Domain-Driven Data Mining: Challenges and Prospects

Longbing Cao, *Senior Member*, *IEEE*

Abstract—Traditional data mining research mainly focus]es on developing, demonstrating, and pushing the use of specific algorithms and models. The process of data mining stops at pattern identification. Consequently, a widely seen fact is that 1) many algorithms have been designed of which very few are repeatable and executable in the real world, 2) often many patterns are mined but a major proportion of them are either commonsense or of no particular interest to business, and 3) end users generally cannot easily understand and take them over for business use. In summary, we see that the findings are not actionable, and lack soft power in solving real-world complex problems. Thorough efforts are essential for promoting the actionability of knowledge discovery in real-world smart decision making. To this end, *domain-driven data mining* (D^3M) has been proposed to tackle the above issues, and promote the paradigm shift from "data-centered knowledge discovery" to "domain-driven, actionable knowledge delivery." In D^3M, ubiquitous intelligence is incorporated into the mining process and models, and a corresponding problem-solving system is formed as the space for knowledge discovery and delivery. Based on our related work, this paper presents an overview of driving forces, theoretical frameworks, architectures, techniques, case studies, and open issues of D^3M. We understand D^3M discloses many critical issues with no thorough and mature solutions available for now, which indicates the challenges and prospects for this new topic.

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

1 Introduction

As one of the most active areas in information technology, data mining and knowledge discovery (data mining or KDD for short) has resulted in probably thousands of algorithms and models. At the mean time, we have seen an extreme imbalance between the number of published algorithms versus those really workable in the business environment. Surveys of data mining for business applications following the above paradigm in various domains [15] have shown that findings cannot make themselves executable in the real world. That is to say, there is a big gap between academic objectives and business goals, and between academic outputs and business expectations. This runs counter to the experiment-based nature and value of KDD as a discipline, which is supposed to enable smart business intelligence for smart decisions in production.

If we scrutinize the reasons for the existing gaps, we can probably point out many things. For instance, academic researchers do not really understand the needs of business people, and do not take the business environment into account. Data mining algorithms and tools generally only focus on the discovery of patterns satisfying expected technical significance.

Compared to the relatively mature situation of algorithm innovation, it is timely and worthwhile to review the major issues surrounding the "gap" that blocks the step of KDD into wider business use. Furthermore, serious efforts should be made to develop workable methodologies, techniques, and case studies to promote another round of booming research and development of data mining in real-world problem solving. In recent years, researchers with strong industrial engagement have realized the need to shift from "data mining" to "knowledge discovery" [20], [1], [19]. Targeting real-world problem solving, knowledge discovery is further expected to migrate into *actionable knowledge discovery and delivery* (AKD). AKD aims to deliver knowledge that is business friendly, and which can be taken over by business people for seamless decision making.

To bridge the gap and enhance real-world problem-solving capabilities, it is necessary to take a critical view of KDD, such as from microeconomic [23] and system perspectives, and to develop fundamental and workable methodologies and frameworks [17] to support AKD. Motivated by lessons learned in tackling complex enterprise data mining applications, and through scrutinizing both macrolevel and microlevel issues and requirements in AKD, we have proposed the methodology of *Domain-Driven Data Mining* (D^3M) [10], [11], [3], [16], which surmounts the traditional data-centered pattern mining framework, for guiding AKD in a complex environment. Since then, intensive studies have been conducted on theoretical development such as general architectures supporting actionable knowledge discovery [17], and techniques such as combined mining [41] and postmining [40]. We have also applied D^3M in tackling real-world problems in government debt prevention in social security [13] and developing actionable trading strategies and trading agents [14], [4], [9].

AKD is critical in promoting and releasing the productivity of KDD for smart information extraction, business operations, and decision making. Both SIGKDD and ICDM panelists have identified it as one of the great challenges in developing the next-generation KDD methodologies and

• *The author is with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia. E-mail: lbcao@it.uts.edu.au.*

Manuscript received 1 Apr. 2009; revised 6 Oct. 2009; accepted 26 Oct. 2009; published online 4 Feb. 2010.
Recommended for acceptance by C. Zhang, P.S. Yu, and D. Bell.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-2009-04-0298.
Digital Object Identifier no. 10.1109/TKDE.2010.32.

systems [1], [19]. The research on D^3M, targeting AKD, discloses unprecedented opportunities for developing next-generation data mining methodology and infrastructure, which foster the potential of paradigm shift from "*data-driven hidden pattern mining*" to "*domain-driven actionable knowledge delivery*," and promote the widespread acceptance of KDD in real business use as extensively as possible. However, we clearly understand that it is not yet at the stage of delivering complete and mature solutions for AKD. The studies on D^3M actually open many new (and existing) research issues, which disclose the need of significant commitment from not only the KDD community but also many related disciplines.

In this paper, rather than introducing specific accomplishments made on D^3M, we focus on presenting a systematic overview of concepts, challenges, techniques, and prospects of D^3M. In Section 2, we discuss issues in traditional data mining and present a multidimensional requirement on domain-driven actionable knowledge discovery. Section 3 discusses the concept and fundamental framework of D^3M involving ubiquitous intelligence, evaluation systems, and delivery for D^3M. In Section 4, we summarize several system architectures and techniques supporting D^3M. Four case studies are briefed in Section 5 to illustrate the use of D^3M in handling real-world problems. Section 6 briefly discusses open issues and trends of D^3M research and development. We conclude this paper in Section 7.

2 D^3M Driving Forces

2.1 Issues of Traditional Data Mining Studies

Existing efforts related to AKD mainly focus on developing more effective interestingness metrics [21], [24], converting and summarizing learned rules through postanalysis and postmining [40], and the combination of multiple relevant techniques [25].

The main efforts of developing effective interestingness metrics focus on *objective technical* interestingness metrics ($t_o()$) [21], [24]. They aim to capture the *complexities of pattern structure and statistical significance*. Other work appreciating *subjective technical* measures ($t_s()$) [28], [30], [34] also recognizes *to what extent a pattern is of interest to particular user preferences*. For example, *probability-based belief* is used to describe user confidence in unexpected rules [30]. There is very limited research on developing business-oriented interestingness, for instance, profit mining [36].

The related work on developing alternative interestingness measures focuses on the technical interestingness only [29]. Emerging research on general business-oriented interestingness is isolated from the technical significance. A question to be asked is "what makes interesting patterns actionable in the real world?" For that, knowledge actionability can be marked as the general interestingness measurement of both *technical* and *business-oriented* interestingness from both *objective* and *subjective* perspectives [12].

To promote the transformation from data mining to knowledge discovery [20], postanalysis, and postmining have been the main approach to filter/prune rules and summarize learned rules [27], reduce redundancy [25], or match against expected patterns by similarity/difference. A recent highlight is to *extract actions from learned rules* by splitting attributes into "hard/soft" [38] or "stable/flexible" [35] to extract actions that may improve the loyalty or profitability of customers. However, most existing post-analysis and postmining focuses on association rules or the combination with specific methods such as classification. This limits the actionability of learned actions and the generalization of proposed approaches.

In recent years, the combination of relevant algorithms has emerged as a powerful tool for identifying more effective patterns. Typical work consists of a combination of two or more methods. For instance, *class association rules* (or *associative classifiers*) build classifiers on association rules [22]. In [31], clustering is used to reduce the number of learned association rules. Williams and Huang [37] propose the hot spot method to cluster very large data sets; decision trees are then used for rule induction. In [16], a comprehensive overview is drawn on combined mining, including approaches of combining multiple data sources, multiple features, and multiple methods for more informative combined patterns.

In summary, the issues surrounding traditional data mining studies can be categorized as follows:

- Real-world business problems are often buried in *complicated environments and factors*. The environmental elements are often filtered or largely simplified in traditional data mining research. As a result, there is a big gap between a syntactic system and its actual target problem. The identified patterns cannot be used for problem solving.
- Even though good data mining algorithms are important, any real-world data mining is a *problem-solving process and system*. It involves many other businesses such as catering for user interactions, environmental factors, connected systems, and deliverables to business decision makers.
- Existing work often stops at pattern discovery, which is mainly based on technical significance and interestingness. *Business concerns* are not considered in assessing patterns. Consequently, the identified patterns are predominantly of technical interest.
- There are *often many* patterns mined but they are not informative and transparent to business people, who cannot easily obtain the *truly interesting* patterns for their businesses.
- A large proportion of the identified patterns may be either *commonsense or of no particular interest* to business needs. Business people feel confused by *why* and *how* they should care about those findings.
- Actions extracted or summarized through *postanalysis and postprocessing* without considering business concerns do not reflect the genuine expectations of business needs, and therefore cannot support smart decision making.
- Business people often do not know, and are also not informed *how to interpret and use/execute* them and *what straightforward actions can be taken* to engage them in business operational systems and decision making.
- Often algorithms are delivered, but they are not executable and operable in the business system. No effective tools are provided to *convert models to executables* that can be integrated into production systems.

They greatly contribute to the significant gap between data mining research and applications, the weak AKD capability, and the bottlenecks of widespread deployment of data mining.

2.2 Multidimensional Requirements on AKD

AKD is important because of multiple dimensions of requirements on both macrolevel and microlevel from real-world applications.

