."

In general, $t_o()$, $t_s()$, $b_o()$, and $b_s()$ of practical applications can be regarded as independent of each other. With their normalization (expressed by $\hat{}$), we can get:

$$\begin{aligned} Int(p) &\rightarrow \hat{I}(\hat{t_o}(), \hat{t_s}(), \hat{b_o}(), \hat{b_s}()) \\ &= \alpha\hat{t_o}() + \beta\hat{t_s}() + \gamma\hat{b_o}() + \delta\hat{b_s}(), \end{aligned} \tag{5}$$

where α, β, γ and δ are weights respectively.

So, the AKD optimization problem is as follows:

$$\begin{aligned} AKD^{e,\tau,m\in M} &\longrightarrow O_{p\in P}(Int(p)) \\ \rightarrow O(\alpha\hat{t_o}()) \wedge O(\beta\hat{t_s}()) &\wedge O(\gamma\hat{b_o}()) \wedge O(\delta\hat{b_s}()). \end{aligned} \tag{6}$$

Definition 3 (Actionability of a pattern). *The* actionability *of a pattern p is measured by $act(p)$:*

$$\begin{aligned} act(p) &= O_{p\in P}(Int(p)) \\ \rightarrow O(\alpha t_o(p)) \wedge O(\beta t_s(p)) &\wedge O(\gamma b_o(p)) \wedge O(\delta b_s(p)) \\ \rightarrow t_o^{act} \wedge t_s^{act} &\wedge b_o^{act} \wedge b_s^{act} \rightarrow t_i^{act} \wedge b_i^{act}, \end{aligned} \tag{7}$$

where t_o^{act}, t_s^{act}, b_o^{act}, and b_s^{act} measure the respective actionable performance.

For example, actionable frequent trading pattern mining [16], [8], [9] considers the satisfaction of *support*, *confidence*, as well as business performance like *sharpe ratio*. Suppose they are independent. We then expect an actionable trading pattern to concurrently satisfy these metrics in a maximal manner.

Due to the inconsistency often existing in different aspects, we often find identified patterns only fitting in one of the following subsets:

$$\begin{aligned} Int(p) \rightarrow \{ &\{t_i^{act}, b_i^{act}\}, \{\neg t_i^{act}, b_i^{act}\}, \\ &\{t_i^{act}, \neg b_i^{act}\}, \{\neg t_i^{act}, \neg b_i^{act}\}\}, \end{aligned} \tag{8}$$

where "$\neg$" indicates unsatisfactory.

However, in real-world mining, as we know, it is very challenging to find the most actionable patterns that are associated with both "optimal" t_i^{act} and b_i^{act}. Quite often, a pattern with significant $t_i()$ is associated with unconfident $b_i()$. Contrarily, patterns with low $t_i()$ are often associated with confident $b_i()$. Clearly, AKD targets patterns confirming the relationship $\{t_i^{act}, b_i^{act}\}$.

Therefore, it is necessary to deal with such possible conflict and uncertainty among respective interestingness elements. However, it is something of an art form and needs to involve domain knowledge and domain experts to tune thresholds and balance differences between $t_i()$ and $b_i()$. Another issue is to develop techniques to balance and combine all types of interestingness metrics to generate uniform, balanced, and interpretable mechanisms for measuring knowledge deliverability and extracting and selecting resulting patterns. A reasonable way is to balance both sides toward an acceptable tradeoff. To this end, we need to develop interestingness aggregation methods, namely the *I-function* (or "$\wedge$") to aggregate all elements of interestingness. In fact, each of the interestingness categories may be instantiated into more than one metric. Their "aggregation" does not mean the essential combination into a single supermeasure, rather indicating the satisfaction of all respective components during the AKD process if possible. They could be checked at the same time or during the AKD processes. There could be several methods of doing the aggregation, for instance, empirical methods such as business-expert-based voting, or more quantitative methods such as multiobjective optimization methods.

Besides the measurement, knowledge actionability also needs to cater for the semantic aspect of the identified actionable patterns. This is particularly important in deploying the patterns. Briefly speaking, the conversion from an identified pattern to a business rule can follow a BusinessRule Model [12] defined in OWL-S [5]. To describe a business rule, it is necessary to specify:

- Object: On what object(s), the actions are taken, with predicates to limit the range;
- Condition: Under what situations, the actions can be taken on the objects, with predicates to specify the conditions; and
- Operation: What actions are to be taken on the objects, with predicates to deliver the specific decision-making activities.

Subsequent to the following specification, actionable patterns are converted into business rules as a form of deliverable, which not only enhances interpretation but also indicates what actions can be taken on what objects under what conditions.

/*BusinessRule Specification*/

$< business\_rule >::= < object >^{+} < condition >^{*}$
$< operation >^{+}$
$< object >::= (All|Any|Giveni|\ldots)$
$< condition >::= (satisfy|related|and|\ldots)$
$< operation >::= (Alert|Action|\ldots)$

4 Actionable Knowledge Discovery Frameworks

4.1 Postanalysis-Based AKD: PA-AKD

PA-AKD is a two-step pattern extraction and refinement exercise. First, generally interesting patterns (which we call "general patterns") are mined from data sets in terms of technical interestingness $(t_o(), t_s())$ associated with the algorithms used. Further, the mined general patterns are pruned, distilled, and summarized into operable business rules (embedding actions) (which we call "deliverables") in terms of domain-specific business interestingness $(b_o(), b_s())$ and involving domain (Ω_d) and meta (Ω_m) knowledge. Fig. 1 illustrates the PA-AKD.

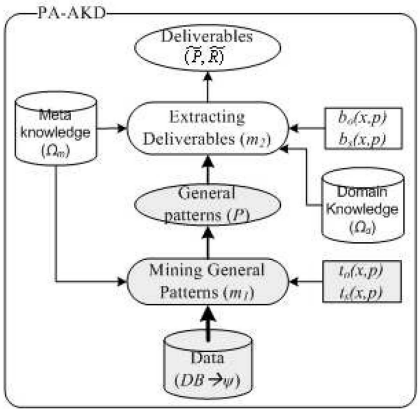


Fig. 1. Postanalysis-based AKD (PA-AKD) approach.

Based on the system view developed for AKD, PA-AKD is a two-step optimization problem that can be expressed as follows:

$$PA-AKD: DB \overset{e,t_i(),m_1}{\longrightarrow} P \overset{e,b_i(),m_2,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P}, \widetilde{R}. \quad (9)$$

The following pseudocode describes the PA-AKD:

FRAMEWORK 1: Post Analysis-based AKD (PA-AKD)
INPUT: target data set DB, business problem Ψ, and thresholds ($t_{o,0}$, $t_{s,0}$, $b_{o,0}$ and $b_{s,0}$)
OUTPUT: actionable patterns $\widetilde{P}$ and operable business rules $\widetilde{R}$
Step 1: Extracting general patterns P;
  FOR $n = 1$ to N
    Develop modeling method m_n with technical interestingness $t_i()$ (i.e., $t_o(), t_b()$);
    Employ method m_n on DB and environment e;
    Extract the general pattern set P^{m_n};
  ENDFOR
Step 2: Extracting actionable patterns $\widetilde{P}$;
  $P = P^{m_1} U \ldots U P^{m_N}$
  FOR $j = 1$ to ($count(P)$)
    Design post-analysis method m_2 by involving domain knowledge Ω_d and business interestingness $b_i()$;
    Employ the method m' on the pattern set P as well as data set DB if necessary;
    Extract the actionable pattern set $\widetilde{P}$;
  ENDFOR
Step 3: Converting pattern $\widetilde{P}$ to business rules $\widetilde{R}$.

