Domain-Driven, Actionable Knowledge Discovery

Longbing Cao, *University of Technology, Sydney*

The complexities of real-world domain problems pose great challenges in the knowledge discovery and data mining (KDD) field. For example, existing technologies seldom deliver results that businesses can act on directly. In this issue of Trends & Controversies, seven short articles report on different aspects of domain-driven KDD, an R&D area that targets the development of effective methodologies and techniques for delivering actionable knowledge in a given domain, especially business.

To begin, my colleague Chengqi Zhang and I propose a framework that boosts data mining capabilities and dependability by synthesizing domain-related intelligence into the development KDD process.

In the second article, Qiang Yang looks at ways of getting data mining output models to correspond to actions. The framework he proposes takes traditional KDD output as input to a process that, in turn, generates actionable output.

The third and fourth articles describe different aspects of visualization for clarifying data patterns and meaning. David Bell describes two video and vision system applications that strengthen data mining practice in general. Michail Vlachos, Bahar Taneri, Eamonn Keogh, and Philip S. Yu present a simple, fast method for visualizing DNA transcripts to enhance data mining utility.

Human intelligence plays important roles in actionable data mining. Ning Zhong proposes a methodology for processing multiple data sources in a systems approach to brain studies.

Privacy imposes a strong constraint on data mining. Mafruz Zaman Ashrafi and David Taniar review the issues and techniques for preserving privacy in aggregated data mining.

Finally, Eugene Dubossarsky and Warwick Graco describe four aspects of moving data mining from a method-driven approach to a process that focuses on domain knowledge. —*Longbing Cao*

Domain-Driven Data Mining: A Framework

Longbing Cao and Chengqi Zhang, *University of Technology, Sydney*

Data mining increasingly faces complex challenges in the real-life world of business problems and needs.[1] The gap between business expectations and R&D results in this area involves key aspects of the field, such as methodologies, targeted problems, pattern interestingness, and infrastructure support. Both researchers and practitioners are realizing the importance of domain knowledge to close this gap and develop actionable knowledge for real user needs.

What is domain-driven data mining?

Domain-driven data mining generally targets actionable knowledge discovery in complex domain problems.[2] It aims first to utilize and mine many aspects of intelligence—for example, in-depth data, domain expertise, and real-time human involvement as well as process, environment, and social intelligence. It metasynthesizes its intelligence sources for actionable knowledge discovery. To achieve this metasynthesis, domain-driven data mining must develop knowledge actionability, enhance knowledge reliability, and interact with methodologies and systems that support existing business uses.

Domain-driven data mining works to expose next-generation methodologies for actionable knowledge discovery, identifying how KDD can better contribute to critical domain problems in theory and practice. It uncovers domain-driven techniques to help KDD strengthen business intelligence in complex enterprise applications. It discloses applications that effectively deploy domain-driven data mining to solve complex practical problems. It also identifies challenges and directions for future R&D in the dialogue between academia and business to achieve seamless migration into business world.

Determining what knowledge to pursue requires both technical and business interests to instantiate both objective and subjective factors.[3] Actionable knowledge discovery should fit the following framework:

$$\forall\, x \in X, \exists\, P\colon x.tech\_obj(P) \wedge x.tech\_subj(P) \wedge x.biz\_obj(P) \wedge biz\_subj(P) \rightarrow act(P)$$

where P indicates pattern interestingness from not only technological and business viewpoints but also objective and subjective perspectives.[4]

Why do we need it?

Traditional KDD is a data-driven trial-and-error process that targets automated hidden knowledge discovery. Researchers commonly use it to let data create and verify their innovations or to develop and demonstrate the use of novel algorithms and methods in discovering knowledge of research interest.

In business, however, KDD must support commercial actions. In addition to business rules, policies, and so on, it

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

Table 1. Data-driven versus domain-driven data mining.

Aspects	Traditional data-driven	Domain-driven
Object mined	Data tells the story	Data and domain tell the story
Aim	Develop innovative approaches	Generate business impacts
Objective	Algorithms are the focus	Solving business problems is the target
Data set	Mining abstract and refined data sets	Mining constrained real-life data
Extendability	Predefined models and methods	Ad hoc, runtime, and personalized model customization
Process	Data mining is an automated process	Humans are integral to the data mining process
Evaluation	Based on technical metrics	Based on actionable options
Accuracy	Results reflect solid theoretical computation	Results reflect complex context in a kind of artwork
Goal	Let data create and verify research innovation; demonstrate and push novel algorithms to discover knowledge of research interest	Let data and metasynthetic knowledge tell the hidden business story; discover actionable knowledge to satisfy real user needs

must take into account such real-world phenomena as evolving scenarios, constrained environments, runtime mining, and distributed and heterogeneous data sources. It must support business requirements for trustworthy, reliable, and cost-effective performance. It must also find ways to integrate human intelligence seamlessly in its processes.

Key issues

Operating on top of a data-driven framework, domain-driven data mining aims to develop specific methodologies and techniques for dealing with these business complexities. This goal can mean developing generic frameworks, domain-specific approaches, or both. Other key issues include

- domain-driven project management;
- actionable knowledge discovery frameworks;
- capturing, representing, and using network intelligence;
- mining in-depth patterns and deep data intelligence; and
- balancing the conflicts between technical performance and business interest.

Table 1 summarizes the differences between data-driven and domain-driven data mining.

Applications

We've used domain-driven data mining in real-world trade-support assignments and for analyzing exceptional behavior in government social security overpayments.[4,5] In one case, we integrated domain intelligence into the automated construction of activity sequences in government-customer contacts. We also discovered high-impact activity-sequence patterns associated with government-customer debt and identified customers and events that would likely prevent or recover debt. To this end, we developed both technical and business measures for patterns relevant to these issues in real, unbalanced social security data. For instance, through metasynthesizing intelligence sources, we get the following interestingness of an activity sequence associated with the occurrence of government debt:

- Technical interestingness: support = 0.01251, confidence = 0.60935, and lift = 1.2187;
- Business interestingness: $\overline{d\_amt}(\ ) = 29{,}526$, the averaged debt amount in cents of those debt-related activity sequences supporting the rule; and $\overline{d\_dur}(\ ) = 15.5$, the averaged debt duration in days of those debt-related activity sequences supporting the rule.

These measures tell us that the selected pattern has not only technical interest but also business impact on debt amount and duration.

Conclusion

Domain-driven KDD represents a paradigm shift from a research-centered discipline to a practical tool for actionable knowledge. Despite many open issues, deployed systems are already showing ways to transmit reliable research in forms that satisfy business needs with direct support for decisions.

References

1. 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, 2003, pp. 191–196.
2. L. Cao and C. Zhang, "Domain-Driven Data Mining: A Practical Methodology," *Int'l J. Data Warehousing and Mining*, vol. 2, no. 4, 2006, pp. 49–65.
3. A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems?" *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 6, 1996, pp. 970–974.
4. 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, 2007, pp. 677–692.
5. L. Cao, Y. Zhao, and C. Zhang, "Mining Impact-Targeted Activity Patterns in Imbalanced Data," to be published in *IEEE Trans. Knowledge and Data Eng*.