On the macrolevel, issues are related to methodological and fundamental aspects. For instance, an intrinsic difference exists in academic thinking and business expectation. An example is that researchers usually are interested in innovative pattern types, while practitioners care about getting a problem solved. A strategic position needs to be taken as to whether to focus on a hidden pattern mining process centered by data, or an AKD-based problem-solving system as the deliverable. The following typical macrolevel issues need to be addressed: environment, human, process, infrastructure, dynamics, evaluation, risk policy, and deliverable.

- Environment: Refer to any factors surrounding data mining models and systems, for instance, domain factors, constraints, expert groups, organizational factors, social factors, business processes, and workflows. They are inevitable and important for AKD. Some factors such as constraints have been considered in current data mining research, but many others have not. It is essential to represent, model, and involve them in AKD systems and processes.
- Human role: To handle many complex problems, human-centered and human-mining-cooperated AKD is crucial. Critical problems related to this include how to involve domain experts and expert groups into the mining process, and how to allocate the roles between human and mining systems.
- Process: Real-world problem solving has to cater for dynamic and iterative involvement of environmental elements and domain experts along the way.
- Infrastructure: The engagement of environmental elements and humans at runtime in a dynamic and interactive way requires an open system with closed-loop interaction and feedback. AKD infrastructure should provide facilities to support such scenarios.
- Dynamics: To deal with the dynamics in data distribution from training to testing and from one domain to another, in domain and organizational factors, in human cognition and knowledge, in the expectation of deliverables, and in business processes and systems.
- Evaluation: Interestingness needs to be balanced between technical and business perspectives from both subjective and objective aspects; special attention needs to be paid to deliverable formats, and its actionability and generalizable capability, as well as the support from domain experts.
- Risk: Risk needs to be measured in terms of its presence and then magnitude, if any, in conducting an AKD project and system.
- Policy: Data mining tasks often involve policy issues such as security, privacy, and trust existing not only in the data and environment, but also in the use and management of data mining findings in an organization's environment.
- Delivery: Determining the right form of delivery and presentation of AKD models and findings so that end users can easily interpret, execute, utilize, and manage the resulting models and findings, and integrate them into business processes and production systems.

On the microlevel, issues related to technical and engineering aspects supporting AKD need to be addressed. Aiming at an AKD-based problem-solving system, we then need to develop facilities for involving system dynamics, the system environment, and interactions in data mining. For instance, to involve environmental elements, what tools are necessary for engaging business processes, organizational factors, and constraints in data mining? The following lists a few dimensions that address these concerns: architecture, process, interaction, adaptation, actionability, and deliverable.

- Architecture: AKD system architectures need to be effective and flexible for incorporating and consolidating specific environmental elements, AKD processes, evaluation systems, and final deliverables.
- Process: Tools and facilities supporting the AKD process and workflow are necessary, from business understanding, data understanding, and human-system interaction to result assessment, delivery, and execution of the deliverables.
- Interaction: To cater for interaction with business people along the way of ADK process, appropriate user interfaces, user modeling, and servicing are required to support individuals and group interactions.
- Adaptation: Data, environmental elements, and business expectations change all the time. AKD systems, models, and evaluation metrics are required to be adaptive for handling differences and changes in dynamic data distributions, cross domains, changing business situations, and user needs and expectations.
- Actionability: What do we mean by "actionability?" How should we measure it? What is the trade-off between technical and business sides? Do subjective and objective perspectives matter? This requires essential metrics to be developed.
- Deliverable: End users certainly feel more comfortable if the models and patterns delivered can be presented in a business-friendly way and be compatible with business operational systems and rules. In this sense, AKD deliverables are required to be easily interpretable, convertible into or presented in a business-oriented way such as business rules, and be linked to decision-making systems.

3 D^3M Theoretical Framework

3.1 D^3M Basic Concepts

Real-world data mining is a complex problem-solving system. The main objective of D^3M is to enhance the

TABLE 1
Working Principle

Aspects	Data-Driven	Domain-Driven
Rationale	Data tells the story	Data and ubiquitous intelligence disclose problem-solving solutions
Objective	Innovative and effective algorithms	Effective problem-solving
Data	Abstract, synthetic and refined data	Real-life data and surrounding information
Process	One-off	Multiple-step, iterative and interactive on demand
Mechanism	Automated	Human-centered or human-mining-cooperated
Infrastructure	Closed pattern mining systems	Closed-loop problem-solving systems in open environment
Usability	Predefined models and processes	Ad-hoc, dynamic and customizable models and processes
Deliverable	Patterns	Business-friendly decision-support actions
Deployment	Solid validation	Well-founded artwork in problem-solving
Evaluation	Technical metrics	Tradeoff between technical significance and business expectation

actionability of identified patterns for problem solving. The term "actionability" measures *the ability of a pattern to prompt a user to take concrete actions to his/her advantage in the real world*. It mainly measures the ability to suggest business decision-making actions.

Let *DB* be a database collected from business problems (Ψ), $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ be the set of items in the *DB*, where $\mathbf{x}_l$ ($l = 1, \ldots, L$) be an item set, and the number of attributes (v) 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 the 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 $p_u^{m_n}$ ($u = 1, \ldots, U$) denotes a pattern discovered by the method m_n. From the viewpoint of systems and microeconomy, AKD is an optimization problem-solving process from business problems (Ψ, with problem status τ) to problem-solving solutions (Φ) with certain objectives in a particular environment. From the modeling perspective, such a problem-solving process is a state transformation from source data $DB(\Psi \rightarrow DB)$ to resulting pattern set $P(\Phi \rightarrow P)$.

$$\Psi \rightarrow \Phi :: DB(v_1, \ldots, v_S) \rightarrow P(f_1, \ldots, f_Q), \tag{1}$$

where v_s ($s = 1, \ldots, S$) are attributes in the source data DB, while f_q ($q = 1, \ldots, Q$) are features used for mining the pattern set P.

The goal of D^3M is to identify actionable patterns. Let $\widetilde{P} = \{\tilde{p_1}, \tilde{p_2}, \ldots, \tilde{p_Z}\}$ be an *Actionable Pattern Set* mined by the method m_n for a given problem Ψ (its data set is DB), in which each pattern $\tilde{p_z}$ is *actionable* for the problem solving if it satisfies the following conditions:

- $t_i(\tilde{p_z}) \geq t_{i,0}$; indicating the pattern $\tilde{p_z}$ satisfying technical interestingness t_i with threshold $t_{i,0}$;
- $b_i(\tilde{p_z}) \geq b_{i,0}$; indicating the pattern $\tilde{p_z}$ satisfying business interestingness b_i with threshold $b_{i,0}$;
- $$R : \tau_1 \overset{A, m_n(\tilde{p_z})}{\longrightarrow} \tau_2;$$

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

Therefore, the discovery of actionable knowledge on data set DB is an iterative optimization process toward the actionable pattern set $\widetilde{P}$.

$$AKD : DB \overset{e,\tau,m_1}{\longrightarrow} P_1 \overset{e,\tau,m_2}{\longrightarrow} P_2 \cdots \overset{e,\tau,m_n}{\longrightarrow} \widetilde{P}. \tag{2}$$

Correspondingly, the AKD is a procedure to find the *Actionable Pattern Set* $\widetilde{P}$ through employing all valid methods M. Its mathematical description is as follows:

$$AKD^{m_i \in M} \longrightarrow O_{p \in P} Int(p), \tag{3}$$

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

The main task of D^3M is to develop AKD-oriented problem-solving systems. AKD-oriented D^3M, on top of the data-driven framework, aims to complement the shortcomings of traditional data mining, through developing proper methodologies and techniques to incorporate domain knowledge, user needs, the human role and interaction, as well as actionability measures into KDD process and systems. It is data and domain intelligence working together to disclose a hidden story to business, and to satisfy real user needs. End users hold the final decision in evaluating the findings and business deliverables. Table 1 compares major aspects under the research of traditional data-driven and domain-driven data mining.

3.2 D^3M Ubiquitous Intelligence

In order to make D^3M systems deliver business-friendly and decision-making rules and actions that are of also solid technical significance, D^3M caters for the effective involvement of the following ubiquitous intelligence surrounding AKD-based problem solving.

3.2.1 In-Depth Data Intelligence

Data Intelligence tells interesting stories or uncovers indicators about a business problem hidden in the data. Even though mainstream data mining focuses on substantial investigation of various data for interesting hidden patterns or knowledge, the real-world data and surroundings are usually much more complicated.

- data type such as numeric, categorical, XML, multimedia, and composite data;
- data timing such as temporal and sequential;
- data spacing such as spatial and temporal-spatial;
- data speed and mobility such as high frequency, high density, dynamic data, and mobile data;
- data dimension such as multidimensional, high-dimensional data, and multiple sequences;
- data relation such as multirelational data and linkage record;
- data quality such as missing data, noise, uncertainty, and incompleteness; and
- sensitivity such as mixing with sensitive information.

Deeper and wider analysis in data and knowledge engineering is required to mine for in-depth data intelligence in complex data. Traditional data mining needs to be further developed for processing and mining real-world data complexities such as multidimensional data, high-dimensional data, mixed data, distributed data, and processing and mining unbalanced, noisy, uncertain, incomplete, dynamic, and stream data.