The key point in this framework is to utilize both domain/metaknowledge and business interestingness in postprocessing the learned patterns. In the real world, this framework can be further instantiated into varied mutations [28], [27], [38]. In fact, many existing methods, such as pruning redundant patterns, summarizing and aggregating patterns to reduce the quantity of patterns, and constructing actions on top of learned patterns, can be further enhanced by expanding the PA-AKD framework and introducing business interestingness and domain/metaknowledge into the AKD process. Cao [9] presents examples of considering domain knowledge and organizational factors in extracting actionable trading strategies in stock markets. Zhao et al. [42] collect case studies of utilizing the PA-AKD framework for extracting effective associations.

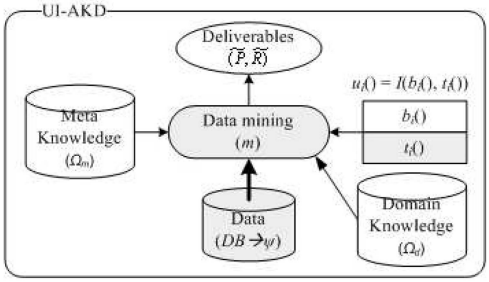


Fig. 2. Unified-interestingness-based AKD approach.

4.2 Unified-Interestingness-Based AKD: UI-AKD

As discussed in Section 3, one of the essential jobs in extracting actionable knowledge is to balance the interestingness concerns of the identified patterns from both technical and business sides. To this end, a straightforward idea is to develop *unified interestingness metrics* capturing and describing both business and technical concerns, and then to extract patterns based on this unified interestingness system. Thus, UI-AKD is based on such a unified interestingness system.

Fig. 2 shows the framework of UI-AKD. It looks just the same as the normal data mining process except for three inherent characteristics. One is the interestingness system, which combines technical interestingness ($t_i()$) with business expectations ($b_i()$) into a unified AKD interestingness system ($i()$). This unified interestingness system is then used to extract truly interesting patterns. The second is that domain knowledge (Ω_d) and environment (e) must be considered in the data mining process. Finally, the outputs are $\widetilde{P}$ and $\widetilde{R}$.

Ideally, UI-AKD can be expressed as follows:

$$UI-AKD: DB \overset{e,i(),m,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P}, \widetilde{R}. \quad (10)$$

Based on the AKD formulas addressed before, $i()$ can be further expressed as follows:

$$i() = Int() = I(t_i(), b_i()). \quad (11)$$

Very often $t_i()$ and $b_i()$ are not dependent, thus

$$i() \rightarrow \eta \hat{t_i}() + \varpi \hat{b_i}(). \quad (12)$$

Weights η and ϖ reflect the interestingness balance/tradeoff negotiated between data analysts and domain experts in terms of business problem, data, environment, and deliverable expectation. In some cases, both weights and aggregation can be fuzzy. In other cases, the aggregation may happen in a step-by-step manner. For each step, weights may be differentiated.

Patterns with $i()$ beating given thresholds (again, this must be mutually determined by stakeholders) come into the actionable pattern list. The pseudocode describing the UI-AKD process is as follows:

FRAMEWORK 2: Unified Interestingness-based AKD (UI-AKD)
INPUT: target data set DB, business problem Ψ, and thresholds ($t_{O,0}$, $t_{s,0}$, $b_{O,0}$ and $b_{s,0}$)
OUTPUT: actionable patterns $\widetilde{P}$ and business rules $\widetilde{R}$
Step 1: Extracting general patterns P;

FOR $n = 1$ to N
Design data mining method m_n by involving domain knowledge Ω_d and considering environment e against unified interestingness $i()$;
Employ the method m_n on DB given e and Ω_d;
Generate pattern set P;
ENDFOR
Step 2: Extracting deliverables $\widetilde{P}$;
Step 3: Converting $\widetilde{P}$ to business rules $\widetilde{R}$.

In practice, the combination of technical interestingness with business expectations may be implemented by various methods. An ideal situation is to generate a single formula $i()$ integrating t_i and b_i, and then to filter patterns accordingly. If such a uniform metric is not available, an alternative way is to calculate t_i and b_i for all patterns, and then rank them in terms of them, respectively. A weight-based voting (weights are determined by stakeholders) can then be taken to aggregate the two ranked lists into a unified pattern set. If there is uncertainty in merging the pattern sets, fuzzy-set-based aggregation and ranking may be helpful. Cao and Zhang [16] introduce fuzzy-set-based aggregation of trading rules in stock markets. First, trading rules are identified through Genetic Algorithms. The identified rules are then ranked in terms of technical significance and trading performance, respectively, and then fuzzified into five significance levels. The two fuzzy sets are then aggregated in terms of fuzzy aggregation rules into an integrated fuzzy set. Final top-n rules are selected from this set.

As shown in (8), in real life, there may exist potential incompatibility between technical and business interestingness values for a particular pattern. The relationships between technical interestingness metrics and business ones may be linear and/or nonlinear. In addition, the simple merger of pattern sets divided on technical and business sides may cause uncertainty. These make it very challenging to develop a unified interestingness system for AKD.

4.3 Combined-Mining-Based AKD: CM-AKD

For many complex enterprise applications, one-scan mining seems unworkable for many reasons. To this end, we propose the *Combined-Mining [18]-based AKD* framework to progressively extract actionable knowledge. Fig. 3 illustrates the CM-AKD.

CM-AKD comprises multisteps of pattern extraction and refinement on the whole data set. First, AKD is split into J steps of mining based on business understanding, data understanding, exploratory analysis, and goal definition. Second, generally interesting patterns are extracted based on technical significance ($t_i()$) (or unified interestingness ($i()$)) into a pattern subset (P_j) in step j. Third, knowledge obtained in step j is further fed into step $j+1$ or relevantly remaining steps to guide corresponding feature construction and pattern mining (P_{j+1}). Fourth, after the completion of all individual mining procedures, all identified pattern subsets are merged into a final pattern set (P) based on environment (e), domain knowledge (Ω_d), and business expectations (b_i). Finally, the merged patterns are converted into business rules as final deliverables ($\widetilde{P}$, $\widetilde{R}$) that reflect business preferences and needs.

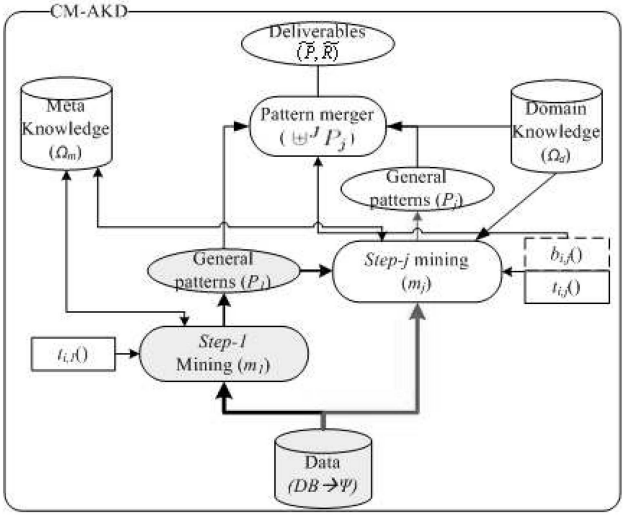


Fig. 3. Combined-mining-based AKD (CM-ADK).