Learning Actions from Data Mining Models

Qiang Yang, *Hong Kong University of Science and Technology*

Data mining and machine learning algorithms aim mostly at generating statistical models for decision making. There are many techniques for computing statistical models from data: Bayesian probabilities, decision trees, logistic and linear regression, kernel and support-vector machines, and cluster and association rules, among others.[1,2] Most techniques represent algorithms that summarize training-data distributions in one way or another. Their output models are typically mathematical formulas or classification results describing test data. In other words, they're *data centric*.

Despite much industrial success, these models don't correspond to actions that will bring about desired world states. They maximize their utility on test data. But data mining methods should do more than produce a model. They should generate actions that can be executed either automatically or semiautomatically. Only in this way can a data mining system be truly considered actionable.

I've developed two techniques that highlight a novel computational framework for actionable data mining.

Extracting actions from decision trees

The first technique uses an algorithm for extracting actions from decision trees such that each test instance falls in a desirable state.[3] A customer-relationship-management problem illustrates our solution. CRM industry competition has heated up in recent years. Customers increasingly switch service providers. To stay profitable, CRM companies want to convert valuable customers from a likely attrition state to a loyal state.

Our approach to this problem exploits decision tree algorithms. These learning algorithms, such as ID3 or C4.5,[1] are among the most popular predictive data-classification methods. In CRM applications, we can build a decision tree from an example customer set described by a feature set. The features can include any kind of information: personal, financial, and so on.

Let's assume an algorithm has already generated a decision tree. To generate actions from this tree, we must first consider how to extract actions when no restrictions exist on their number. In the training data, some values under the class attribute are more desirable than others. For example, in a banking application, a customer loyalty status of "stay" is more desirable than "not stay." For each test data instance—that is, for each customer under consideration—we want to decide a sequence of actions that would transform a customer from "not stay" to "stay" status. We can extract this set of actions from the decision trees, using the following algorithm:

1. Import customer data using appropriate data collection, cleaning, and preprocessing techniques, and so on.
2. Build customer profiles from the training data, using a decision-tree learning algorithm, such as C4.5,[1] to predict whether a customer is in the desired status. Refine the algorithm to use the area under the receiver-operating-characteristic curve.[4] (The ROC curve lets us evaluate candidate-predication ranking instead of accuracy.) The Laplace correction is a further refinement to avoid extreme probability values.
3. Search for optimal actions for each customer. This critical step generates actions, and I describe it in more detail later.
4. Produce reports for domain experts to review the actions and selectively deploy them.

In this algorithm, when a customer falls into a particular leaf node with a certain probability of having the desired status, the algorithm tries to "move" the customer into other leaves with higher probabilities of having the desired status. A data analyst can then convert the probability gain into an expected gross profit. However, moving a customer from one leaf to another means changing some of the customer's attribute values. The data analyst generates an action to denote each such change in which an attribute A's value is transformed from v_1 to v_2. Furthermore, these actions can incur costs, which a domain expert defines in a cost matrix.

> Despite much industrial success, data-centric output models don't correspond to actions that will bring about desired world states.

On the basis of a domain-specific cost matrix for actions, we can define an action's net profit:

$$P_{\text{net}} = P_E \times P_{\text{gain}} - \sum_i \text{Cost}_i$$

where P_{net} denotes the net profit, P_E denotes the total profit of having the customer in the desired status, P_{gain} denotes the probability gain, and Cost_i denotes each action's cost. The algorithm has taken into account both continuous and discrete attribute versions.

This solution corresponds to a simple situation, which limits the number of actions. However, when we limit actions by number or total costs, the problem of selecting the action subset to execute becomes NP-hard. In an extension of this work, I've developed a greedy algorithm that can give a high-quality approximate solution.[3] My colleagues and I have run many tests to show that these methods are useful in action generation and performance. Furthermore, we've also considered a decision-tree ensemble to make the solutions more robust.

Learning from frequent-action sequences

The second technique uses an algorithm that can learn relational action models from frequent item sets. This technique applies to automatic planning systems, which often require formally defined action models with an initial state and a goal. However, building such models from scratch is difficult, so many researchers have explored various approaches to learning them from examples instead.

The Action-Relation Modeling System automatically acquires action models from recorded user plans.[5] ARMS takes a collection of observed traces as its input. It determines a collection of frequent-action sets by applying a frequent-item-set-mining algorithm to the traces. It then takes these sets as the input to another modeling system, called *weighted Max-Sat*, which can generate relational actions.

To better understand relational-action representations, consider an example input and output in the Depot problem domain from an AI planning competition.[6] The training data consists of training plan samples that are similar to this (abbreviated) sequence of actions: drive(?x:truck ?y:place ?z:place), lift(?x:hoist ?y:crate ?z:surface ?p:place), drop(?x:hoist ?y:crate ?z:surface ?p:place), load(?x:hoist ?y:crate ?z:truck ?p:place), unload(?x:hoist ?y:crate ?z:truck ?p:place), where "?" indicates variable parameters.

From such input sequences, we want to learn the preconditions, add lists, and delete lists for all actions. When ARMS completes the three lists for an action, its action model is complete. We want to learn an action model for every action in a problem domain. In this way, we "explain" all

training examples successfully. For example, a learned action model for the action load(?x:hoist ?y:crate ?z:truck ?p:place) might have the preconditions (at ?x ?p), (at ?z ?p), (lifting ?x ?y); the delete list (lifting ?x ?y); and the add list (at ?y ?p), (in ?y ?z), (available ?x), (clear ?y).

ARMS proceeds in two phases. In phase one, it finds frequent-action sets from plans sharing a common parameter set. In addition, with the help of the initial and goal states, ARMS finds some frequent relation-action pairs and uses them to make an initial guess on the preconditions, add lists, and delete lists of actions in this subset. It uses these action subsets and pairs to obtain a constraint set that must hold for the plans to be correct.

In phase two, ARMS takes the frequent item sets as input and transforms them into constraints in the form of a weighted Max-Sat representation.[7] It solves the constraints using a weighted Max-Sat solver and produces action models as a result. The process iterates until it models all actions.

Future work

While ARMS action models are deterministic, in the future we plan to extend the framework to learning probabilistic action models that can handle uncertainty. We can add other constraints to allow partial observations between actions and prove the system's formal properties. We've tested ARMS successfully on all the Stanford Research Institute Problem Solver's planning domains from a recent AI Planning Competition based on training action sequences.[5] In related work,[8] we consider ways to generate actions to cost-effectively acquire missing data.

Acknowledgments

I thank the Hong Kong Research Grants Council (grant 621606) for its support.

References

1. J.R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann, 1993.
2. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proc. 20th Int'l Conf. Very Large Data Bases* (VLDB 94), Morgan Kaufmann, 1994, pp. 487–499.
3. Q. Yang et al., "Extracting Actionable Knowledge from Decision Trees," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 1, 2007, pp. 43–56.
4. J. Huang and C. X. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 3, 2005, pp. 299–310.
5. Q. Yang, K. Wu, and Y. Jiang, "Learning Action Models from Plan Examples Using Weighted Max-Sat," *Artificial Intelligence*, vol. 171, nos. 2–3, 2007, pp. 107–143.
6. M. Ghallab et al., "PDDL—The Planning Domain Definition Language," tech. manual, AIPS Planning Committee, 1998.
7. H. Kautz and B. Selman, "Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search," *Proc 13th Nat'l Conf. Artificial Intelligence*, AAAI, 1996, pp. 1194–1201.
8. Q. Yang et al., "Test-Cost Sensitive Classification on Data with Missing Values," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 5, 2006, pp. 626–638.