Fundamental underpinnings for dealing with real-life data complexities consist of *data quality enhancement* to enhance data quality and readiness for pattern mining, *data matching and integration* to match/integrate data from multiple heterogeneous data sources, *information coordination* to access multiple data sources through coordination techniques such as multiagent coordination, *feature extraction* such as extracting and representing features mixing semistructured or ill-structured data with structured data, *parallel computing* for processing multiple sources of high frequency data in parallel, *collective intelligence* of data such as through aggregating intelligence identified in individual data sources, *dimension reduction* to reduce the number of dimensions to a handleable level, *space mapping* such as mapping an input space to a feature hypersphere, *computational complexity* of data such as engaging quantum computing for more efficient computation, and *data policy* such as privacy and security processing during pattern mining while protecting sensitive information from disclosure, and so on.

3.2.2 Domain Intelligence

Domain Intelligence emerges from domain factors and resources that not only wrap a problem and its target data but also assist in problem understanding and problem solving. Domain intelligence involves qualitative and quantitative aspects. These are instantiated in terms of aspects such as domain knowledge, background information, prior knowledge, expert knowledge, constraints, organization factors, business process, and workflow, as well as environment intelligence, business expectation, and interestingness.

D^3M highlights the role of domain intelligence in actionable knowledge discovery and delivery. To incorporate domain intelligence into data mining, the following theoretical underpinnings are essential: *formal modeling* of domain factors and resources, *interaction design* to provide interfaces and channels for interactions between domain experts and a data mining system at the running time, *representation and involvement of domain knowledge* into the data mining system, *involving domain factors in the data mining model and process* to study techniques and tools for incorporating domain factors into data mining models and process, *catering for business process and workflow* in the mining process, *business interestingness* to measure the pattern importance of business interest, and the trade-off strategies for filtering patterns of both technical and business importance.

In particular, domain knowledge in business fields often takes forms of precise knowledge, concepts, beliefs, relations, or vague preference and bias. The integration of domain knowledge is subject to how it can be represented and filled into the knowledge discovery process. Ontological engineering and semantic web can be used for representing such forms from both syntactic and semantic perspectives. For example, ontology-based specifications can be developed to build a business ontological domain representing domain knowledge that can be mapped to a low-level domain for a mining system.

3.2.3 Network Intelligence

Network Intelligence emerges from both web intelligence [42] and broad-based network intelligence such as information and resources distribution, linkages among distributed objects, hidden communities and groups, information and resources from network and in particular the web, information retrieval, searching, and structuralization from distributed and textual data. The information and facilities from the networks surrounding the target business problem either consist of the problem constituents, or can contribute to useful information for actionable knowledge discovery. Therefore, they should be catered for in AKD.

In saying "network intelligence," we expect to fulfill the power of network information and facilities for data mining in terms of, but not limited to, the following aspects:

- discovering the business intelligence in networked data related to a business problem,
- discovering networks and communities existing in a business problem and its data,
- involving networked constituent information in pattern mining on target data, and
- utilizing networking facilities to pursue information and tools for AKD.

To incorporate the above networked information or facilities into data mining, fundamental underpinnings such as the following aspects are important: *application integration* to link applications for data collection and integration, *data gateway and management* for accessing/mining local data, *data and feature fusion* for fusing data and features, *distributed computing* for mining local data, *information retrieval and searching* for retrieving and searching data from the network, *distributed data mining* (DDM) for mining distributed data sets, *combined mining* to mine for patterns consisting of subpatterns from individual data sets, *linkage analysis* for mining links and networks in the networked data, *group formation* for identifying communities and groups in data, and *data mobility* to mine for patterns in mobile network data.

3.2.4 Human Intelligence

Human Intelligence refers to 1) explicit or direct involvement of human empirical knowledge, belief, intention,

expectation, runtime supervision, evaluation, and expert groups into AKD; 2) implicit or indirect involvement of human intelligence such as imaginary thinking, emotional intelligence, inspiration, brainstorm, reasoning inputs, and embodied cognition like convergent thinking through interaction with other members in dynamic data mining and assessing identified patterns.

In enterprise data mining, both individuals and groups may be involved in the data mining process, which involves both intrapersonal and interpersonal levels of human intelligence. The interpersonal level of human intelligence is also discussed in human social intelligence. An example is an interactive system for the mining and understanding of abnormal cross-market trading behavior within a large exchange. A group of domain analysts who are familiar with relevant market models and cases are involved in tuning the models and evaluating the mined patterns. These experts sometimes discuss with each other and come up with refined parameters and models.

To involve human intelligence in AKD, many issues need to be studied. *Interactive data mining* [2] and *human-centered interactive data mining* deal with interface design and major roles played by humans in pattern mining. For complex cases, *human-centered data mining* or *human-aided data mining* are essential for incorporating human intelligence. Fundamental studies are essential on representing, modeling, processing, analyzing, and engaging human intelligence into AKD process, models, and deliverables. Typical challenges such as dynamic involvement, cognitive emergence, group-based involvement and divergence of opinions, and consensus building are important for handling complex and unclear data mining applications.

To support the involvement of human intelligence for AKD, the following techniques are essential: *dynamic user modeling* to capture user characteristics and inputs into data mining systems, *online user interaction* supporting online users to interact with a data mining system remotely, *group decision making* in pattern discovery such as involving a group of domain experts to evaluate and filter the identified patterns, *adaptive interaction* for users to adapt to the pattern discovery process and for models to adapt to user thinking and decisions, *distributed interaction* catering for multiple users to interact with models and each other on pattern discovery and evaluation, and *consensus building* for a group of users to form optimal and mutually agreed findings by dealing with thinking convergence and divergence.

3.2.5 Social Intelligence

Social Intelligence refers to the intelligence that lies behind group interactions, behaviors, and corresponding regulation. Social intelligence covers both human social intelligence and animat/agent-based social intelligence. Human social intelligence is related to aspects such as social cognition, emotional intelligence, consensus construction, and group decision. Animat/agent-based social intelligence involves swarm intelligence, action selection, and the foraging procedure. Both sides also engage social network intelligence and collective interaction, as well as business rules, law, trust, and reputation for governing the emergence and use of social intelligence.

In mining patterns in complex data and social environments, both types of social intelligence are essential in many aspects, for instance,

- the use of human social intelligence for supervised data mining and evaluation;
- the establishment of social data mining software on the basis of software agents, for instance, multiagent data mining and warehousing, to facilitate human-model interaction, group decision making, self-organization and autonomous action selection by data mining agents;
- developing performance evaluation models including trust and reputation models to evaluate and maintain the quality of social data mining software; and
- project management, business process management, and finding delivery from a data analyst to an operational department.

The engagement of social intelligence in data mining relies on fundamental techniques such as: *social computing* studying social software supporting data mining, *software agent technology* developing agent-based data mining systems, *swarm and collective intelligence* looking after swarm-based optimization and self-organization that can be very important in autonomous distributed data mining, *divergence and convergence of thinking* to construct consensus in social network for mining and evaluating patterns, and *agent mining* [8] to utilize multiagents to enhance data mining.

3.2.6 Intelligence Metasynthesis

In the above, we state the needs and techniques for incorporating ubiquitous intelligence into the data mining process and systems to enhance knowledge discovery. It is often more difficult to have everything well integrated in a data mining system, in addition to the challenges of modeling and involving specific intelligence. New data mining methodologies and techniques need to be developed to cater for the ubiquitous intelligence.

There are some requirements for the methodologies supporting the integration of ubiquitous intelligence, for instance, facilitating the human-model interaction and interactive data mining in a mining process-oriented, distributed, online, and group-based manner; supporting human-centered data mining in the sense that a group of human experts form the constituent of a data mining system and infrastructure, for instance, experts call and adjust models to mine for initial patterns and further call for the next step of postmining for more actionable patterns by supervising the pattern extraction and business-friendly deliverables; suitability for developing social data mining software that caters for social interaction, group behavior, and collective intelligence in the system.

An optional methodology is *Intelligence Metasynthesis*, which has been proposed to study open complex giant systems [32], [33]. It has been further expanded to handle open complex intelligent systems [6]. The main idea of the methodology is to develop a metasynthesis space (m-space) supporting metasynthetic interaction (m-interaction) and computing (m-computing) [7]. The methodology actually fosters an m-space for metasynthetic interaction and integration of ubiquitous computing techniques, including data and knowledge engineering, human-centered computing, organizational computing, social computing, behavior computing, and network computing.

The principle of intelligence metasynthesis is helpful for involving, synthesizing, and using ubiquitous intelligence surrounding actionable knowledge discovery in complex data. The future work is to apply the theory into developing an m-space which engages and facilitates the above intelligence through supporting a human-centered and human-machine-cooperated problem-solving process and group-based interaction and decision. An m-space is able to support human-centered data mining, dynamic and distributed interaction, group-based problem solving, engaging human knowledge and role in modeling and evaluation, by satisfying domain knowledge, constraints, and organizational factors.