CM-AKD can be formalized as follows:

$$CM-AKD: \underbrace{DB \overset{e,t_{i,j}()[i_{i,j}()],m_j,\Omega_d,\Omega_m}{\longrightarrow} \{P_j\}}_{J} \overset{e,b_{i,j}(),\uplus^J P_j,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P}, \widetilde{R}, \tag{13}$$

where $t_{i,j}$ and $b_{i,j}$ are technical and business interestingness of model m_j, and $[i_{i,j}()]$ indicates the alternative checking of unified interestingness, $\uplus^J P_j$ is the merger function, Ω_m is the metaknowledge consisting of metadata about patterns, features, and their relationships.

The CM-AKD process can be expressed as follows:

FRAMEWORK 3: Combined Mining-based AKD (CM-AKD)
INPUT: target data set DB, business problem Ψ, and thresholds ($t_{o,0}$, $t_{s,0}$, $b_{o,0}$ and $b_{s,0}$)
OUTPUT: actionable patterns $\widetilde{P}$ and operable business rules $\widetilde{R}$;
Step 1: AKD is split into J steps of mining;
Step 2: *Step-j mining*: Extracting general patterns P_j ($j = 1, \ldots, J$);
FOR $j = 1$ to J
Develop modeling method m_j with technical interestingness $t_{i,j}()$ (i.e., $t_o(), t_b()$) or unified $i_{i,j}()$
Employ method m_j on the environment e and data DB engaging meta-knowledge Ω_m;
Extract the general pattern set P_j;
ENDFOR
Step 3: *Pattern merging*: Extracting actionable patterns $\widetilde{P}$;
FOR $j = 1$ to J
Design the pattern merger functions $\uplus^J P_j$ by involving domain (Ω_d) and meta (Ω_m) knowledge, and business interestingness $b_{i,j}()$;
Employ the method $\uplus P_j$ on the pattern set P_j;
Extract the actionable pattern set $\widetilde{P}$;
ENDFOR
Step 4: Converting patterns $\widetilde{P}$ to rules $\widetilde{R}$.

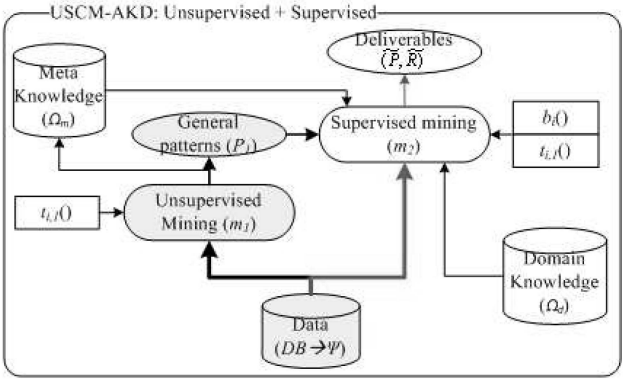


Fig. 4. Unsupervised + supervised learning-based CM-AKD (USCM-AKD).

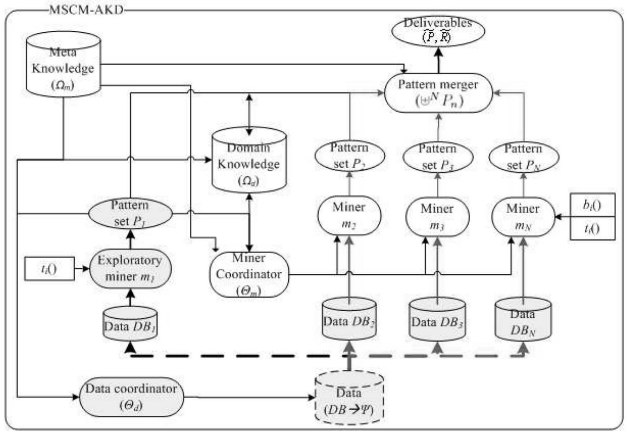


Fig. 5. Multisource + combined-mining-based AKD.

This framework can be instantiated into a few mutations by employing technical and business interestingness at various stages, and by combining miscellaneous data mining models in a multistep and iterative manner. One example is an *unsupervised + supervised learning-based CM-AKD: USCM-AKD*. As shown in Fig. 4, the USCM-AKD first deploys an *unsupervised learning method* to mine general patterns in terms of technical interestingness $t_{i,j}()$ associated with the methods m_1. New variables triggered by the unsupervised learning process are added into the metaknowledge base Ω_m. The original data set is then filtered, transformed, and/or aggregated, guided by knowledge obtained in previous learning to generate a transformed data set for further mining. The learned patterns P_1 are then used to guide the extraction of deliverables $\widetilde{P}$ and $\widetilde{R}$ by a *supervised learning method* m_2 on the transformed data set concerning both technical ($t_i()$) and business ($b_i()$) interestingness.

An example is to develop sequential classifiers [41]. First, we mine for the most discriminative sequential patterns, in which an aggressive strategy is used to select a small set of sequential patterns. Second, pattern pruning and serial coverage test are done on the mined patterns. Those patterns passing the serial test are used to build the subclassifiers on the first level of the final classifier. Third, the training samples that cannot be covered are fed back to the sequential pattern mining procedure with updated parameters. This process continues until the predefined thresholds are reached or all samples are covered. Patterns generated in each loop form the subclassifier on each level of the final classifiers.

In addition, the CM-AKD framework can be further joined with the PA-AKD approach to generate a more comprehensive framework: *Combined Mining + Postanalysis-based AKD* (CMPA-AKD). In the CMPA-AKD approach, multistep mining may be conducted by checking technical interestingness only, and leaving the checking of business interests to the postanalysis component. In some other cases, multistep mining is based on unified interestingness while pattern merging is conducted during postanalysis.

4.4 Multisource + Combined-Mining-Based AKD: MSCM-AKD

Enterprise applications often involve multiple-subsystems-based and heterogeneous data sources that cannot be integrated, or are too costly to do so. Another common situation is that the data volume is so large that it is too costly to scan the whole data set. Mining such complex and large volumes of data challenges existing data mining approaches. To this end, we propose a *Multisource + combined-mining-based AKD* framework. Fig. 5 shows the idea of MSCM-AKD.

MSCM-AKD discovers actionable knowledge either in multiple data sets or data subsets ($DB_1, \ldots, DB_N$) through partition. First, based on domain knowledge, business understanding, and goal definition, one of the data sets or certain partial data (say DB_n) is selected for mining exploration (m_1). Second, the exploration results are used to guide either data partition or data set management through a *data coordinator agent* Θ_{db} (coordinating data partition and/or data set/feature selection in terms of iterative mining processes, see more from AMII-SIG[1] regarding agents in data mining), and to design strategies for managing and conducting *parallel pattern mining* on each data set or subset and/or *combined mining* [18] on relevant remaining data sets. The deployment of method m_n, which could be either in parallel or combined, is determined by data/business understanding and objectives. Third, after the mining of all data sets, patterns P_n identified from individual data sets are merged ($\uplus^N P$) and extracted into final deliverables ($\widetilde{P}$,$\widetilde{R}$).