Transductive reasoning, which argues from particulars to particulars, works better in many situations because it produces a tailored local model as a result for each new input.

Actionable Data Mining in Video and Vision Systems

David Bell, *Queen's University Belfast*

Techniques for actionable data mining are often application specific: "Horses for courses" is a commonly used rule. However, even application-specific techniques can generalize to ideas that encourage both formal and pragmatic advances. Here, I describe two practical scenarios based on behavioral data mining in video and vision systems. The scenarios involve traffic surveillance and a predator's observation of its prey. I will show how both scenarios exert a strong pull on data mining technology and, at the same time, get a push from current data mining technology.

The applications' pull

Both applications are oriented toward profiling particular individuals and events from a population. In addition to the usual data mining problems of representation and performance, the applications share several characteristics: they both involve inductive and transductive reasoning, adaptive learning over time, and unstructured data.

Inductive reasoning is concerned with building a global model to capture general data patterns across a complete data space and using this model to subsequently predict output values for a particular input. However, such models are difficult to create and update, and they're often not necessary. As the 19th century philosopher John Stuart Mill noted: "The child who having burnt his fingers, avoids to thrust them again into the fire, has reasoned or inferred, though he never thought of the general maxim, Fire burns."

Transductive reasoning, which argues from particulars to particulars, works better in many situations because it produces a tailored local model as a result for each new input—for example, an individual's behavior traces from sampled database records. Transduction is often appropriate in situations that focus on behaviors that unfold over time. A predator animal hunting its prey at Queen's University Belfast is one example we've used in our studies. However, inductive reasoning is also useful—for example, in studying species behavior.

For adaptive learning applications, video and vision systems need techniques for mining time-series data. There are several traditional approaches. Some look at similarities between two sequences, some seek optimal algorithms for classifying sequences into similar subsequences, some search for repeating cycles, and others try to extract explicit rules over the time series. QUB's Knowledge and Data Engineering group has developed some ideas and techniques for addressing this specific problem environment. They involve capturing coarsened behavior components from pixel values and classifying sequences as such components.

Video and vision systems also pose extreme requirements on data mining techniques relative to unstructured data. Some estimates say that 85 percent of the data we use is unstructured—for example, in emails, workbooks, consumer comments, and other textual documents. Video data is often much more difficult to assemble for pattern recognition than table data. For example, consider a video of a predator animal trying to gain an advantage on potential prey by watching it

Table 2. Illustrative tuple traces for five time slots relative to an attack decision.*

Tuple	T0	T1	T2	T3	T4	Attack
1	Fwd S	Pause	Fwd S	Fwd Q	Pause	No
2	Fwd S	Pause	Fwd S	Pause	Bwd Q	No
3	Bwd Q	Pause	Fwd S	Fwd Q	Fwd S	Yes
4	Pause	Fwd Q	Fwd S	Fwd Q	Fwd S	Yes
5	Surp	Bwd Q	Pause	Fwd S	Fwd Q	Yes
6	Surp	Bwd Q	Pause	Surp	Bwd Q	No

*Fwd S/Q: forward movement, slowly or quickly; Bwd: backward movement; Surp: Surprised

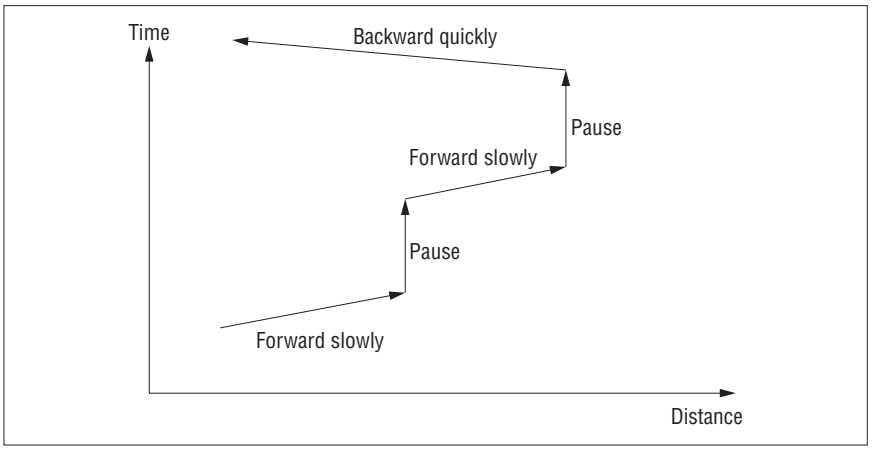


Figure 1. An example of five time slots/molecules from an episode (tuple 2 in table 2).

adapt to its environment. What will its next move be? What is its idiosyncratic behavior pattern? The irrelevant detail and noise in the visual data is huge. Furthermore, it varies between individuals and even episodes. Paradoxically, though, video can have other sorts of structure, as I will show in the next section.

The technology's push

Traffic and security surveillance systems are potentially important instances of actionable, transductive mining. They use video clips and other image data that contain multiple, potentially interesting behavioral and other sequences that vary with time. Our work in this domain includes collaboration with QUB's Speech, Image, and Vision Systems group, which has developed—among other things—a sensor-based system that automatically detects and tracks moving objects.[1] Our work is concerned with capturing such systems' output for higher-level analysis using data mining and, where appropriate, other AI techniques for dynamic scene analysis. The applications envisaged include predicting traffic incidents by learning activity patterns from stored trajectories and spotting abnormal behavior.

Suppose a vehicle performs a U-turn near a roadblock. To initiate tracking this vehicle, the system must learn normal behavior and then identify unusual patterns. The idea is to capture coarsened patterns of behavior fragments—for example, speed behavior. We can obtain tuples tracing the patterns in various scenarios. We can have an expert describe abnormal behavior classes or, alternatively, the system can identify them automatically. The system then looks for matches with the patterns being sought. Such traffic surveillance systems have close similarities to predator-prey scenarios.

This outline is greatly simplified. Individual actors have idiosyncratic behavior patterns, and existing representation schemes aren't always rich enough to capture movements and other behaviors in enough detail to help in inductive or transductive reasoning. Nevertheless, patterns often carry across several episodes of a behavior or activity, so video and vision systems offer an analysis medium for classifying the patterns to gain more general insights into the behavior or activity.

To demonstrate our techniques and provide insights into animal behavior, we've implemented a system for capturing time-varying behavioral pattern sequences of robots, which serve as animal subjects and have the advantage of being fully under the experimenter's control. The system can use either video clip observations or a vision system's output. The system inputs can be either real episodes of a robot predator and a robot prey or synthetic episodes—for example, simulations of a prey robot emulating the vision system of Sony's Aibo dog robot. The detailed time slices of action are coarsened to provide gross, molecular units of behavior. For both individuals and populations, or for a particular episode, we can represent combinations of these behavior units in table form, to be mined using various techniques. For example, the output in table 2 shows molecular behavior units such as Fwd S or Fwd Q (forward slowly/quickly) and more exotic coarsened units such as Surprised (Surp). Figure 1 illustrates the behavior of tuple 2. The table shows the outcome of this behavior—in this case, not to attack.