To support AKD, an m-space needs many m-computing facilities, for instance,

- mechanisms for acquiring and representing unstructured, ill-structured, and uncertain knowledge such as empirical knowledge stored in domain experts' brains;
- mechanisms for acquiring and representing expert thinking such as imaginary thinking and creative thinking in group heuristic discussions;
- mechanisms for acquiring and representing group/collective interaction behavior and impact emergence, such as behavior informatics; and
- mechanisms for modeling learning-of-learning, i.e., learning other participants' behavior which is the result of self-learning or ex-learning, such as learning evolution and intelligence emergence.

3.3 D^3M Evaluation System

The D^3M evaluation system caters for significance and interestingness ($Int(p)$) of a pattern (p) from both technical and business perspectives. $Int(p)$ is measured in terms of *technical interestingness* ($t_i(p)$) and *business interestingness* ($b_i(p)$) [12].

$$Int(p) = I(t_i(p), b_i(p)), \tag{4}$$

where $I(.)$ is the function for aggregating the contributions of all particular aspects of interestingness.

Further, $Int(p)$ is described in terms of *objective*(o) and *subjective*(s) factors from both *technical*(t) and *business*(b) perspectives.

$$Int(p) = I(t_o(), t_s(), b_o(), b_s()), \tag{5}$$

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.

We say p is truly *actionable* (i.e., $\widetilde{p}$) to both academia and business if it satisfies the following condition:

$$Int(p) = t_o(\mathbf{x}, \widetilde{p}) \wedge t_s(\mathbf{x}, \widetilde{p}) \wedge b_o(\mathbf{x}, \widetilde{p}) \wedge b_s(\mathbf{x}, \widetilde{p}), \tag{6}$$

where "$\wedge$" indicates the interestingness "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:

$$\begin{aligned} Int(p) &\rightarrow \hat{I}(\hat{t_o}(), \hat{t_s}(), \hat{b_o}(), \hat{b_s}()) \\ &= \alpha\hat{t_o}() + \beta\hat{t_s}() + \gamma\hat{b_o}() + \delta\hat{b_s}(). \end{aligned} \tag{7}$$

The AKD optimization problem in D^3M can be expressed as follows:

$$\begin{aligned} &AKD^{e,\tau,m\in M} \longrightarrow O_{p\in P}(Int(p)) \\ &\quad \rightarrow O(\alpha\hat{t_o}()) + O(\beta\hat{t_s}()) + O(\gamma\hat{b_o}()) + O(\delta\hat{b_s}()). \end{aligned} \tag{8}$$

The *actionability* of a pattern p is measured by $act(p)$:

$$\begin{aligned} &act(p) = O_{p\in P}(Int(p)) \\ &\quad \rightarrow O(\alpha\hat{t_o}(p)) + O(\beta\hat{t_s}(p)) + O(\gamma\hat{b_o}(p)) + O(\delta\hat{b_s}(p)) \\ &\quad \rightarrow t_o^{act} + t_s^{act} + b_o^{act} + b_s^{act} \\ &\quad \rightarrow t_i^{act} + b_i^{act}, \end{aligned} \tag{9}$$

where t_o^{act}, t_s^{act}, b_o^{act}, and b_s^{act} measure the respective actionable performance in terms of each aspect.

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

$$\begin{aligned} Int(p) \rightarrow \{&\{t_i^{act}, b_i^{act}\}, \{\neg t_i^{act}, b_i^{act}\}, \\ &\{t_i^{act}, \neg b_i^{act}\}, \{\neg t_i^{act}, \neg b_i^{act}\}\}, \end{aligned} \tag{10}$$

where "$\neg$" indicates the corresponding element is not satisfactory. Ideally, we look for actionable patterns p that can satisfy the following condition:

IF

$$\begin{aligned} \forall p \in \widetilde{P}, \exists \mathbf{x} : t_o(\mathbf{x}, p) \wedge t_s(\mathbf{x}, p) \wedge b_o(\mathbf{x}, p) \\ \wedge\, b_s(\mathbf{x}, p) \rightarrow act(p), \end{aligned} \tag{11}$$

THEN:

$$p \rightarrow \widetilde{p}. \tag{12}$$

In real-world data mining, it is often very challenging to find the most actionable patterns that are associated with both "optimal" t_i^{act} and "optimal" b_i^{act}. Clearly, D^3M favors patterns confirming the relationship $\{t_i^{act}, b_i^{act}\}$. There is a need to deal with possible conflict and uncertainty among respective interestingness elements. Technically, there is an opportunity to develop techniques to balance and combine all types of interestingness metrics to generate uniform, balanced, and interpretable mechanisms for measuring knowledge deliverability. Under sophisticated situations, domain experts from both computation and business areas need to interact with each other, ideally through an m-space with intelligence metasynthesis facilities such as letting one run models with quantitative outcomes to support discussions with other experts. If $t_i()$ and $b_i()$ are inconsistent, experts argue and compromise with each other through m-interactions in the m-space, such as happens in a board meeting, but with substantial online resources, models, and services.

3.4 D^3M Delivery System

Well-experienced data mining professionals attribute the weak executable capability of existing data mining findings to the lack of proper tools and mechanisms for implementing the ideal deployment of the resulting models and algorithms by business users rather than analysts. In fact, the barrier and gap comes from the weak, if not nonexistent, capability of existing data mining deployment systems, found in presentation, deliverable, and execution aspects.

They form the D^3M delivery system, which is much beyond the identified patterns and models themselves.

- Presentation: studies how to present data mining findings that can be easily recognized, interpreted, and taken over as needed.
- Deliverable: studies how to deliver data mining findings and systems to business users so that the findings can be readily reformated, transformed, or cut and pasted into their own business systems and presentation on demand, and the systems can be understood and taken over by end users.
- Execution: studies how to integrate data mining findings and systems into production systems, and how the findings can be executed easily and seamlessly in an operational environment.

Supporting techniques need to be developed for AKD presentation, deliverable, and execution. For instance, the following lists some such techniques.

- Presentation: typical tools such as visualization techniques are essentially helpful; visual mining could support the whole data mining process in a visual manner.
- Deliverable: business rules are widely used in business organizations, and one method for delivering patterns is to convert them into business rules; for this, we can develop a tool with underlying ontologies and semantics to support the transfer from pattern to business rules.
- Execution: tools to make deliverables executable in an organization's environment need to be developed; one such effort is to generate PMML to convert models to executables so that the models can be integrated into production systems, and run on a regular basis to provide cases for business management.

4 D^3M Architectures and Techniques

To support the involvement of ubiquitous intelligence and the delivery of actionable knowledge, it is essential to develop effective system architectures for constructing AKD systems, and effective techniques for supporting D^3M.

4.1 D^3M Architectures

From the system viewpoint, we summarize several high-level and general AKD frameworks by introducing their concepts and working mechanisms. More information can be found in [17].

4.1.1 Postanalysis-Based AKD

Postanalysis AKD (PA-AKD) is a two-step pattern extraction and refinement exercise. First, generally interesting patterns (which we call "general patterns") (P) are mined from data sets by 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 ($\widetilde{P}$ and $\widetilde{R}$, 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.

$$PA-AKD: DB \overset{e,t_i(),m_1}{\longrightarrow} P \overset{e,b_i(),m_2,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P}, \widetilde{R}. \quad (13)$$

The key point in this framework is to utilize both domain/meta knowledge and business interestingness in postprocessing the learned patterns. In the real world, this framework can be further instantiated into varied mutations [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/meta knowledge into the AKD process.

4.1.2 Unified-Interestingness-Based AKD

Unified-Interestingness-based AKD (UI-AKD) develops unified interestingness metrics, which are defined for capturing and describing both business and technical concerns. The mined patterns are further converted into deliverables based on domain knowledge and semantics. UI-AKD looks just the same as the normal data mining process except for three inherent characteristics. One is the interestingness system, which combines technical significance ($t_i()$) 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:

$$UI-AKD: DB \overset{e,i(),m,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P}, \widetilde{R}. \quad (14)$$

If $t_i()$ and $b_i()$ are not dependent, thus

$$i() \rightarrow \eta \hat{t_i}() + \varpi \hat{b_i}(). \quad (15)$$

Weights η and ϖ reflect the interestingness balance/trade-off negotiated between data analysts and domain experts in terms of the business problem, data, environment, and deliverable expectation.

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 the two types of measures, 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.

4.1.3 Combined-Interestingness-Based AKD

Combined-Interestingness-based AKD (CM-AKD) comprises multisteps of pattern extraction and refinement on the whole data set. First, J steps of mining are conducted 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 relevant remaining steps to guide the 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. CM-AKD can be formalized as follows:

$$CM-AKD: \underbrace{DB \overset{e,t_{i,j}()[i_{i,j}()],m_j,\Omega_d,\Omega_m}{\longrightarrow} \{P_j\}}_{J} \overset{e,b_{i,j}(),\uplus^J P_j,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P},\widetilde{R}, \tag{16}$$

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.

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). An example is *Multisource + Combined-Mining-Based AKD* (MSCM-AKD) framework. MSCM-AKD can be expressed as follows:

$$MSCM-AKD: \underbrace{DB_n[DB \overset{\otimes}{\longrightarrow} DB_n] \overset{e,t_{i,n}()[u_{i,n}()],m_n,\Omega_m}{\longrightarrow} \{P_n\}}_{N} \overset{e,b_{i,n}(),\uplus^N P_n,\Omega_d,\Omega_m}{\longrightarrow} \widetilde{P},\widetilde{R}, \tag{17}$$

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, and $\otimes$ indicates the data partition if the source data need to be split.