MSCM-AKD can be expressed as follows:

$$MSCM - AKD: \underbrace{DB_n[DB \xrightarrow{\otimes} DB_n] \xrightarrow{e,t_{i,n}()[u_{i,n}()],m_n,\Omega_m} \{P_n\}}_{N} \xrightarrow{e,b_{i,n}(),\uplus^N P_n,\Omega_d,\Omega_m} \widetilde{P}, \widetilde{R}, \quad (14)$$

where $t_{i,n}$ and $b_{i,n}$ are technical and business interestingness of model m_n on data set/subset n, and $[i_{i,n}()]$ indicates the alternative checking of unified interestingness as in UI-AKD, $\uplus^N P_n$ is the merger function, $\otimes$ indicates the data partition if the source data needs to be split.

The MSCM-AKD process is expressed as follows:

FRAMEWORK 4: Multi-Source + Combined Mining Based AKD (MSCM-AKD)

INPUT: target data sets DB, business problem Ψ, and thresholds ($t_{o,0}$, $t_{s,0}$, $b_{o,0}$ and $b_{s,0}$)

1. www.agentmining.org.

OUTPUT: actionable patterns $\widetilde{P}$ and business rules $\widetilde{R}$
Step 1: Identify or partition whole source data into N data sets DB_n $(n = 1, \ldots, N)$;
Step 2: *Data Set-n mining*: Extracting general patterns P_n on data set/subset DB_n;
  FOR $l = n$ to (N)
    Develop modeling method m_n with technical interestingness $t_{i,n}()$ (i.e., $t_o(), t_b()$) or unified $i_{i,n}()$
    Employ method m_n on the environment e and data DB_n engaging meta-knowledge Ω_m;
    Extract the general pattern set P_n;
  ENDFOR
Step 3: *Pattern merger*: Extracting actionable patterns $\widetilde{P}$;
  FOR $l = n$ to N
    Design the pattern merger functions $\uplus^N P_n$ to merge all patterns into $\widetilde{P}$ by involving domain and meta knowledge Ω_d and Ω_m, and business interestingness $b_i()$;
    Employ the method $\uplus P_n$ on the pattern set P_n;
    Extract the actionable pattern set $\widetilde{P}$;
  ENDFOR
Step 4: Converting patterns $\widetilde{P}$ to business rules $\widetilde{R}$.

The MSCM-AKD framework can also be instantiated into a number of mutations. For instance, for a large volume of data, MSCD-AKD can be instantiated into *data partition + unsupervised + supervised-based AKD* by integrating data partition into combined mining. An example is as follows: First, the whole data set is partitioned into several data subsets based on the data/business understanding and domain knowledge jointly by data miners and domain experts, say data sets 1 and 2. Second, an *unsupervised learning* method is used to mine one of the preference data sets, say data set 1. Some of the mined results are then used to design new variables for processing the other data set. *Supervised learning* is further conducted on data set 2 to generate actionable patterns by checking both technical and business interestingness. Finally, the individual patterns mined from both data subsets are combined into deliverables.

In Section 6, we introduce a real-life case study in extracting deliverables for government debt prevention. The deliverables take forms of either combining arrangement activities initiated by government officers with repayment activities conducted by debt-associated customers, or combining demographics patterns with arrangement-repayment activity sequential patterns. Government officers who receive such knowledge feel more comfortable in applying them to their routine processes and rules to prevent debts.

5 Discussions

The D^3M views real-world AKD as a *closed optimization* system. "Closed" indicates the problem solving is a *closed process* starting from business problem definition and ending with operable business rules fed back into business problem solving. "Optimization" means that the AKD targets *optimal* solutions namely *actionable and operable* patterns and produces operable business rules. Based on the above principles, the proposed AKD frameworks present many promising characteristics.

First, the proposed AKD frameworks are *general and flexible*, and can cover many common problems and applications. Basically, they enclose many key features that are critical for offering flexibility and expandability to handle practical challenges in mining complex enterprise applications. These include catering for the organizational environment and domain knowledge (all frameworks care about domain knowledge and environment, and the interestingness system has been expanded to facilitate business concerns), mining multiple data sources [32] and large volumes of data (see MSCM-AKD), postprocessing the learned patterns as per business needs (see PA-AKD), and supporting multistep and combined mining (see CM-AKD and MSCM-AKD), as well as closed data mining (by delivering operable business rules, see Section 4.4).

These general features, on one hand, can be instantiated into many concrete approaches. As we discussed in introducing each framework, they can be instantiated into various mutations. For instance, the postanalysis-based approach can be embodied on top of association mining, clustering, and classification to extract actionable associations, clusters, and classes. The *unsupervised + supervised* learning process can be instantiated into approaches such as *association + classification and clustering + classification*. On the other hand, they can fit into requirements and constraints in many practical applications, for instance, analyzing rare but significant linkages isolated in multiple organizations' data, and dealing with complex data structure mixing heterogeneous and distributed data sources.

Second, the AKD frameworks are *effective and workable* for extracting knowledge that can be taken over by business people for instant decision making. There are three key factors contributing to effectiveness and workability. 1) The extraction of patterns is based on both technical significance and business expectations, and as a result, they are of business interest. 2) The frameworks support mining complex data and knowledge in the real world. 3) The delivery of business rules as the mining deliverables make them operable and bridge the gap between the deliverables and business needs.

In addition, the deep study of D^3M-based AKD has disclosed many open issues and broad prospects in developing next-generation KDD, i.e., knowledge processing and decision-support methodologies, techniques, and systems for real-world applications. The issues are:

- *AKD as a closed problem-solving system*: current KDD is weak in feeding back the resulting solutions to business problems. The extraction of operable business rules presents a feasible way to achieve such an objective. Further work is necessary in defining universal representation and modeling languages for such a purpose.
- *AKD problem-solving environment*: KDD researchers increasingly recognize the significance of understanding, involving, and tackling "environment" factors in AKD modeling and presenting deliverables. Environmental factors refer to the surroundings related to human beings, business process, policies, rules, workflow, organizational factors, and networking factors [7]. Exemplary explanation can be found in [16], [8], [9]. With the system view, it is necessary to develop techniques to describe, represent, and involve environmental elements and

facilitate the interaction between an AKD system and its environment.

- *Ubiquitous intelligence surrounding and assisting in AKD problem-solving systems*: AKD inevitably engages human intelligence, domain intelligence, network intelligence, and organizational and social intelligence. Appropriate metasynthesis [17] of such intelligence can greatly enhance the power of AKD in handling complex data and applications.
- *Representation and integration of ubiquitous intelligence in AKD*: it is necessary to develop effective mechanisms to represent, transform, map, search, coordinate, and integrate such ubiquitous intelligence in AKD systems.
- *Actionability checking*: this involves what actionability system is, and how to evaluate actionability. In AKD, a critical issue is what metrics and at what stage of AKD process should actionability be checked. Appropriated combination strategies may be necessary for checking actionability from both technical and business perspective in terms of objective and subjective aspects. In practice, identifying/pruning generally technical-interesting patterns may be conducted first followed by checking business interestingness of identified patterns and accordingly pruning them.
- *Status optimization and transferability*: the system status transformation by taking a decision-making action is subject to many factors and constraints such as cost and transferability. It is essential to consider them in cost-sensitive status optimization and transformation. For that, for example, the corresponding mechanisms and metrics for cost-benefit analysis can be helpful.