The table structure is more complex if the sequence includes a second actor, of course. The similarities between this scenario and other monitoring and adaptation applications are significant. From the time series of behavior chunks, a predator robot might see, for example, how a prey robot learns. If it observes more examples of the prey's behavior, as in table 2's other tuples, it can learn about behavior patterns. If the predator observes other members of the prey's "species," it might generalize its knowledge to the species.

Other applications

We study a variety of other data mining techniques and applications that organically interact with these techniques. For example, we work on more structured data sets for studies with belief networks, rough sets, associative-mining methods, and a variety of text-mining algorithms for different applications. The results help in debugging multiplexer networks in telecommunications applications, facilitating nuclear safety and applications such as content management and technology watch, or simply turning inert, unstructured text into actionable knowledge.

References

1. F. Campbell-West and P. Miller, "Evaluation of a Robust Least Squares Motion Detection

Algorithm for Projective Sensor Motions Parallel to a Plane," *Proc. SPIE Opto-Ireland Image and Vision Conf.*, Soc. Photo-Optical Instrumentation Engineers, 2005, pp. 225–236.

Visual Mining of DNA Sequences

Michail Vlachos and Philip S. Yu, *IBM T.J. Watson Research Center*
Bahar Taneri, *Scripps Genome Center*
Eamonn Keogh, *University of California, Riverside*

Now that complete genome mappings are available for several species, comparative DNA analysis is a routine application, providing many insights into the conservation, regulation, and function of genes and proteins. Even though many consider the human eye to be the ultimate data mining tool, visual comparison between DNA nucleotides can be difficult for humans, given that typical DNA data sets contain thousands of nucleotides (typical gene length is 2,000 bases, and the whole human genome consists of 3 billion base pairs).

Humans are better at conceptualizing and comparing shapes than text. Here, we present a method for visualizing DNA nucleotide sequences into numerical trajectories in space. The trajectories capture each sequence's nucleotide content, allowing for quick, easy comparison of different DNA sequences. Additionally, the resulting trajectories are organized in 2D space in a way that preserves their relative distances as accurately as possible. We illustrate this approach in one of many biological applications, visually determining the molecular phylogenetic relationship of species.

Comparative mitochondrial DNA (mtDNA) analyses have proved useful in establishing phylogeny among a wide range of species.[1] mtDNA is passed on only from the mother during sexual reproduction, making the mitochondria clones. This means that mtDNA changes are minor from generation to generation. Unlike nuclear DNA, which changes by 50 percent each generation, mtDNA mutations are rare—that is, mtDNA has a long *memory*. In this study, we use mtDNA to identify the evolutionary distances among species.

From DNA sequences to trajectories

DNA sequences are combinatorial sequences of four nucleotides; adenine, cytosine, thymine, and guanine, which are denoted by A, C, T, and G. By scanning each symbol in a DNA sequence, we can construct a trajectory that starts from a fixed point on the Cartesian coordinate system and moves in four directions according to the given nucleotide.

To quantify the similarity between the resulting trajectories, we utilize a *warping distance*,[2] which allows for elastic matching between the DNA trajectories, supporting local compressions and decompressions. The warping distance has an additional desirable property: it supports matchings between trajectories of different lengths. Figure 2 illustrates the flexible matching between trajectories that warping distance achieves. Figure 2a shows the mapping between the points in the resulting human and chimpanzee mtDNA trajectories, and figure 2b shows the mapping between the human and cat mtDNA trajectories.

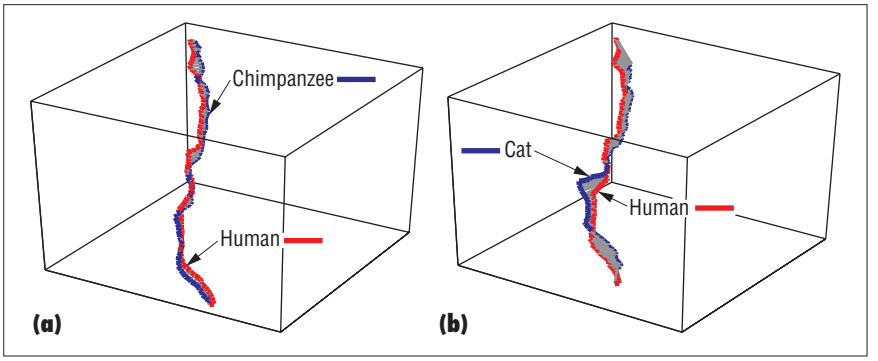


Figure 2. Matching DNA trajectories using warping distance: (a) human versus chimpanzee and (b) human versus cat.

Spanning-tree visualization

Now we need a fast visualization technique that also accurately highlights the pairwise relationships between objects in two dimensions.[3] We can easily retain the distances between any three objects A, B, and C in two dimensions by placing the objects on the vertices of a triangle constructed as follows: if $D(A, B)$ is the distance between A and B, we can map the third point at the intersection of circles transcribed with centers A and B and radii of, respectively, $D(A, C)$—that is, the distance between points A and C—and $D(B, C)$. Given the triangle inequality, the circles either intersect at two positions or are tangent. Any position on the circles' intersection will retain the original distance. So, when mapping n objects, we can retain three distances for the first three objects and two distances (with respect to two reference points) for the remaining $n - 3$ objects, preserving a total of $3 + 2(n - 3)$ distances. Using the minimum-spanning-tree (MST) with this triangulation method, we can preserve two distances per object on the 2D space: the distance to each object's nearest neighbor and the distance with respect to a pivot point.

However, this mapping is valid only for metric distances because only then are the transcribed circles guaranteed to intersect with respect to the two reference points. The warping distance we use in assessing DNA trajectory similarities is a nonmetric distance, which means the circles might not intersect. Hence, we must extend the technique to properly identify the third point's position so that it's as close as possible to the circumference of the (possibly) nonintersecting reference circles.

Once we've incorporated these additions on the mapping method, we have a powerful visualization technique for nonmetric distances. It preserves as well as possible not only nearest-neighbor distances (local structure) but also global distances with respect to a single reference (pivot) point. The latter supports global data viewing using the point as a pivot. Finally, the method is computationally spartan because we can construct the MST in $O(V \log E)$ time for a graph of V vertices and E edges.