4.2 D^3M Techniques

As we discuss in Section 3, effective techniques need to be developed to tackle many issues in implementing D^3M. In this section, we briefly introduce two techniques: *combined mining* for complex knowledge in complex data, and *agent-driven data mining* for enhancing interaction, coordination, and distributed processing in complex data mining applications.

4.2.1 Combined Mining

Combined Mining is one of the general methods of analyzing complex data for identifying complex knowledge. The deliverables of combined mining are *combined patterns*. For a given business problem (Ψ), we suppose there are the following key entities associated with it in discovering interesting knowledge for business decision support: Data Set $\mathcal{D}$, Feature Set $\mathcal{F}$, Method Set $\mathcal{R}$, Interestingness Set $\mathcal{I}$, Impact Set $\mathcal{T}$, and Pattern Set $\mathcal{P}$. Based on the above variables, a general pattern discovery process can be described as follows: patterns $\mathcal{P}_{n,m,l}$ are identified through data mining method $\mathcal{R}_l$ deployed on features $\mathcal{F}_k$ from a data set $\mathcal{D}_k$ in terms of interestingness $\mathcal{I}_{m,l}$.

$$\mathcal{P}_{n,m,l}: \mathcal{R}_l(\mathcal{F}_k) \rightarrow \mathcal{I}_{m,l}, \tag{18}$$

where $n = 1,\ldots,N; m = 1,\ldots,M; l = 1,\ldots,L$.

From a high-level perspective, combined mining represents a generic framework for mining complex patterns in complex data as follows:

$$\mathcal{P} := \mathcal{G}(\mathcal{P}_{n,m,l}), \tag{19}$$

in which, atomic patterns $\mathcal{P}_{n,m,l}$ from either individual data sources $\mathcal{D}_k$, individual data mining methods $\mathcal{R}_l$, or particular feature sets $\mathcal{F}_k$, are combined into groups with members closely related to each other in terms of pattern similarity or difference.

The *cardinality* of constituent atomic patterns in a combined pattern can be varying. For instance, the following lists two kinds of general structures.

- Pair patterns: $\mathcal{P} ::= \mathcal{G}(P_1, P_2)$, two atomic patterns P_1 and P_2 are correlated to each other in terms of pattern merging method $\mathcal{G}$ into a pair. From such patterns, contrast and emerging patterns [18] can be further identified.
- Cluster patterns: $\mathcal{P} ::= \mathcal{G}(P_1,\ldots,P_n)(n > 2)$, more than two patterns are correlated to each other in terms of pattern merging method $\mathcal{G}$ into a cluster. A group of patterns such as combined association clusters [41] can be further discovered.

In some cases of pair pattern mining, there is a certain relationship between items X_1 and X_2. One situation is $X_2 = X_1 \cup X_p, T_1 \neq T_2$, we then have *incremental pair patterns*. An incremental pair pattern is a special pair of combined patterns as follows:

$$\mathcal{P}: \begin{cases} X_{\mathrm{p}} \rightarrow T_1, \\ X_{\mathrm{p}} \wedge X_{\mathrm{e}} \rightarrow T_2, \end{cases} \tag{20}$$

where $X_{\mathrm{p}} \neq \emptyset$, $X_{\mathrm{e}} \neq \emptyset$, and $X_{\mathrm{p}} \cap X_{\mathrm{e}} = \emptyset$.

With combined mining, atomic patterns or combined patterns can be further organized into *cluster patterns* by putting similar or related patterns together, which can be more informative than their constituent patterns. A *cluster pattern* is defined as follows: Assume there are k atomic patterns $X_i \rightarrow T_i, (i = 1,\ldots,k)$, $k \geq 3$ and $X_1 \cap X_2 \cap \cdots \cap X_k = X_{\mathrm{p}}$, a *cluster pattern* ($\mathcal{P}$) is in the form of

$$\mathcal{P}: \begin{cases} X_1 \rightarrow T_1, \\ \cdots, \\ X_k \rightarrow T_k, \end{cases} \tag{21}$$

where $k > 2, X_{\mathrm{p}}$ is the *prefix* of cluster $\mathcal{P}$.

Similar to incremental pair patterns, for cluster patterns, we have *incremental cluster patterns*. We, here, illustrate the incremental cluster sequences. An incremental cluster sequence is a special cluster of combined patterns with additional items appending to every previously adjacent constituent pattern. An example is as follows:

$$\mathcal{P}: \begin{cases} X_{\mathrm{p}} \rightarrow T_1, \\ X_{\mathrm{p}} \wedge X_{\mathrm{e},1} \rightarrow T_2, \\ X_{\mathrm{p}} \wedge X_{\mathrm{e},1} \wedge X_{\mathrm{e},2} \rightarrow T_3, \\ \cdots, \\ X_{\mathrm{p}} \wedge X_{\mathrm{e},1} \wedge X_{\mathrm{e},2} \wedge \cdots \wedge X_{\mathrm{e,k-1}} \rightarrow T_k, \end{cases} \tag{22}$$

where $\forall i, 1 \leq i \leq k-1$, $X_{i+1} \cap X_i = X_i$, and $X_{i+1} \setminus X_i = X_{e,i} \neq \emptyset$, i.e., X_{i+1} is an *increment* of X_i. The above cluster of rules show the impact of pattern increment on their outcomes.

In combined mining, the word "combined" principally refers to either one or more of the following aspects:

- The combination of multiple data sources ($\mathcal{D}$): combined pattern set $\mathcal{P}$ consists of multiple atomic patterns identified in several data sources, namely $\mathcal{P} = \{\mathcal{P}'_k \mid \mathcal{P}'_k : \mathcal{I}'_k(X_j); X_j \in \mathcal{D}_k\}$; for instance, demographic data and transactional data are two data sets that can be involved in mining for demographic-transactional patterns.
- The combination of multiple features ($\mathcal{F}$): combined pattern set $\mathcal{P}$ involves multiple features, namely $\mathcal{P} = \{\mathcal{F}_k \mid \mathcal{F}_k \subset \mathcal{F}, \mathcal{F}_k \in \mathcal{D}_k, \mathcal{F}_{j+k} \in \mathcal{D}_{j+k}; j, k \neq 0\}$, for instance, features of customer demographics and behavior.
- The combination of multiple methods ($\mathcal{R}$): patterns in the combined set reflect the results mined by multiple data mining methods, namely $\mathcal{P} = \{\mathcal{P}'_k \mid \mathcal{R}'_k \rightarrow \mathcal{P}'_k\}$, e.g., association mining and classification.

Correspondingly, we can have three types of general frameworks supporting combined mining: *multifeature combined mining*, *multimethod combined mining*, and *multisource combined mining*. Let us take Multimethod Combined Mining as an example. The focus of multimethod combined mining is to combine multiple data mining algorithms as needed in order to generate more informative knowledge. In fact, the combination of multiple data mining methods has been recognized as an essential and effective strategy in dealing with complex applications. Multimethod combined mining can be further categorized into Parallel Multimethod Combined Mining and Serial Multimethod Combined Mining.

For instance, suppose we have L data mining methods $\mathcal{R}_l$ $(l = 1, \ldots, L)$, the *serial multimethod combined mining* is a gradual process as follows:

- First, based on the understanding of domain knowledge, data, business environment, and meta knowledge, select a suitable method (say $\mathcal{R}_1$) on the data set $\mathcal{D}$; consequently, we obtain the resulting pattern set $\mathcal{P}_1$:

$$\mathcal{D} \overset{e,\mathcal{R}_l,\mathcal{F}_l,\mathcal{I}_l,\Omega_m}{\longrightarrow} \mathcal{P}_1, or \tag{23}$$

$$\{\mathcal{R}_1, \mathcal{F}_1, \mathcal{I}_1\} \overset{e,\mathcal{D},\Omega_m}{\longrightarrow} \mathcal{P}_1. \tag{24}$$

- Then, supervised by the resulting patterns $\mathcal{P}_1$ and deeper understanding of the business and data during mining P_1, select the second data mining method $\mathcal{R}_2$ to mine $\mathcal{D}$ for pattern set P_2:

$$\{\mathcal{R}_2, \mathcal{F}_2, \mathcal{I}_2\} \overset{e,\mathcal{D},\Omega_m,\mathcal{P}_1}{\longrightarrow} \mathcal{P}_2, \tag{25}$$

  where $\mathcal{P}_1$ contributes to the discovery of $\mathcal{P}_2$.
- Iteratively, select the next data mining method to mine the data with supervision of the corresponding patterns from the previous stages. Repeat this process until the data mining objective is met, and we get the eventual pattern set P.

$$\{\mathcal{R}_L, \mathcal{F}_L, \mathcal{I}_L\} \rightarrow \mathcal{P}. \tag{26}$$

4.2.2 Agent-Driven Data Mining

Multiagent technology is good at user interaction, autonomous computing, self-organization, coordination, cooperation, communication, negotiation, peer-to-peer computing, mobile computing, collective intelligence, and intelligence emergence. These main strengths of multiagent technology can greatly complement data mining, in particular complex data mining problems in aspects such as data processing, information processing, pattern mining, user modeling and interaction, infrastructure, and services. It forms a promising research area, called agent mining [5], [8]. One of its main tasks is agent-driven data mining.