The effectiveness, general capability, flexibility, and adaptability of the proposed AKD frameworks have been tested and demonstrated in several problem domains, for example, mining social security data for debt recovery and prevention [13], [43], [18], and identifying actionable trading strategies [9], and discovering exceptional trading behavior in capital market microstructure data [16], [8], [9]. Due to space limitation, we cannot illustrate these examples one by one. However, interested readers can access more details from the references. In the following section, we illustrate a case study using the MSCM-AKD framework for discovering actionable combined patterns in social security data. It consists of customer demographic components and customer transactional activities in social security areas for government officers to prevent customer debt.

6 Case Study: Mining Actionable Combined Patterns in Social Security Data

6.1 Problem Overview

Social security data are widely seen in welfare states, but they have rarely been analyzed. The data consist of customer demographics, government overpayment (debt) information, activities such as government arrangements for debtors' payback agreed by both parties, and debtors' repayment information. Such data contain important information about the experience and performance of government service objectives and social security policies, and may include evidence and indicators for recovering, detecting, preventing, and predicting debt occurrences.

The analysis of social security data needs to concern environmental factors. This involves government service policies, business rules, debt management processes and rules, and debt arrangement and repayment activities and rules, etc. We cater for them in data extraction such as involving arrangement and repayment activity data, data preparation such as policy-sensitive seasonal effect analysis and activity independence analysis, and pattern structures such as combined patterns [18].

Activity mining [11], [13] has been proposed based on AKD frameworks to identify patterns of high impact-oriented customer behavior and customer-government officer contacts that are likely to be correlated to debts. Some of the methods are based on a UI-AKD framework by developing interestingness metrics measuring both technical significance and business impact of the patterns, for instance, in identifying *sequential impact-contrasted activity patterns* where P is frequently associated with both patterns $P \rightarrow T$ and $P \rightarrow \bar{T}$ in separated data sets, and *sequential-impact-reversed activity patterns* in which both $P \rightarrow T$ and $PQ \rightarrow \bar{T}$ are frequent. This section briefly illustrates the MSCM-AKD framework in identifying combined associations and combined association clusters for debt prevention, by following the approach of combined mining [18]. The resulting combined patterns consist of items from two heterogeneous data sets in terms of domain knowledge, business policies, a new interestingness system, and discussions with domain experts. For more details on theoretical analysis and implementation, please refer to [18].

6.2 Combined Associations and Association Clusters

Based on the MSCM-AKD framework, combined associations and association clusters are defined as follows:

Definition 4 (Combined associations). *A combined association rule p is in the form of $x + y \rightarrow c$, where x and y are different item sets from two heterogeneous data sets, and c is a target item or class.*

For example, x is a demographic item set, y is a transactional item set, and c is the class of a customer.

The combined associations can be organized into rule clusters by putting similar or related rules together, which can provide more useful knowledge than separated rules.

Definition 5. (Combined association cluster). *A combined association cluster P is a set of combined associations with the same x (or y) but different y (or x)*

$$P_1 : \begin{cases} x + y_1 \rightarrow c_{k_1} \\ x + y_2 \rightarrow c_{k_2} \\ \cdots \\ x + y_m \rightarrow c_{k_m} \end{cases}, \qquad P_2 : \begin{cases} x_1 + y \rightarrow c_{l_1} \\ x_2 + y \rightarrow c_{l_2} \\ \cdots \\ x_n + y \rightarrow c_{l_n}, \end{cases} \tag{15}$$

where P_1 and P_2 are two combined clusters. The rules in cluster P_1 have the same x but different y, which makes them associated with various results c. By contrast, the rules in P_2 have the same y but different x.

Here "combined" means: 1) each single rule consists of item sets from different data sets and 2) each rule cluster is

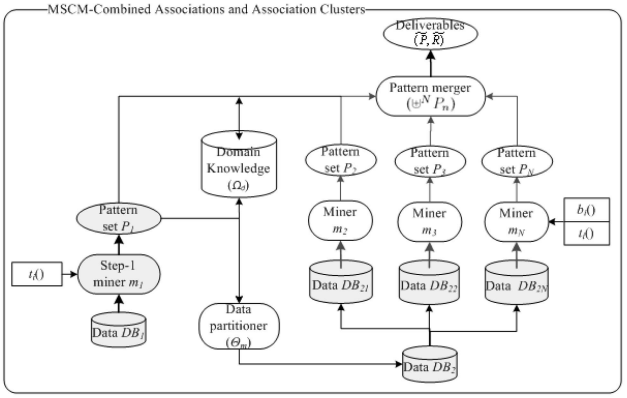


Fig. 6. MSCM-AKD-based framework for mining combined associations and association clusters.

composed of two or more related local rules identified in individual data sets. The process of mining combined associations and association clusters in social security data is shown in Fig. 6, where DB_1 is demographic data, DB_2 is transactional data.

Mining Combined Associations and Association Clusters
INPUT: target data sets DB_1 and DB_2
OUTPUT: Combined association rules and association clusters

Step-1 mining: Mining frequent patterns on the whole population in data set DB_1. Select the top-m frequent demographic patterns discovered p_i $(i = 1, 2, \ldots, m)$;

Step-2 partition: Partitioning the data set DB_2 into n sub-data sets DB_{2n} $(n = 1, \ldots, m)$ based on the identified top demographic patterns.

Step-3 mining: Mining associations on data set DB_{2n} in terms of the groups of people identified in *Step-1*. Identifying $\{Q_n = y_{nj} \rightarrow c_{nj}\}$ $(n = 1, \ldots, N;$ $j = 1, \ldots, J)$, the top-J frequent patterns discovered from DB_{2n}.

Step-4 merging patterns: Merging results discovered in the relevant groups into combined patterns like: 1) $x_1 + y_1 \rightarrow c_1$ and $x_1 + y_2 \rightarrow c_2$, and 2) $x_1 + y_1 \rightarrow c_1$ and $x_2 + y_1 \rightarrow c_3$.

In utilizing the MSCM-AKD framework, two critical steps are data partition and pattern merging. In this exercise, the partition of debt-related transactional data is driven by domain knowledge and top-m frequent demographic patterns recognized by domain experts. These top-m frequent demographic patterns actually represent m groups of customers. As a result, each sub-data-set in DB_2 is associated with a particular group of people in the whole population.

Patterns identified in respective data sets are finally merged in terms of combined associations and combined association clusters. A combined pattern integrates one demographic component, namely one of the m frequent demographic groups, into one to many items from the transactional data sets based on inputs of business analysts. Business inputs represent the expectations of domain experts, including whether a pattern makes sense in business or not, and whether it indicates significant business impacts (namely business interestingness).

Finally, a combined pattern is interesting only if it satisfies the interestingness of a corresponding combined pattern type, as defined in the following sections:

6.3 Interestingness for Combined Associations and Association Clusters

To support the discovery of combined associations and association clusters, we developed corresponding interestingness metrics.