Applications to evolutionary biology

Several examples demonstrate the usefulness of the proposed trajectory transformation and 2D mapping technique. First, we use mtDNA from *Homo sapiens* and seven related species to construct the spanning-tree visualization. Figure 3 depicts the results,

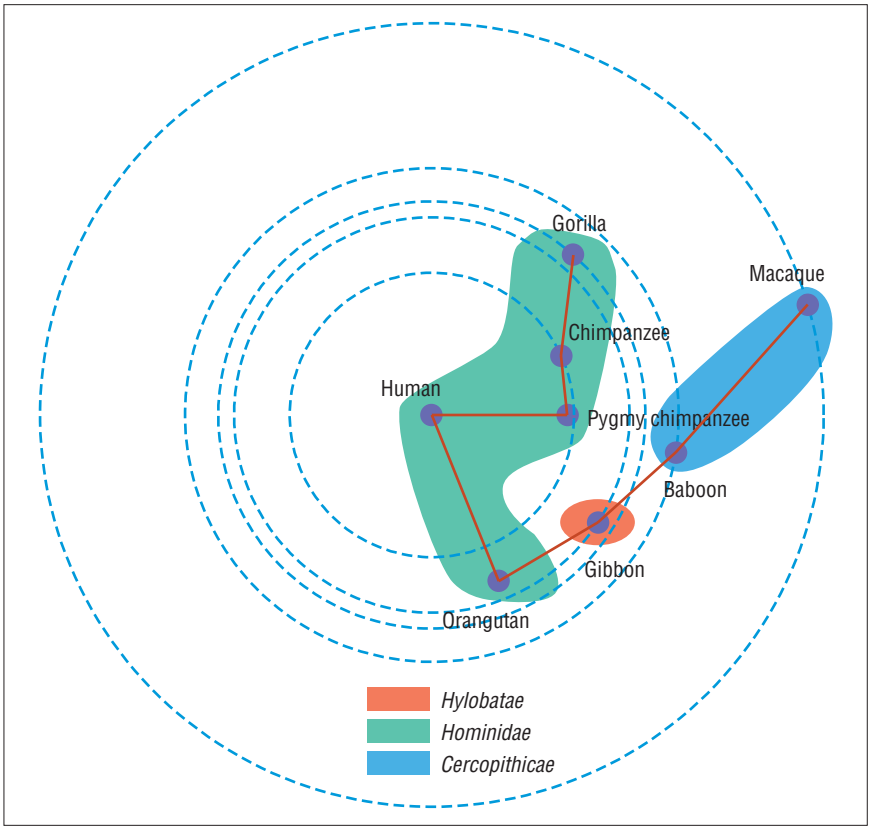


Figure 3. A visualization of humans and related species using the spanning-tree visualization method.

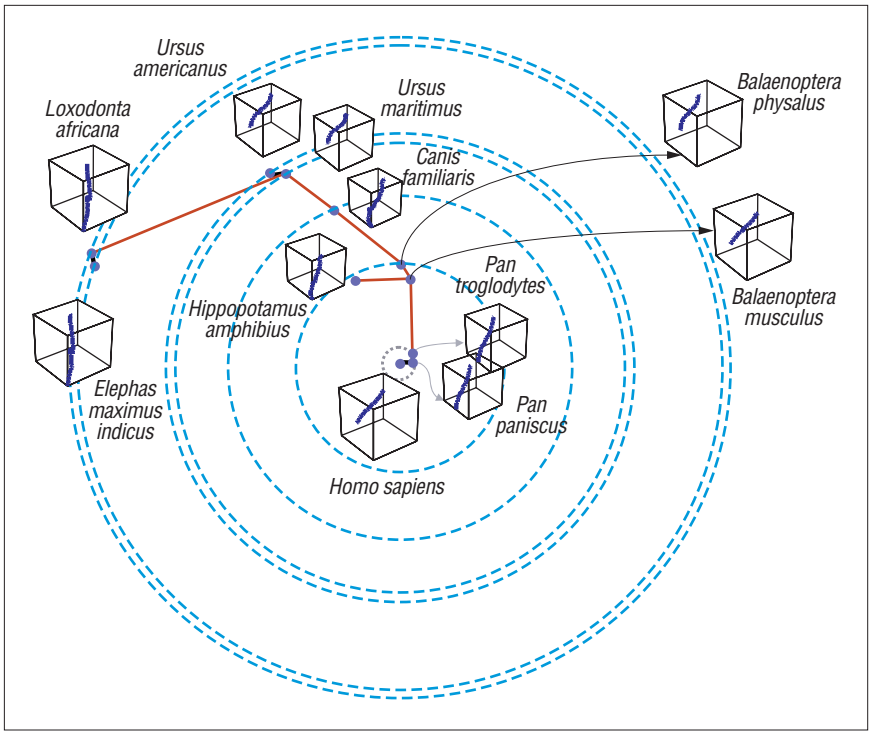


Figure 4. An evolutionary visualization of various species with respect to humans.

which agree with current evolutionary views. The mapping not only is accurate with regard to the evolutionary distance of the species but also preserves the clustering between the original groups that the various species belong to. Specifically, human, pygmy chimpanzee, chimpanzee, and orangutan belong to the *hominidae* group, gibbon to the *hylobatae* group, and baboon and macaque to the *cercopithicae* group. Our results corroborate earlier findings on evolutionary distances of *Homo sapiens* to other mammalian species.[2] Human and orangutan divergence took place approximately 11 million years ago, whereas gibbon and human divergence occurred approximately 15 million years ago.[5] According to the same source, gorilla divergence occurred about 6.5 million years ago and chimpanzee divergence took place about 5.5 million years ago.

Figure 4 illustrates our second example, which involves a larger mammalian data set and again takes the human as the referential point. On this plot, we use the formal species names and overlay the DNA trajectory of the respective mtDNA sequence. At first glance, the closeness of the hippopotamus with the whales might seem like a misplacement. Intuitively, a hippopotamus has greater affinity with an elephant. In fact, however, the hippopotami are more closely related to whales than to any other mammals. Whales and hippopotami diverged 54 million years ago, whereas the whale-hippopotamus group parted from the elephants about 105 million years ago. The group that includes hippopotami and whales-dolphins, but excludes all other mammals in figure 4, is *Cetartiodactyla*.[6] In general, the figure illustrates the strong visualization capacity of the spanning-tree technique, particularly in unveiling the similarities and connections between the different species.

Future work

Here, we've used only small data instances to present this novel DNA representation and visualization technique. However, we expect to find many applications in mining large sequence collections, especially in conjunction with advanced compression and indexing techniques. We intend to apply our techniques to additional biomedical applications, including screening and diagnostic techniques for cancer data, where they could distinguish cancer transcripts from unaffected ones and identify different cancer stages.

References

1. A.K. Royyuru et al., "Inferring Common Origins from mtDNA," *Research in Computational Molecular Biology*, LNCS 3909, Springer, 2006, pp. 246–247.
2. M. Vlachos et al., "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures," *Proc. 9th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining* (KDD 03), ACM Press, 2003, pp. 216–225.
3. R. Lee, J. Slagle, and H. Blum, "A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space," *IEEE Trans. Computers*, vol. C-26, no. 3, 1977, pp. 288–292.
4. Z. Cheng et al., "A Genome-Wide Comparison of Recent Chimpanzee and Human Segmental Duplications," *Nature*, vol. 437, no. 7055, 2005, pp. 88–93.
5. R. Stauffer et al., "Human and Ape Molecular Clocks and Constraints on Paleontological Hypotheses," *J. Heredity*, vol. 92, no. 6, 2001, pp. 469–474.
6. B.M. Ursing and U. Arnason, "Analyses of Mitochondrial Genomes Strongly Support a Hippopotamus-Whale Clade," *Proc. Royal Soc. London*, series B, vol. 265, 1998, pp. 2251–2255.