Agent-driven data mining can contribute to the problem solving of many data mining issues, e.g., multiagent data mining infrastructure and architecture, multiagent interactive mining, multiagent-based user interaction, automated pattern mining, multiagent distributed data mining, multiagent dynamic mining, multiagent mobility mining, multiagent multiple data source mining, multiagent peer-to-peer data mining, and multiagent web mining. In the following, we discuss the unique roles of agents in supporting DDM.

In enterprise applications, data are distributed in heterogeneous sources coupling in either a tight or loose manner. Distributed data sources associated with a business line are often complex, for instance, some is of high frequency or density, mixing static and dynamic data, or mixing multiple structures of data. In some cases, multiple sources of data are stored in parallel storage systems. Local data sources can be of restricted availability due to privacy, their commercial value, and the like, which also prevents, in many cases, its centralized processing even in a collaborative mode. For such data, data integration and data matching are difficult to conduct. It is not possible to store them in centralized storage and not feasible to process them in a centralized manner. To mine the data, the infrastructure and architecture of existing distributed data mining systems requires more flexible, intelligent, and scalable enhancement.

Agent technology can help with these challenges by involving autonomy, interaction, dynamic selection and gathering, scalability, multistrategy, and collaboration. Other challenges include privacy, mobility, time constraint (stream data, it is too late to extract and then mine), and computational costs and performance requests. In particular, multiagent technology can complement distributed data mining in many aspects, for instance,

- Distributed and multiple data sources are often isolated from each other. For in-depth understanding of a business problem, it is essential to bring relevant data together through centralized integration or localized communication. From this, agent planning and collaboration, mobile agents, agent communication, and negotiation can benefit.
- Data and device mobility require the perception and action of data mining algorithms on a mobile basis. Mobile agents can adapt to mobility very well.
- Proactively assisting an agent is necessary to drastically limit how much the user has to supervise and interfere with running the data mining process.
- In changing and open distributed environment, KDD agents may be applied to adaptively select

data sources according to given criteria such as the expected amount, type, and quality at the considered source, actual network, and KDD server load. Agents may be used, for example, to dynamically control and manage the process of data gathering.
- Some data distributed in different storages are dependent on time, e.g., time difference.
- For some complex application settings, an appropriate combination of multiple data mining techniques may be more beneficial than applying a particular one. KDD agents may learn in, due course which of their deliberative actions to choose, depending on the type of data retrieved from different sites and the mining tasks to be pursued.
- KDD agents may operate independently on data gathered at local sites and then combine their respective models. Alternatively, they may agree to share potential knowledge as it is discovered, in order to benefit from the additional options of other agents.
- Distributed local data are not allowed to be extracted and integrated with other sources directly, due to privacy issues. A KDD agent with authority to access and process the data locally can dispatch identified local patterns for further engagement with findings from other sources.
- In some organizations, business logic, process, and work-flow determine the order of data storage and access. This, therefore, augments the complexity of DDM. Agents located in each storage area can communicate with each other and dispatch the DDM algorithm agents instantly, once the response is over.

In fact, agent-driven data mining provides a unique approach to involve, represent, and tackle

1. data intelligence such as agent-based distributed data access and collaboration,
2. human intelligence such as through user-agent interaction, and user modeling and servicing,
3. domain and organizational factors such as through multiagent swarm intelligence, collective intelligence, and intelligence emergence,
4. network intelligence such as through mobile agents and multiagent coordination and communication, and
5. social intelligence such as building multiagent social cognition and interaction to involve a group of experts in the data mining process.

5 Case Studies

This section illustrates several case studies of D^3M in the real world. Sections 5.1 and 5.2 illustrate the processing of in-depth data intelligence aiming for combined patterns in social security data by multisource combined mining and multimethod combined mining, respectively. The identified combined patterns are more informative than any single sort of pattern identified by traditional methods. Section 5.3 illustrates the involvement of domain intelligence for dealing with complex sequences by utilizing UI-based AKD and agent mining for coupled trading sequences in stock markets. The identified exceptional coupled sequences satisfy both technical and business interestingness. Section 5.4 discusses the involvement of domain, organizational, and social factors in identifying actionable trading strategies in stock markets. The identified trading strategies satisfy business performance.

5.1 Mining Combined Patterns in Social Security

Real-life data often involve multiple sources of information. This case study illustrates the use of multisource combined mining in Section 4.2.1 for deep data intelligence, in particular, incremental pair patterns in debt-related activities in the social security area. The exercise is conducted on four data sources: activity files recording activity details, debt files logging debt details, customer files enclosing customer circumstances, and earnings files storing earnings details. Zhao et al. [41] and Cao et al. [13] introduced approaches for identifying incremental pair patterns and cluster patterns consisting of multiple features from various data sources.

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

$$\{u = age : 65+, v = withholding \ \& \ irregular + withholding \rightarrow c = C\}. \quad (27)$$

5.2 Identifying Sequence Classifiers in Social Security

Multimethod combined mining can disclose more informative patterns in complex data. This case study illustrates the combination of sequential pattern mining with classification to develop sequential classifiers to identify discriminating interactive activities associated with government debt in the social security area. We propose a novel closed-loop sequence classification method. First, a small set of most discriminating sequential patterns are mined. These patterns are then used for coverage test on the training data set. If the sequential pattern set is small enough, there must be some samples that have not been covered by the mined patterns. These uncovered samples are further fed back to the next loop of sequential pattern mining. Again, a coverage test is implemented on the newly mined patterns. The remaining samples that cannot still be covered are fed back for sequential pattern mining until the predefined thresholds are reached or all samples are covered.

The proposed closed-loop sequence classification algorithms are tested on Centrelink data involving debt life cycle, debt generation business processes, and Centrelink interventions on debt. We combine sequential pattern mining with classification to build a sequential classifier that can predict whether a customer tends to result in having

TABLE 2
The Performance of Sequence Classification Algorithms

min_sup	$No.Pattern$	CCR_{CMAR}	$CCR_{Highest}$	CCR_{Multi}
1%	39,220	75.0%	72.7%	75.2%
2%	10,254	74.4%	71.8%	74.9%
5%	1,116	69.4%	70.9%	72.4%
10%	208	64.2%	61.0%	66.7%

debt or not. In Table 2, at all min_sup levels, CCR_{Multi} outperforms $CCR_{Highest}$. It verifies that the classifier only using the highest ranking pattern for one instance suffers from overfitting. Between the two algorithms both using multiple patterns for one instance, CCR_{Multi} and CCR_{CMAR}, CCR_{Multi} outperforms CCR_{CMAR} at all min_sup levels. When min_sup becomes greater, the difference between the two algorithms increases, which means our algorithm is more robust than CCR_{CMAR} when fewer patterns are discovered for classification. More technical details can be found in [39].

5.3 Detecting Abnormal and Dynamic Coupled Behavior in Stock Markets

Many applications involve complex sequences, for instance, multiple sequences couple with each other. An even more challenging situation is the change of data. This example identifies deep data intelligence in such coupled and changing data. For instance, in capital markets, there are multiple activity streams, such as *trade* stream, *buy order* stream, and *sell order* stream. Such streams hide information about abnormal trading behavior, for instance, manipulating markets. This case study shows the use of combined-interestingness-based architecture which combines technical and business performance, and the use of agent mining for adaptive knowledge discovery that can adapt to data changes.

We use the Coupled Hidden Markov Model (CHMM) to model multiple activity streams as multiple processes, reflecting the relationships among the streams, and the transitions between hidden states and observations. This makes it possible for us to observe the state change of an individual stream, as well as the correlation among relevant streams.

$$\lambda^{CHMM} = (A, B, C, \pi). \tag{28}$$

Furthermore, we propose an Adaptive CHMM (ACHMM) driven by agents. A pattern Change Detection Agent detects changes in the output of CHMM, and then the Planning Agent triggers the adjustment and retraining of the CHMM model to adapt to the source data dynamics. ACHMM is implemented through a closed-loop system, which consists of CHMM, Change Detection Agent, Model Adjusting Agent, and Planning Agent. The adaptation is mainly based on the detection of change between the identified patterns and the benchmark ones. The benchmark patterns are those updated most recently which reflect the pattern change. The working process of ACHMM is as follows:

PROCESS: Agent-based abnormal behavior pattern discovery by ACHMM
INPUT: Orderbook transactions $\mathcal{D}$, domain knowledge Ψ
OUTPUT: Abnormal behavior patterns $\mathcal{P}$
Step 1: Stream Extraction Agent splits orderbook transactions into sequences $\mathcal{D}_k$ $(k = 1, \ldots, K)$ in terms of market microstructure theory (Ψ). In our case, we get Buy_order stream D_B, Sell_order stream D_S, and Trade stream D_T;
Step 2: Training CHMM-based model $\mathcal{R}_k$ on stream D_k, respectively, or on the selected streams $\{ D_i, \ldots, D_k\}$;
Step 3: Change Detection Agent detects the change of the identified patterns compared with the benchmarks;
IF there is any significant change;
Informing the Planning Agent;
Triggering Model Adjusting Agent to adjust CHMM models;
Retraining CHMM models, reextracting streams if necessary;
Testing CHMM-based models;
ENDFOR
Step 5: Retraining the CHMM model;
Step 6: Testing the CHMM model;
Step 7: Exporting final pattern set $\mathcal{P}$.