For a combined association rule $x + y \rightarrow c$, the traditional interestingness metrics such as *support*, *confidence*, and *lift* contribute little to selecting actionable combined association rules. To measure the interestingness of a combined association, we define two new *lifts* based on traditional *support*, *confidence* and *lift*

$$Lift_x(x + y \rightarrow c) = \frac{Conf(x + y \rightarrow c)}{Conf(y \rightarrow c)}, \tag{16}$$

$$Lift_y(x + y \rightarrow c) = \frac{Conf(x + y \rightarrow c)}{Conf(x \rightarrow c)}. \tag{17}$$

$Lift_x(x + y \rightarrow c)$ is the lift of x with y as a precondition, which shows how much x contributes to the rule. Similarly, $Lift_y(x + y \rightarrow c)$ gives the contribution of y in the rule. The following can be derived from the above formulas:

$$Lift_x(x + y \rightarrow c) = \frac{Lift(x + y \rightarrow c)}{Lift(y \rightarrow c)}, \tag{18}$$

$$Lift_y(x + y \rightarrow c) = \frac{Lift(x + y \rightarrow c)}{Lift(x \rightarrow c)}. \tag{19}$$

Based on the above *lifts*, the interestingness (I_{rule}) of a single combined association is defined as follows:

$$\begin{aligned} I_{\text{rule}}(x + y \rightarrow c) &= \frac{Lift_x(x + y \rightarrow c)}{Lift(x \rightarrow c)} \\ &= \frac{Lift_y(x + y \rightarrow c)}{Lift(y \rightarrow c)}. \end{aligned} \tag{20}$$

I_{rule} indicates whether the contribution of x (or y) to the occurrence of c increases with y (or x) as a precondition. Therefore, "$I_{\text{rule}} < 1$" suggests that $x + y \rightarrow c$ is less interesting than $x \rightarrow c$ and $y \rightarrow c$. The value of I_{rule} falls in [0,+∞). When $I_{\text{rule}} > 1$, the higher I_{rule} is, the more interesting the rule is.

Furthermore, we define the interestingness of a combined association cluster. First, the interestingness of a pair of combined association rules is defined as follows: Suppose p_1 and p_2 are a pair of combined associations with different consequents within a single rule cluster, say, $p_1 = (x_1 + y_1 \rightarrow c_1)$ and $p_2 = (x_2 + y_2 \rightarrow c_2)$, where $x_1 = x_2$, $y_1 \neq y_2$, and $c_1 \neq c_2$ (or $x_1 \neq x_2$, $y_1 = y_2$, and $c_1 \neq c_2$), the interestingness (I_{pair}) of the rule pair $\{p_1, p_2\}$ is defined as:

$$I_{\text{pair}}(p_1, p_2) = Conf(p_1)\ Conf(p_2). \tag{21}$$

I_{pair} measures the contribution of the two different parts in antecedents to the occurrence of different classes in a group of customers with the same demographics or the

TABLE 2
Arrangement Rules "$y \rightarrow c$"

y		c	$Conf(\%)$	$Count$	$Lift$
Arrangements	Repayments	Class			
irregular	cash or post office	A	82.4	4088	1.8
withholding	cash or post office	A	87.6	13354	1.9
withholding & irregular	cash or post office	A	72.4	894	1.6
withholding & irregular	cash or post office & withholding	B	60.4	1422	1.7

TABLE 3
Sample Combined Associations

Rules	x	y		c	Cnt	$Conf$ (%)	I_{rule}	$Lift$	$Lift_x$	$Lift_y$	$Lift$ of $x \rightarrow c$	$Lift$ of $y \rightarrow c$
	Demographics	Arrangements	Repayments	Class								
p_1	age:65+	withholding & irregular	withholding	C	50	63.3	2.91	3.40	2.47	4.01	0.85	1.38
p_2	income:0 & remote:Y & marrital:sep & gender:F	withholding	cash or post & withholding	B	20	69.0	1.47	1.95	1.34	2.15	0.91	1.46
p_3	income:0 & age:65+	withholding	cash or post & withholding	A	1123	62.3	1.38	1.35	1.72	1.09	1.24	0.79
p_4	income:0 & gender:F & benefit:P	withholding	cash or post	A	469	93.8	1.36	2.04	1.07	2.59	0.79	1.90

TABLE 4
Sample Combined Association Clusters

Clusters	Rules	x	y		c	Cnt	$Conf$ (%)	I_{rule}	I_{cluster}	$Lift$	$Lift_x$	$Lift_y$	$Lift$ of $x \rightarrow c$	$Lift$ of $y \rightarrow c$
		demographics	arrangements	repayments										
R_1	p_5	marital:sin	irregular	cash or post	A	400	83.0	1.12	0.67	1.80	1.01	2.00	0.90	1.79
	p_6	&gender:F	withhold	cash or post	A	520	78.4	1.00		1.70	0.89	1.89	0.90	1.90
	p_7	&benefit:N	withhold & irregular	cash or post & withhold	B	119	80.4	1.21		2.28	1.33	2.06	1.10	1.71
	p_8		withhold	cash or post & withhold	B	643	61.2	1.07		1.73	1.19	1.57	1.10	1.46
	p_9		withhold & vol. deduct	withhold & direct debit	B	237	60.6	0.97		1.72	1.07	1.55	1.10	1.60
	p_{10}		cash	agent	C	33	60.0	1.12		3.23	1.18	3.07	1.05	2.74
R_2	p_{11}	age:65+	withhold	cash or post	A	1980	93.3	0.86	0.59	2.02	1.06	1.63	1.24	1.90
	p_{12}		irregular	cash or post	A	462	88.7	0.87		1.92	1.08	1.55	1.24	1.79
	p_{13}		withhold & irregular	cash or post	A	132	85.7	0.96		1.86	1.18	1.50	1.24	1.57
	p_{14}		withhold & irregular	withhold	C	50	63.3	2.91		3.40	2.47	4.01	0.85	1.38
R_3	p_{15}	benefit:Y	irregular	cash or post	A	218	79.6	1.15	0.52	1.73	0.97	2.06	0.84	1.79
	p_{16}	&age:22-25	cash	cash or post	C	483	65.6	0.78		3.53	1.38	1.99	1.78	2.56
R_4	p_{17}	income:0	irregular	cash or post	A	191	76.7	1.03	0.48	1.66	0.93	1.85	0.90	1.79
	p_{18}	&age:22-25	cash	cash or post	C	440	62.1	1.08		3.34	1.31	2.76	1.21	2.56

same transactions. Such knowledge can help design business campaigns and improve business process. The value of I_{pair} falls in [0, 1]. The larger I_{pair} is, the more interesting and actionable a pair of rules are.

For an association cluster P with J combined associations $p_1, p_2, \ldots, p_J$, its interestingness (I_{cluster}) is:

$$I_{\text{cluster}}(P) = \max_{p_i, p_j \in R, i \neq j, c_i \neq c_j} \{I_{\text{pair}}(p_i, p_j)\}. \quad (22)$$

The above definition of I_{cluster} indicates that interesting clusters are the rule clusters with interesting rule pairs, and the other rules in the cluster provide additional information. Similar to I_{pair}, the value of I_{cluster} also falls in [0, 1].

With the above interestingness and traditional metrics: *support*, *confidence*, *lift*, $Lift_x$, $Lift_y$, and I_{rule}, interesting combined associations are filtered from the learned rules. Learned rules with high *support* and *confidence* are further merged into association clusters ranked by I_{cluster}.