Actionable Knowledge Discovery: A Brain Informatics Perspective

Ning Zhong, *Maebashi Institute of Technology*

Brain informatics pursues a holistic understanding of human intelligence through a systematic approach to brain research. BI regards the brain as an information-processing system and emphasizes cognitive experiments to understand its mechanisms for analyzing and managing data. *Multiaspect data analysis* is an important BI methodology because the brain is too complex for a single data mining algorithm to analyze all the available cognitive experimental data. MDA supports an agent-based approach that has two main benefits for addressing the complexity and diversity of human brain data and applications:

- its agents can cooperate in a multiphase process and support multilevel conceptual abstraction and learning, and
- its agent-based approach supports task decomposition for distributing data mining subtasks over the Grid.

MDA requires a Web-based BI portal that can support a multiphase mining process based on a conceptual data model of the human brain. Generally speaking, MDA can mine several kinds of rules and hypotheses from different data sources, but brain researchers can't use MDA results directly. Instead, an explanation-based reasoning process must combine and refine them into more general results to form *actionable knowledge*. From an application's viewpoint, the BI provides the knowledge-flow management for distributed Web inference engines that employ actionable knowledge and related data sources to implement knowledge services.[1]

A BI portal for MDA

Building a BI portal requires the development of a multilayer, data mining grid system to support MDA. At the Maebashi Institute of Technology's Department of Life Science and Informatics, we've been developing a systematic approach to support this goal. We use powerful instruments, such as functional magnetic resonance imaging (fMRI) and electroencephalography (EEG), to measure, collect, model, transform, manage, and mine human brain data obtained from various cognitive experiments.[1]

> Multiaspect data analysis is important because the brain is too complex for a single data mining algorithm to analyze available cognitive experimental data.

fMRI provides images of functional brain activity that show dynamic patterns throughout the brain for a given task; fMRI image resolution is excellent, but the process is relatively slow. EEG provides information about the electrical fluctuations between neurons that also characterize brain activity, and it measures brain activity in near real time. Discovering new knowledge and models of human information-processing activities requires not only individual data obtained from a single measuring method but also multiple data sources measuring methods.

Our work focuses on human information processing activities on two levels:

- spatiotemporal features and flow based on functional relationships between activated brain areas for each given task, and
- neural structures and neurobiological processes related to the activated areas.

More specifically, at the current stage, we want to understand how neurological processes support a cognitive process. We're investigating how a specific part of the brain operates in a specific time, how the operations change over time, and how the activated areas work cooperatively to implement a whole information-processing system. We're also looking at individual differences in performance.

BI's future will be affected by the ability to mine fMRI and EEG brain activations on a large scale. Key issues are how to design the psychological and physiological experiments for obtaining data from the brain's information-processing mechanisms and how to analyze and manage such data from multiple aspects for discovering new human information-processing models. Researchers are also studying how to use data mining and statistical learning techniques to automate fMRI image analysis and understanding.[2,3]

Web intelligence[4] and Grid computing[5] provide ideal infrastructures, platforms, and technologies for building a BI portal to deal with MDA's huge, distributed data sources. They can support a data mining grid composed of many agent components. Each agent can do some simple task, but when the Grid integrates all these agents, they can carry out more complex BI tasks.

Using data mining agents entails both preprocessing and postprocessing steps. Knowledge discovery generally involves background knowledge from experts, such as brain scientists, about a domain, such as cognitive neuroscience, to guide a spiral, multiphase process to find interesting and novel rules and features hidden in data. On the basis of such data, BI generates hypotheses that it deploys on the Grid for use by various knowledge-based inference and reasoning technologies.[1] From a top-down perspective, the knowledge level is also the application level. Both the mining and data levels support brain scientists in their work and the portal in updating its resources. From a

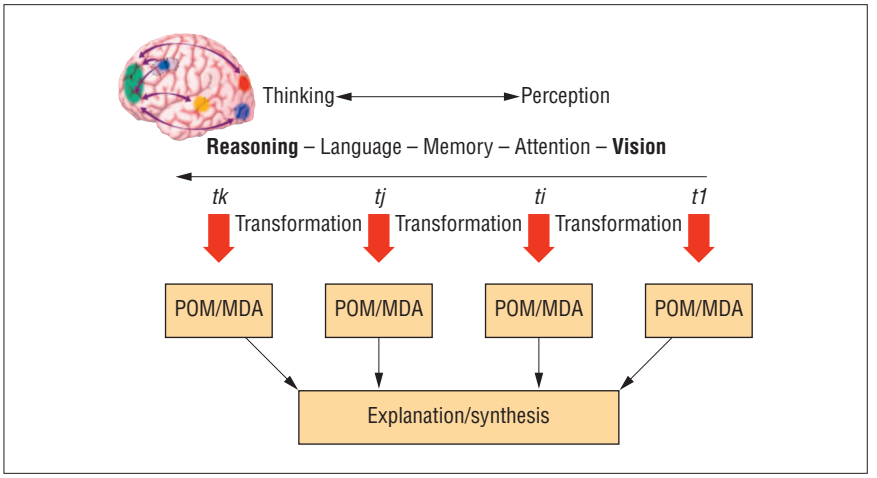


Figure 5. Investigating the spatiotemporal features and flow of a human information processing system.

bottom-up perspective, the data level supplies the data services for the mining level, and the mining level produces new rules and hypotheses for the knowledge level to generate actionable knowledge.

Peculiarity-oriented mining

fMRI and EEG data are peculiar with respect to a specific state or the related part of a stimulus. *Peculiarity-oriented mining* is a proposed knowledge discovery methodology that automates fMRI and EEG data analysis and understanding. POM doesn't use conventional fMRI image processing or EEG frequency analysis, and it doesn't require human-expert-centric visualization.

Figure 5 illustrates the methodology applied to interpreting the spatiotemporal features and flow of a human information processing system. In the cognitive process from perception (in this case, a cognitive task stimulated by vision) to thinking (reasoning), the system collects data from several event-related points in time and transforms them into various forms for POM-centric MDA. Finally, the system explains the results of the separate analyses and synthesizes them into a whole flow.

The proposed POM/MDA methodology shifts the focus of cognitive science from a single type of experimental data analysis toward a deep, holistic understanding of human information-processing principles, models, and mechanisms.

References

1. N. Zhong et al., "Building a Data Mining Grid for Multiple Human Brain Data Analysis," *Computational Intelligence*, vol. 21, no. 2, 2005, pp. 177–196.
2. F.T. Sommer and A. Wichert, eds., *Exploratory Analysis and Data Modeling in Functional Neuroimaging*, MIT Press, 2003.
3. N. Zhong et al., "Peculiarity Oriented fMRI Brain Data Analysis for Studying Human Multi-Perception Mechanism," *Cognitive Systems Research*, vol. 5, no. 3, 2004, pp. 241–256.
4. J. Liu et al., "The Wisdom Web: New Challenges for Web Intelligence (WI)," *J. Intelligent Information Systems*, vol. 20, no. 1, 2003, pp. 5–9.
5. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.

Privacy-Preserving Data Mining: Past, Present, and Future

Mafruz Zaman Ashrafi, *Institute for Infocomm Research*
David Taniar, *Monash University*

Privacy issues grow in importance as data mining increases its power to discover knowledge and trends from ever-larger data stores. The reason is simple and straightforward: once information is released, it's difficult to control and so impossible to protect from misuse.

Aggregating data from various sources increases the risk of privacy breaches, but many applications require data sharing to ensure their resulting model accuracy. Unless they aggregate data from different sources, the mining models they generate can contain many false positives and useless patterns. More importantly, the models might exclude knowledge that's critical in decision making. For example, homeland-defense applications profile individuals by combining privacy-sensitive data sources from various domains. Without combining these sources, the data mining models might tag an innocent person as a criminal or vice versa.