We test the ACHMM on a real data set from a stock exchange. The data set covers 388 trading days from June 2004 to December 2005. We use the data from June 2004 to December 2004 as training data, and the data from January 2005 to December 2005 as test data. Fig. 1 shows the *accuracy*, *precision*, *recall*, *specificity*, *return*, and *abnormal return* of CHMM, ACHMM compared to three single HMMs for trades (HMM-T), buy orders (HMM-B) and sell orders (HMM-S), and a CHMM combining the three single HMMs (IHMM). It shows that CHMM and ACHMM can always outperform any single HMM and IHMM in terms of both technical and business performance.

5.4 Developing Actionable Trading Strategies in Stock Markets

Trading strategies could assist traders in making a profit in stock markets. To make them actionable, organizational and domain factors in stock markets need to be considered, and the business performance by implementing such trading strategies should be checked. Market organization factors consist of the following fundamental entities: $M = \{I, A, O, T, R, E\}$, in which I represents traded instruments, A represents market participants, O represents order book forms, T represents time frame, R represents market rules, and E represents execution system. $O = \{(t, b, p, v) \mid t \in T, b \in B, p \in P, v \in V\}$ is further represented by attributes T, behavior B, price P, and volume V, which are attributes of trading strategy set Ω. The elements in M form the constrained market environment of trading strategy optimization. Σ is a Constraint set on Ω.

$$\Sigma = \left\{\delta_i^k \;\middle|\; c_i \in C, 1 \le k \le N_i\right\}, \tag{29}$$

where δ_i^k stands for the kth constraint attribute of a constraint type c_i, $C = \{M, D\}$ is a constraint type set covering all types of constraints in market microstructure M and data D in the searching niche, and N_i is the number of constraint attributes for a specific type c_i.

Correspondingly, actionable trading strategy set Ω' is a conditional function of Σ, which is described as:

$$\Omega' = \left\{(\omega, \delta) \mid \omega \in \Omega, \delta \in \left\{\left(\delta_i^k, a\right) \;\middle|\; \delta_i^k \in \Sigma, a \in A\right\}\right\}, \tag{30}$$

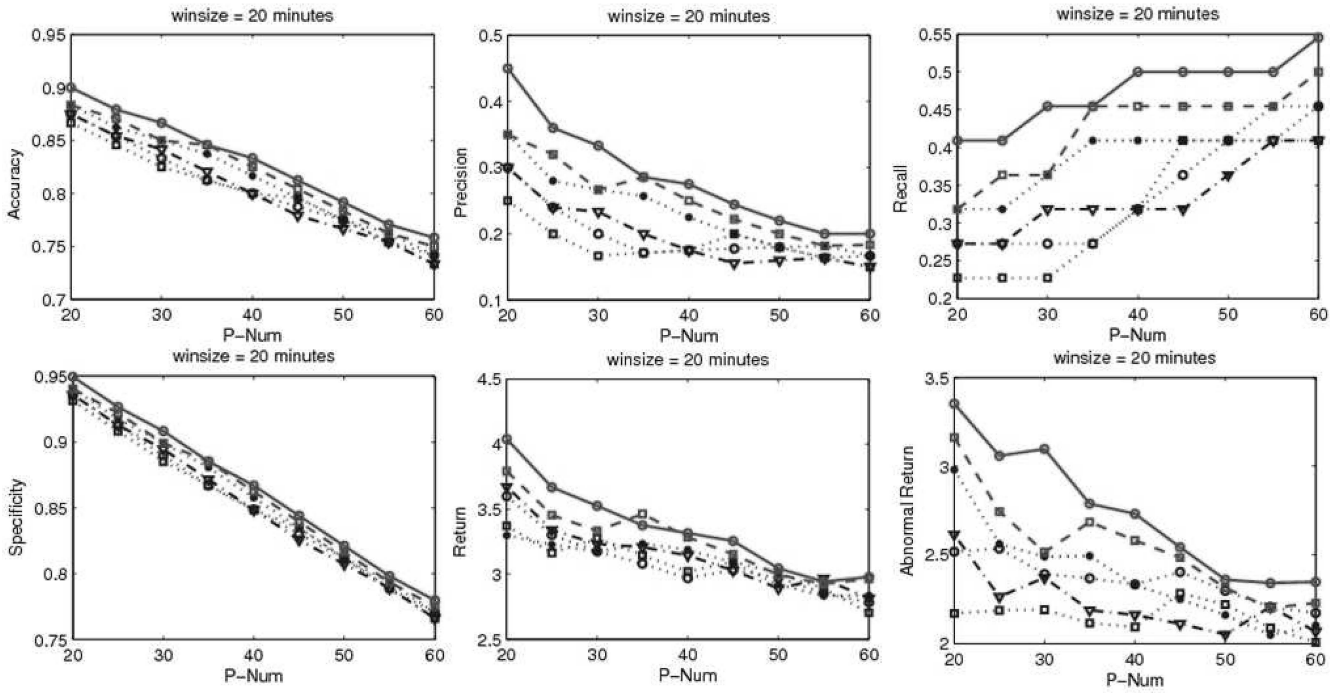

Fig. 1. Technical and business performance of six models.

where ω is an "optimal" trading pattern instance, and δ indicates specific constraints on the discovered pattern recommended to a trading agent a.

Another consideration is trader preference. Trader preferences may be embodied in terms of achieving high benefit but as low cost and risk as possible when taking certain trading positions. As the representative of traders, trading agents should target those positions with high benefit per cost.

Based on the above principle, we develop approaches for searching, optimizing, identifying, and integrating actionable trading strategies [9], [26], [4]. For instance, we can generate golden trading strategies to obtain higher benefit-cost ratios, and then construct collaborative trading agents concurrently executing positions recommended by individual golden strategies, which can greatly increase benefits while controlling very low costs compared with those taking positions recommended by either an individual strategy or randomly chosen strategies only. Table 3 shows the positions recommended by each golden strategy identified by *Collaborative Trading Agents* in 2006 Hong Kong United Exchange data.

In Table 4, *lift* measures how good a trading strategy is.

TABLE 3
Trading Agent Positions Recommended by Five Trading Strategy Classes (Excerpt)

Date	MA Pos	FR Pos	CB Pos	SR Pos	OBV Pos
2006-11-16	1	1	0	1	1
2006-11-17	1	1	0	1	1
2006-11-20	1	1	0	1	1
2006-11-21	-1	-1	0	1	1
2006-11-22	-1	-1	0	1	1

6 Open Issues and Trends

While D^3M opens great opportunities for the paradigm shift from data-centered hidden pattern discovery to domain-driven actionable knowledge delivery, there are many fundamental problems that require investigation. These issues have either existed previously in data mining or emerge in real-life applications and systems.

- How to involve domain knowledge in the data mining modeling process?
- How to involve human qualitative intelligence such as imaginary thinking in interactive data mining?
- How to involve expert groups in complex data mining applications?
- How to involve network resources in active and runtime mining?
- How to support social interaction and cognition in data mining?
- How to build an intelligent data mining infrastructure that can synthesize ubiquitous intelligence?
- How to measure knowledge actionability, and balance technical significance with business expectation when multiple objectives are involved?
- How to make data mining operable, executable, repeatable, and trustful in an easy and business-friendly manner?

TABLE 4
Lift Comparison between Random Chosen Strategies and Golden Strategies

Lift	MA-CMN	FR-XY	OBV-B	CB-NXC	SR-NC
Random	10%	0	20%	10%	10%
Optimized	70%	80%	80%	90%	100%

- How to present and deliver data mining findings that are executable and interpretable in production systems?

Accordingly, many open issues await further research investigation. For instance, to effectively synthesize the ubiquitous intelligence in actionable knowledge discovery, many research issues need to be studied or revisited.

- Typical research issues and techniques in *Data Intelligence* include mining in-depth data patterns disclosing deep knowledge in data and context, combined patterns consisting of information from heterogeneous sources, and structured knowledge in mix-structured data.
- Typical research issues and techniques in *Domain Intelligence* consist of static and dynamic representation, modeling and involvement of domain knowledge, constraints, organizational factors, and business interestingness into models and the AKD process, and of both syntactic and semantic aspects.
- Typical research issues and techniques in *Network Intelligence* include involving not only general techniques such as information retrieval, text mining, web mining, web intelligence into data mining, but also the involvement of web/networked facilities to support web-based data mining, the discovery of networks and communities in networked data, and web/network knowledge management in the data mining process and systems.
- Typical research issues and techniques in *Human Intelligence* include representation and involvement of empirical and implicit knowledge, reasoning and situated computing capabilities, as well as runtime human-machine interaction into data mining process and models at both individual and group levels.
- Typical research issues and techniques in *Social Intelligence* include collective intelligence, social network analysis, and social cognition interaction in data mining systems, building social data mining software that can cater for ubiquitous intelligence in a social context.
- Typical issues in *intelligence metasynthesis* consist of building metasynthetic interaction as the working mechanism, and metasynthetic space (m-space) as a data-mining-based problem-solving system.