6.4 Experiments

We test the above methods in government social security data with debts raised in the calendar year 2006 and the corresponding customers and arrangement/repayment activities. The cleaned sample data contains 355,800 customers with their demographic attributes, arrangements, and repayments. There are 7,711 traditional associations mined. The association rules are illustrated in Table 2. Tables 3 and 4 show samples of combined associations and association clusters, respectively. From the tables, it is clear that the combined associations cannot be discovered by traditional association rule techniques.

Compared with the single associations from respective data sets, the combined associations and combined association clusters are much more workable than single rules presented in the traditional way. They contain much richer information from multiple aspects rather than from a single one, or a collection of separated single rules. For instance, the following combined association shows that customers aged 65 or more, whose arrangement method is of "withholding" plus "irregular," and who actually repay in the approach of "withholding," can be classified into class "C" (high risk of payback). Obviously, this pattern combines heterogeneous information regarding the specific group of the debtor's demographic, repayment, and arrangement method

$$\{x = age : 65+, y = withholding \,\&\, irregular + withholding \rightarrow c = C\}. \quad (23)$$

Finally, combined patterns can be transformed into operable business rules that may indicate direct actions for business decision making. For instance, for the above combined association, it actually connects key business elements with segmented customer characteristics, and we can generate the following business rule by extending the BusinessRule specification:

DELIVERING BUSINESS RULES: Customer Demographic-Arrangement-Repayment combination business rules
For All *customer i ($i \in I$ is the number of valid customers)*
Condition:
satisfies *S/he is a debtor aged 65 or plus*;
relates
S/he is under arrangement of "withholding" and "irregularly",
and
His/her favorite Repayment method is "withholding";
Operation:
Alert = "*S/he has 'High' risk of paying off debt in a very long timeframe.*"
Action = "*Try other arrangements and repayments in R_2, such as trying to persuade her/him to repay under 'irregular' arrangement with 'cash or post.'*"
End-All

The converted business rules are deliverables presented to business people. They are convenient and it is easy for clients to embed them into their routine business processes and operational systems for filtering debtors and monitoring the debt recovery process. Our clients feel more comfortable in understanding, interpreting, and actioning these business rules than those patterns directly mined in the data. Therefore, combined patterns are more business-friendly and indicate much more straightforward decision-making actions to be taken by business analysts in the business world, while this cannot be achieved by patterns identified by traditional methods.

In addition, the use of combined mining leads to combined patterns consisting of attributes from different business units or by partitioning into organized segments. Through attribute segmentation or merger, it is manageable to differentiate attribute impact on business objectives, and thus, extract more and more informative patterns and more operable decision-making actions.

7 Conclusion

A common problem in mainstream KDD research is its dominating focus on algorithm innovation and neglect of real-life decision-making capability. Consequently, data mining applications face the significant problem of workability of deployed algorithms, tools, and resulting deliverables. To fundamentally change such situations, and empower the workable capability and performance of advanced data mining in real-world production and economy, there is an urgent need to develop next-generation data mining methodologies and techniques that target the paradigm shift from data-centered hidden pattern mining to domain-driven actionable knowledge delivery.

This paper has formally defined the AKD concepts, processes, actionability of patterns, and operable deliverables. With such components, we have proposed four types of AKD frameworks capable of handling various business problems and applications. These frameworks support closed-optimization-based problem solving from a business problem/environment definition, to actionable pattern discovery, and to operable business rule conversion. Deliverables extracted in this way are not only of technical significance but also are capable of smoothly integrating into business processes.

Substantial experiments in significant data mining applications such as financial data mining and mining social security data have shown that the proposed frameworks have the potential to handle the limitations in existing methodologies and approaches. They are sufficiently general, flexible, and workable to be instantiated into various approaches for tackling complex data and business applications.

Following the D^3M theory, there are many issues to be studied, for instance, defining operable business rules by involving ontological techniques for representing both syntactic and semantic components.

Acknowledgments

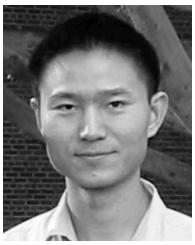
This work is sponsored in part by the Australian Research Council Discovery Grants (DP0988016, DP0773412, and DP0667060) and ARC Linkage Grant (LP0989721 and LP0775041).

References

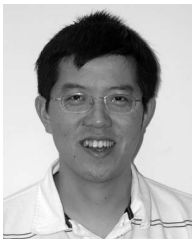
[1] G. Adomavicius and A. Tuzhilin, "Discovery of Actionable Patterns in Databases: The Action Hierarchy Approach," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD '97),* pp. 111-114, 1997.
[2] C. Aggarwal, "Towards Effective and Interpretable Data Mining by Visual Interaction," *ACM SIGKDD Explorations Newsletter,* vol. 3, no. 2, pp. 11-22, 2002.
[3] M. Ankerst, "Report on the SIGKDD-2002 Panel the Perfect Data Mining Tool: Interactive or Automated?" *ACM SIGKDD Explorations Newsletter,* vol. 4, no. 2, pp. 110-111, 2002.
[4] J.F. Boulicaut and B. Jeudy, "Constraint-Based Data Mining," *The Data Mining and Knowledge Discovery Handbook,* pp. 399-416, Springer, 2005.
[5] K. Breitman, M. Casanova, and W. Truszkowski, *Semantic Web.* Springer, 2007.
[6] L. Cao, "Domain-Driven Actionable Knowledge Discovery," *IEEE Intelligent Systems,* vol. 22, no. 4, pp. 78-89, July/Aug. 2007.
[7] L. Cao, "Domain-Driven Data Mining: Empowering Actionable Knowledge Delivery," *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '09) Tutorial,* 2009.