Because the consequences of inaccuracy are so serious, privacy has emerged as a top data mining research priority. Mitigating privacy risk in distributed data mining models involves two broad issues: privacy-preserving data aggregation and data mining model accuracy.

Privacy in frequent-item-set mining

The distributed data mining process discloses each participating source site's item-set frequency. Privacy-preserving methods in this distributed process mainly rely on the way each site shares its local support of its item sets without exposing the exact frequency. Three well-known approaches to achieve this goal are *randomization*, *secure multiparty computation* (SMC), and *cryptography*.

Randomization approaches are based on randomized data sets from various sites.[1] Participating sites preserve privacy by discarding some data set items and inserting new ones. They send the results to a centralized third-party site and ensure accuracy by including statistical estimates of the original item frequency and randomization variance. Applications can use these estimates, together with the Apriori algorithm,[2] to mine the nonrandomized transaction frequencies while looking only at the randomized frequencies.

Randomization techniques in frequency-item-set mining are either *transaction invariant* or *item invariant*. The transaction-invariant technique will breach privacy if the given data set's individual transaction size is large. For example, a transaction size $|t| \succ 10$ is doomed to fail at protecting privacy. On the other hand, the item-invariant technique includes all items in a perturbed transaction $|t'| = R(t)$. This technique assumes the probability of these items to be the same, and it completely ignores the correlation between them. The resulting frequent-item sets might be unable to accurately reflect many of the

original items. The item-invariant technique also incurs additional computational cost. Furthermore, if each participating site has a large data set, distorting the data set can involve huge computation.

SMC-based techniques discover an item set's global frequency without involving a trusted third party.[3,4] Unlike randomization approaches, SMC approaches neither perturb the original data set nor send it to a centralized site. Instead, they perform a secure computation in two cycles.

To discover a global frequency, in the first cycle, known as *obfuscation*, every participating site generates its local frequency x and obfuscates it by performing a function $f_i(x \oplus r_i)$, where r is a random noise. The site then sends the obfuscated frequency to an adjacent site. The adjacent site j obfuscates its local frequency in the same manner $f_j(x \oplus r_j)$, combines the obfuscated frequencies, and sends the result to the next adjacent site. The process continues until no participating site remains. At this point, the obfuscated frequency includes each item set's local frequency and noise. In the second cycle, *de-obfuscation*, each site repeats this process with one exception: instead of adding noise, it removes noise from the obfuscated frequency. At the end of this round, each site can discover the exact global frequency of a given item set.

Although generating global frequency of any item set is quite straightforward with SMC, privacy is still vulnerable if the participating sites collude with each other. For example, sites i and $i + 2$ in the chain can collude to find the exact support of site $i + 1$. SMC also incurs high communication costs because each site must communicate with others many times to find each item set's exact global frequency.

SMC's communication costs increase exponentially as the number of participating sites increases. In fact, these approaches aren't feasible when the number of participants is large—for example, in an online survey. Crypto-based systems can overcome this limitation by using public-key cryptography to generate global frequency.[5] Similar to randomization, crypto systems use a centralized site to aggregate all participating sites' frequencies without losing accuracy as the cost of privacy. However, if each data set has many local frequent-item sets, this approach not only incurs high computation costs but also increases communication costs.

Future trends

A privacy-preserving mining model isn't easy to achieve, and it's perhaps impossible to achieve privacy without trade-offs. A data mining application determines the cost it must pay to reach the required privacy level. Figure 6 summarizes these trade-offs for the approaches we've described.

Apart from these trade-offs, other privacy questions also need serious and immediate attention. For example, does the data mining pattern itself breach privacy? Can the data mining model's privacy be exposed by associating the model with public data sources? Such problems might not be visible immediately, but the threats they pose are real.

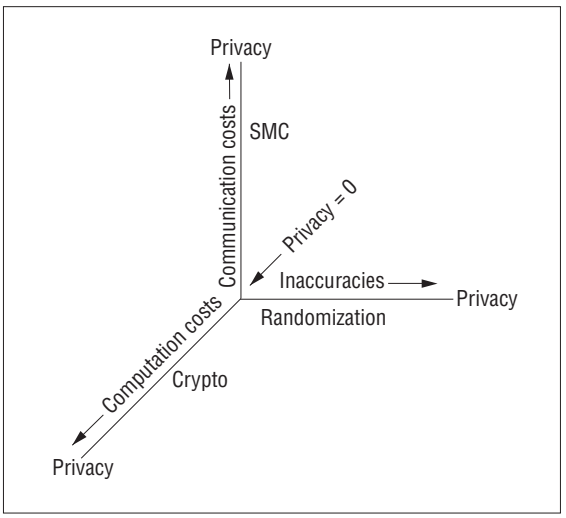


Figure 6. Privacy-preserving frequent-item-set mining approaches and their trade-offs.

References

1. A.V. Evfimievski et al., "Privacy Preserving Mining of Association Rules," *Information Systems*, vol. 29, no. 4, 2004, pp. 343–364.
2. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Database," *Proc. 20th Int'l Conf. Very Large Databases*, IEEE CS Press, 1994, pp. 407–419.
3. M. Kantarcioglu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 9, 2004, pp. 1026–1037.
4. M.Z. Ashrafi, D. Taniar and K. Smith, "Reducing Communication Cost in a Privacy Preserving Distributed Association Rule Mining," *Proc. Database Systems for Advanced Applications*, LNCS 2973, Springer, 2004, pp. 381–392.
5. Z. Yang, S. Zhong and R.N. Wright, "Privacy-Preserving Classification of Customer Data without Loss of Accuracy," *Proc. 2005 SIAM Int'l Conf. Data Mining* (SDM 05), Soc. Industrial and Applied Math., 2005; www.siam.org/meetings/sdm05/downloads.htm.

Toward Knowledge-Driven Data Mining

Eugene Dubossarsky, *Ernst & Young*
Warwick Graco, *Australian Taxation Office*

Data mining faces many challenges, but the primary one is to move from a method-driven approach to a process driven by domain problems and knowledge. Here, we describe four key aspects of successfully meeting this challenge: moving to intelligent algorithms that utilize expert knowledge, converting dumb data points to smart ones, combining business knowledge with technical knowledge to optimize mining and modeling results, and using discovery and detection results to improve intelligence analysis.

Intelligent algorithms

Many data mining algorithms lack smarts and are biased in their capabilities. For example, K-means clustering algorithms are biased toward recovering clusters that have hyperspherical shapes.[1] They don't perform well in recovering other naturally occurring forms, such as banana-shaped clusters. We need intelligent algorithms that can identify clusters in data regardless of shapes, sizes, and spatial orientations.

Such algorithms require two key components. First is a range of tools and techniques for working out the optimum fit between data and classification and prediction models and for discovering data configurations and correlations such as associations, clusters, and classes. Second, intelligent algorithms require expert knowledge. This knowledge can take the form of heuristics and routines that manage data nuances and tailor the data mining to specific tasks. One solution is supervised clustering—for

Table 3. WIKID hierarchy.