In addition, many specific issues need to be studied as required, for instance,

- The integration of multiagents with data mining for involving and employing ubiquitous intelligence through agent-based AKD systems and m-spaces.
- The next-generation data mining methodologies, frameworks, and processes toward actionable knowledge discovery.
- Effective principles and techniques for acquiring, representing, modeling, and engaging intelligence in real-world data mining through automated, human-centered means, and/or a human-machine-cooperated manner.
- Workable and operable tools and systems balancing technical significance and business concerns, and delivering actionable knowledge expressed as operable business rules seamlessly engaging business processes and systems.
- Tools supporting the seamless integration and immediate execution of delivered models and systems in the production environment.
- Project management methodologies for governing domain-driven AKD projects.
- Techniques supporting dynamic mining, evolutionary mining, real-time stream mining, and domain adaptation.
- Techniques for enhancing reliability, trust, cost, risk, privacy, utility, and other organizational/social issues.
- Techniques for handling inconsistencies between, and seamless integration of, the mined knowledge and the existing domain knowledge.

7 Conclusion

Based on our experience in conducting large-scale data analysis for several domains, such as finance data mining and social security mining, we have proposed the D^3M methodology for the real-world data mining problem solving. D^3M emphasizes the development of methodologies, techniques, and tools for *actionable knowledge discovery and delivery* by incorporating relevantly ubiquitous intelligence surrounding data-mining-based problem solving. Ubiquitous intelligence consists of in-depth data intelligence, human intelligence, domain intelligence, network intelligence, and organizational/social intelligence. It is essential to synthesize such ubiquitous intelligence in actionable knowledge discovery and delivery.

This paper has presented an overview of D^3M by systematically addressing a series of aspects, including challenges facing traditional data mining methodologies, the fundamental framework and relevant techniques for D^3M. As the analysis has indicated, D^3M has demonstrated that there are plenty of opportunities for bridging the gap between technical and business expectations, and in handling the extreme imbalance existing in data mining research and development, which is so critical for dealing with complex data mining applications and systems and for promoting the widespread use of data mining. There are many promising theoretical and practical topics and issues awaiting further investigation through cross-disciplinary effort. The development of D^3M methodologies and techniques has the potential to contribute to a paradigm shift from "data-centered knowledge discovery" to "domain-driven, actionable knowledge delivery."

Acknowledgments

Thanks are given to Dr. Yanchang Zhao, Dr. Huaifeng Zhang, Mr. Yuming Ou, as well as the guest editors and reviewers, and others for their comments, discussions, and/or experiments on the case studies. This work is sponsored in part by Australian Research Council Discovery Grants (DP1096218, DP0988016, DP0773412) and ARC Linkage Grant (LP0989721, LP0775041).

REFERENCES

[1] 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.

[2] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*. Springer, 2008.

[3] L. Cao, "Domain-Driven Actionable Knowledge Discovery," *IEEE Intelligent Systems,* vol. 22, no. 4, pp. 78-89, July 2007.

[4] L. Cao, "Developing Actionable Trading Strategies," *Knowledge Processing and Decision Making in Agent-Based Systems,* pp. 193-215, Springer, 2008.

[5] *Data Mining and Multi-Agent Integration,* L. Cao, ed. Springer, 2009.

[6] L. Cao and R. Dai, *Open Complex Intelligent Systems,* Post & Telecom, 2008.

[7] L. Cao, R. Dai, and M. Zhou, "Metasynthesis: M-Space, M Interaction and M-Computing for Open Complex Giant Systems," *IEEE Trans. SMC—Part A,* vol. 39, no. 5, pp. 1007-1021, Sept. 2009.

[8] L. Cao, V. Gorodetsky, and P. Mitkas, "Agent Mining: The Synergy of Agents and Data Mining," *IEEE Intelligent Systems,* vol. 24, no. 3, pp. 64-72, May/June 2009.

[9] L. Cao and T. He, "Developing Actionable Trading Agents," *Knowledge and Information Systems: An Int'l J.,* vol. 18, no. 2, pp. 183-198, 2009.

[10] L. Cao and C. Zhang, "Domain-Driven Data Mining: A Practical Methodology," *Int'l J. Data Warehousing and Mining,* vol. 2, no. 4, pp. 49-65, 2005.

[11] 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, 2006.

[12] 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.

[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 Y. Ou, "Market Microstructure Pattern Analysis for Powering Trading and Surveillance Agents," *J. Universal Computer Science,* vol. 14, no. 14, pp. 2288-2308, 2008.

[15] *Data Mining for Business Applications,* L. Cao, P. Yu, C. Zhang, and H. Zhang, eds. Springer, 2008.

[16] L. Cao, P. Yu, C. Zhang, and Y. Zhao, *Domain Driven Data Mining.* Springer, 2009.

[17] L. Cao, Y. Zhao, H. Zhang, D. Luo, and C. Zhang, "Flexible Frameworks for Actionable Knowledge Discovery," *IEEE Trans. Data and Knowledge Eng,* preprint, 4 June 2009, doi: 10.1109/TKDE.2009.143.

[18] G. Dong and L. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD '99),* pp. 43-52, 1999.

[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, 1996.

[21] A. Freitas, "On Objective Measures of Rule Surprisingness," *Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '98),* pp. 1-9, 1998.

[22] H. Kargupta, B. Park, D. Hershbereger, and E. Johnson, "Collective Data Mining: A New Perspective toward Distributed Data Mining," *Advances in Distributed and Parallel Knowledge Discovery,* MIT/AAAI Press, 2000.

[23] 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.

[24] R. Hilderman and H. Hamilton, "Applying Objective Interestingness Measures in Data Mining Systems," *Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '00),* pp. 432-439, 2000.

[25] B. Lent, A.N. Swami, and J. Widom, "Clustering Association Rules," *Proc. Int'l Conf. Data Eng. (ICDE '97),* pp. 220-231, 1997.

[26] L. Lin and L. Cao, "Mining In-Depth Patterns in Stock Market," *Int'l J. Intelligent System Technologies and Applications,* vol. 4, nos. 3/4, pp. 225-238, 2008.

[27] B. Liu, W. Hsu, and Y. Ma, "Pruning and Summarizing the Discovered Associations," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD '99),* pp. 125-134, 1999.

[28] B. Liu, "Analyzing the Subjective Interestingness of Association Rules," *IEEE Intelligent Systems,* vol. 15, no. 5, pp. 47-55, Sept./Oct. 2000.

[29] E. Omiecinski, "Alternative Interest Measures for Mining Associations in Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 1, pp. 57-69, Jan./Feb. 2003.

[30] B. Padmanabhan and A. Tuzhilin, "A Belief-Driven Method for Discovering Unexpected Patterns," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* pp. 94-100, 1998.

[31] B. Park and H. Kargupta, "Distributed Data Mining: Algorithms and Systems, Applications," *Data Mining Handbook,* N. Ye, ed., 2002.

[32] X. Qian, J. Yu, and R. Dai, "A New Scientific Field: Open Complex Giant Systems and the Methodology," *Chinese J. Nature,* vol. 13, no. 1, pp. 3-10, 1990.

[33] X.S. Qian and H.S. Tsien, "Revisiting Issues on Open Complex Giant Systems," *Pattern Recognition and Artificial Intelligence,* vol. 4, no. 1, pp. 5-8, 1991.

[34] A. Silberschatz and A. Tuzhilin, "On Subjective Measures of Interestingness in Knowledge Discovery," *Knowledge Discovery and Data Mining,* vol. 8, no. 6, pp. 275-281, 1995.

[35] A. Tzacheva and Z. Ras, "Action Rules Mining," *Int'l J. Intelligent Systems,* vol. 20, no. 7, pp. 719-736, 2005.

[36] K. Wang, S. Zhou, and J. Han, "Profit Mining: From Patterns to Actions," *Proc. Int'l Conf. Extending Database Technology (EBDT),* 2002.

[37] G. Williams and Z. Huang, "Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases," *Lecture Notes in Artificial Intelligence,* pp. 340-348, Springer, 1997.

[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] H. Zhang, Y. Zhao, L. Cao, C. Zhang, and H. Bohlscheid, "Customer Activity Sequence Classification for Debt Prevention in Social Security," to be published in *J. Computer Science and Technology.*

[40] *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction,* Y. Zhao, C. Zhang, and L. Cao, eds. IGI Press, 2008.

[41] Y. Zhao, H. Zhang, L. Cao, C. Zhang, and H. Bohlscheid, "Combined Pattern Mining: From Learned Rules to Actionable Knowledge," *Proc. Australasian Joint Conf. Artificial Intelligence (AI '08),* pp. 393-403, 2008.

[42] *Web Intelligence,* N. Zhong, J. Liu, and Y.Y. Yao, eds. Springer, 2003.

Longbing Cao is a professor at the University of Technology, Sydney, Australia. He is the director of the Data Sciences and Knowledge Discovery Lab, and the Data Mining Research Leader of the Australian Capital Markets Cooperative Research Centre. His research interests include data mining, multiagent technology, agent mining, and behavior informatics. He has real-life experience on customer analytics, behavior analytics, fraud detection, compliance analysis and risk analysis, in areas such as telecommunication, capital markets, government services, banking, and insurance. He is a senior member of the IEEE.