[8] L. Cao and Y. Ou, "Market Microstructure Pattern Analysis for Powering Trading and Surveillance Agents," *J. Universal Computer Science,* vol. 14, no. 14, pp. 2288-2308, 2008.
[9] L. Cao, "Developing Actionable Trading Strategies," *Knowledge Processing and Decision Making in Agent-Based Systems,* N. Nguyen and L. Jain, eds., Springer, 2008.
[10] L. Cao, P. Yu, C. Zhang, and H. Zhang, *Data Mining for Business Applications.* Springer, 2008.
[11] L. Cao, Y. Zhao, C. Zhang, and H. Zhang, "Activity Mining: From Activities to Actions," *Int'l J. Information Technology and Decision Making,* vol. 7, no. 2, pp. 259-273, 2008.
[12] L. Cao, P. Yu, C. Zhang, and Y. Zhao, *Domain Driven Data Mining.* Springer, 2009.
[13] L. Cao, Y. Zhao, and C. Zhang, "Mining Impact-Targeted Activity Patterns in Imbalanced Data," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 8, pp. 1053-1066, Aug. 2008.
[14] L. Cao and C. Zhang, "Knowledge Actionability: Satisfying Technical and Business Interestingness," *Int'l J. Business Intelligence and Data Mining,* vol. 2, no. 4, pp. 496-514, 2007.
[15] L. Cao and C. Zhang, "The Evolution of KDD: Towards Domain-Driven Data Mining," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 21, no. 4, pp. 677-692, 2007.
[16] L. Cao and C. Zhang, "Fuzzy Genetic Algorithms for Pairs Mining," *Proc. Pacific Rim Int'l Conf. Artificial Intelligence (PRICAI '06),* pp. 711-720, 2006.
[17] L. Cao, R. Dai, and M. Zhou, "Metasynthesis: M-Space, M Interaction and M-Computing for Open Complex Giant Systems," *IEEE Trans. Systems, Man, and Cybernetics-Part A,* vol. 39, no. 5, pp. 1007-1021, Sept. 2009.
[18] L. Cao, H. Zhang, Y. Zhao, and C. Zhang, "Combined Mining: Discovering More Informative Knowledge in e-Government Services," technical report, Univ. of Technology Sydney, 2008.
[19] U. Fayyad, G. Shapiro, and R. Uthurusamy, "Summary from the KDD-03 Panel—Data mining: The Next 10 Years," *ACM SIGKDD Explorations Newsletter,* vol. 5, no. 2, pp. 191-196, 2003.
[20] U. Fayyad and P. Smyth, "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining,* U. Fayyad and P. Smyth, eds., pp. 1-34, AAAI Press/MIT Press, 1996.
[21] A. Freitas, "On Objective Measures of Rule Surprisingness," *Proc. Second European Symp. Principles of Data Mining and Knowledge Discovery (PKDD '98),* pp. 1-9, 1998.
[22] O.G. Ali and W. Wallace, "Bridging the Gap between Business Objectives and Parameters of Data Mining Algorithms," *Decision Support Systems,* vol. 21, pp. 3-15, 1997.
[23] H. Kargupta, B. Park, D. Hershbereger, and E. Johnson, "Collective Data Mining: A New Perspective toward Distributed Data Mining," *Advances in Distributed Data Mining,* H. Kargupta and P. Chan, eds., AAAI/MIT Press, 1999.
[24] J. Kleinberg, C. Papadimitriou, and P. Raghavan, "A Microeconomic View of Data Mining," *Data Mining and Knowledge Discovery,* vol. 2, no. 4, pp. 311-324, 1998.
[25] R. Hilderman and H. Hamilton, "Applying Objective Interestingness Measures in Data Mining Systems," *Proc. Symp. Principles of Data Mining and Knowledge Discovery (PKDD),* pp. 432-439, 2000.
[26] B. Lent, A.N. Swami, and J. Widom, "Clustering Association Rules," *Proc. 13th Int'l Conf. Data Eng.,* pp. 220-231, 1997.
[27] B. Liu, W. Hsu, and Y. Ma, "Pruning and Summarizing the Discovered Associations," *Proc. ACM SIGKDD,* 1999.
[28] B. Liu and W. Hsu, "Post-Analysis of Learned Rules," *Proc. Nat'l Conf. Artificial Intelligence/Innovative Applications of Artificial Intelligence Conf. (AAAI/IAAI),* 1996.
[29] B. Liu, W. Hsu, S. Chen, and Y. Ma, "Analyzing Subjective Interestingness of Association Rules," *IEEE Intelligent Systems,* vol. 15, no. 5, pp. 47-55, Sept./Oct. 2000.
[30] E. Omiecinski, "Alternative Interest Measures for Mining Associations," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 1, pp. 57-69, Jan./Feb. 2003.
[31] B. Padmanabhan and A. Tuzhilin, "A Belief-Driven Method for Discovering Unexpected Patterns," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 94-100, 1998.
[32] B. Park and H. Kargupta, "Distributed Data Mining: Algorithms and Systems, Applications," *Data Mining Handbook,* pp. 341-358, 2002.
[33] A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems," *IEEE Trans. Knowledge and Data Eng.,* vol. 8, no. 6, pp. 970-974, Dec. 1996.
[34] A. Silberschatz and A. Tuzhilin, "On Subjective Measures of Interestingness in Knowledge Discovery," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 275-281, 1995.
[35] P. Tan, V. Kumar, and J. Srivastava, "Selecting the Right Interestingness Measure for Association Patterns," *Proc. ACM SIGKDD,* pp. 32-41, 2002.
[36] A. Tzacheva and Z. Ras, "Action Rules Mining," *Int'l J. Intelligent Systems,* vol. 20, no. 7, pp. 719-736, 2005.
[37] K. Wang, S. Zhou, and J. Han, "Profit Mining: From Patterns to Actions," *Proc. Int'l Conf. Extending Database Technology (EBDT),* 2002.
[38] Q. Yang, J. Yin, C. Ling, and R. Pan, "Extracting Actionable Knowledge from Decision Trees," *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 1, pp. 43-56, Jan. 2007.
[39] Y. Yao and Y. Zhao, "Explanation-Oriented Data Mining," *Encyclopedia of Data Warehousing and Mining,* J. Wang, ed., pp. 492-497, 2005.
[40] S. Yoon, L. Henschen, E. Park, and S. Makki, "Using Domain Knowledge in Knowledge Discovery," *Proc. Eighth Int'l Conf. Information and Knowledge Management,* pp. 243-250, 1999.
[41] H. Zhang, Y. Zhao, L. Cao, C. Zhang, and H. Bohlscheid, "Customer Activity Sequence Classification for Debt Prevention in Social Security," *J. Computer Science and Technology,* vol. 24, no. 6, pp. 1000-1009, 2009.
[42] *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction,* Y. Zhao, C. Zhang, and L. Cao, eds. IGI Press, 2008.
[43] Y. Zhao, H. Zhang, L. Cao, C. Zhang, and Y. Ou, "Data Mining Application in Social Security Data," *Data Mining for Business Applications,* L. Cao, P. Yu, C. Zhang, and H. Zhang, eds., Springer, 2008.

Longbing Cao received the PhD degrees in both intelligence sciences and computing sciences. He is a professor at the University of Technology, Sydney, and the data mining research leader of the Australian Capital Markets Cooperative Research Centre. His research interests include data mining, multiagent technology, agent and data mining integration, and behavior informatics. He is a senior member of the IEEE.



Yanchang Zhao is a research fellow at the Data Sciences and Knowledge Discovery Research Lab, Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia. His research interests include association rules, time series, sequential patterns, clustering, and postmining. He is a member of the IEEE.



Huaifeng Zhang is a research fellow at the Data Sciences and Knowledge Discovery Research Lab, Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia. His research interests include combined pattern mining, sequence classification, pattern recognition, computer vision, behavior analysis and modeling, etc. He is a member of the IEEE.

Dan Luo is a research fellow at the Data Sciences and Knowledge Discovery Research Lab, Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia. Her research focuses on domain-driven data mining, actionable knowledge discovery, and business intelligence.

Chengqi Zhang is a research professor of information technology and the director of the Centre for Quantum Computation and Intelligent Systems at the University of Technology, Sydney, Australia. He is the chair of the Australian Computer Society National AI Committee and a data mining research leader of the Australian Capital Markets Cooperative Research Centre. His research interests include data mining, multiagent technology, and their integration. He is a senior member of the IEEE.

E.K. Park received the PhD degree in computer science from Northwestern University. He is a professor of computer science at the University of Missouri at Kansas City. His research interests include data mining, bioinformatics, information and knowledge management, computer communications and networks, optical networks, distributed systems, and object-oriented methodology. He is a program director, Division of Computing and Communications Foundations at the US National Science Foundation.