Concept	Description
Wisdom	Knowledge rightly applied to solve difficult problems and issues
Intelligence	Evaluated knowledge from which relevant insights and understanding have been extracted
Knowledge	Deep and detailed information on a topic
Information	Data on issues
Data	Basic facts and figures

example, user-driven selection of cluster seeds and distance metrics.[2] Clustering techniques can also employ active learning techniques to obtain expert feedback for optimal results.[3] Other data mining techniques should also incorporate expert knowledge.

Intelligent data

Current data mining techniques are applied to dumb data points. Intelligent data points could aid both data mining and data modeling. Each data point should have metadata to explain when it was created, who created it, and what it represents. In addition, metaknowledge in the form of expert explanation and interpretation could clarify each data point's significance.

Dumb data points don't help an algorithm discriminate whether the data is relevant to solving specific mining and modeling problems. Efficient algorithms need to distinguish noise from signal—a distinction that depends on the problem context. Data that's noise in one context might be signal in another. For example, prevailing weather conditions might not be relevant to explaining certain crime patterns, but they're likely to be important in explaining patterns of disease.

We need algorithms that can interrogate the metadata and metaknowledge attached to data points in much the same way the Semantic Web proposes to do for Web content.[4] The Semantic Web expresses Web content in a form that software agents can read. This allows them to find, share, and integrate information more easily. In a similar way, data mining algorithms could use metadata and metaknowledge to eliminate data points that have little bearing on the problem being solved. These same sources would also help users understand data mining results.

Business knowledge

Data mining must marry business knowledge and technical knowledge. Data miners often lack business domain knowledge when they undertake mining and modeling tasks. This essential knowledge comes from business experts, who can help throughout the data mining life cycle. These experts must guide the exploration process, acting as navigators while data miners do the driving.

Method-driven data mining can produce many pages of results with little or no significance. Knowledge-driven data mining needs business experts to identify the important results and interpret them in the form of metaknowledge. Metaknowledge puts results in perspective and helps users understand their significance for purposes such as increasing productivity, lowering costs, and improving outcomes. For example, Tatiana Semenova highlighted how to use metaknowledge derived from interpreting data mining results to develop best-practice medical guidelines to improve patient outcomes.[5]

Another perspective on this issue is mining the tacit knowledge of experienced people to predict outcomes. Research has shown that such knowledge can produce superior results in prediction markets ranging from political forecasting to commercial sales forecasts.[6]

Intelligence

Confusion abounds around what "intelligence" means. To clarify it here, we use the WIKID (pronounced "Why-kid") hierarchy that's emerged from the knowledge management community (see table 3). In this hierarchy, intelligence refers to information that's been analyzed and evaluated—part of a process of making informed assessments of what will happen, when, and how. It relies on the capacity to acquire and apply knowledge. Analysts who perform this function are responsible for marshalling the facts, drawing deductions from what is known, and providing options to decision makers. Intelligence is the lifeblood of organizations and helps to determine where resources should be invested to exploit opportunities and to counter threats.

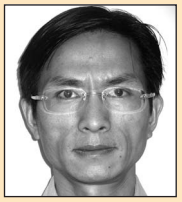
Intelligence and data mining have a symbiotic relationship. Intelligence helps to resolve where to focus data mining—for example, to detect fraudulent insurance claims. Equally, data mining research and practice discover new knowledge to evaluate.

Conclusion

Knowledge is the fuel that drives data mining. The work we've described here can move the field toward integrating knowledge fully into data mining practice and so enhance the intelligence of its results. ■

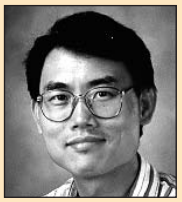
References

1. M. Mu-Chun Su and C. Chou, "A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, 2001, pp. 674–680.
2. S. Basu, A. Banerjee, and R.J. Mooney, "Semi-supervised Clustering by Seeding," *Proc. 19th Int'l Conf. Machine Learning* (ICML 02), Morgan Kaufmann, 2002, pp. 19–26.
3. S. Tong, "Active Learning: Theory and Applications," doctoral dissertation, Dept. of Computer Science, Stanford Univ., 2001.
4. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, May 2001, pp. 34–43.
5. T. Semenova, "Identification of Interesting Patterns in Large Data Health Data Bases," doctoral dissertation, Research School of Information Sciences and Eng., Australian Nat'l Univ., 2006.
6. A. Leigh and J. Wolfers, "Wisdom of the Masses," *Australian Financial Rev.*, 8 June 2007, pp. 3–4.

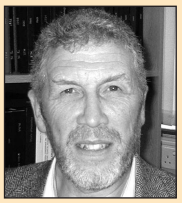


Longbing Cao is a senior lecturer of the Faculty of Information Technology at the University of Technology, Sydney. Contact him at lbcao@it.uts.edu.au.

Chengqi Zhang is a professor of the Faculty of Information Technology at the University of Technology, Sydney. Contact him at chengqi@it.uts.edu.au.



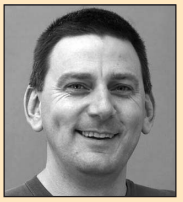
Qiang Yang is a professor at the Hong Kong University of Science and Technology. Contact him at qyang@cs.ust.hk.



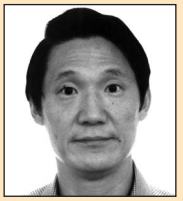
David Bell is a professor in the School of Electronics, Electrical Engineering and Computer Science at Queen's University, Belfast. Contact him at da.bell@qub.ac.uk.

Michail Vlachos is a research staff member at the IBM T.J. Watson Research Center, New York. Contact him at vlachos@us.ibm.com.

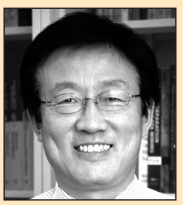
Bahar Taneri is an assistant professor of psychology at Eastern Mediterranean University, Cypress, and a member of the Scripps Genome Center at the University of California, San Diego. Contact her at bahar@genomes.ucsd.edu.



Eamonn Keogh is an associate professor of computer science at the University of California, Riverside. Contact him at eamonn@cs.ucr.edu.



Philip S. Yu is the manager of the Software Tools and Techniques group at the IBM T.J. Watson Research Center, New York. Contact him at psyu@us.ibm.com.

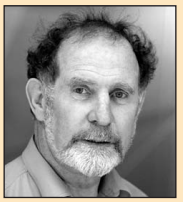


Ning Zhong is a professor in the Department of Life Science and Informatics at the Maebashi Institute of Technology, Japan. He's also an adjunct professor in the International Web Intelligence Consortium Institute at the Beijing University of Technology. Contact him at zhong@maebashi-it.ac.jp.

Mafruz Zaman Ashrafi is a research fellow at the Institute for Infocomm Research. Contact him at mashrafi@i2r.a-star.edu.sg.

David Taniar is a senior lecturer in computing at Monash University. Contact him at david.taniar@infotech.monash.edu.au.

Eugene Dubossarsky is the director of Predictive Business Intelligence at Ernst & Young in Sydney and a Senior Visiting Fellow at the University of New South Wales. Contact him at eugene@alcesronin.com.



Warwick Graco is a senior data miner in the Office of the Chief Knowledge Officer of the Australian Taxation Office. Contact him at warwick.graco@ato.gov.au.