Overcoming the Challenge of Variety: Big Data Abstraction, the Next Evolution of Data Management for AAL Communication Systems

Rui Mao, Honglong Xu, Wenbo Wu, Jianqiang Li, Yan Li, and Minhua Lu

ABSTRACT

With the extensive use of information technology in AAL communication systems, a data management model has recently embodied the 3-V characteristics of big data: volume, velocity, and variety. A lot of work has been done on volume and velocity, but not as much has been reported on variety. To handle the variety of data, universal solutions with acceptable performance are usually much more cost effective than customized solutions. To achieve universality, a basic idea is to first define a universal abstraction that covers a wide range of data types, and then build a universal system for universal abstraction. Traditional database management systems commonly use a multidimensional data type, or feature vectors, as a universal abstraction. However, many new data types in AAL systems cannot be abstracted into multidimensional space. To find a more universal data abstraction and build more universal systems, we propose the concept of big data abstraction, with metric space as a universal abstraction for AAL data types. Furthermore, to demonstrate how metric-space data abstraction works, we survey the state of the art in metric space indexing, a fundamental task in data management. Finally, open research issues are discussed.

INTRODUCTION

Today, with the extensive use of information technology, data management and analysis for ambient assisted living (AAL) communication systems has gradually come to embody the 3-V characteristics of big data [1]:

- Volume: The amount of data in regard to computation and storage is extremely large.
- Velocity: The speed of data input and output is extremely high.
- Variety: The range of data types and sources is extremely wide.

As a result, AAL data has grown beyond the capability of most available database management tools or traditional data processing applications. A revolutionary approach to big data management in AAL is in great need.

Scholars and practitioners all over the world have done considerable intensive research on big data. However, most of the effort was spent on volume and velocity, but not as much on variety.

Some common data types in AAL communication systems are listed in Table 1.

To cope with the large number of data types, there are two basic types of solutions, customized ones and universal ones. Customized solutions build a customized system for each individual type of data, while universal solutions build a single system that can support a wide range of data types. If the performance is acceptable to the applications, universal solutions are much more cost effective. As a result, most commercial database management systems (DBMSs) are universal solutions so that they can be sold to many customers to maximize the profit.

A basic question is how to develop universal solutions. Looking back on the history of DBMSs, one can discover the basic paradigms of universal solutions. That is, one first defines a universal abstraction that covers a wide range of data types, and then builds a universal system for the universal abstraction based on its properties. Since every particular data type is a special case of the universal abstraction, a solution to the universal solution works for any data type it covers.

Commonly, traditional DBMSs make use of a multidimensional data type as a universal abstraction. That is, most data types are essentially represented by one or multiple numbers (i.e., a feature vector). However, many new data types in AAL systems cannot be abstracted into multidimensional space, and a more universal abstraction is needed for AAL data.

We propose a new concept of big data abstraction using metric space as the universal abstraction for AAL data types. Informally, a metric space [2] is a set with a distance function defined on its elements, where the distance function satisfies the triangle inequality. We show that metric space is more universal than multidi-

Rui. Mao, Honglong Xu, Yan Li, and Minhua Lu (corresponding author) are with Shenzhen University.

Wenbo Wu is with the University of Georgia.

Jianqiang Li is with Beijing University of Technology.

0163-6804/15/$25.00 © 2015 IEEE

mensional space and covers a wide range of AAL data types.

Big data abstraction is in its early stage of development. To demonstrate how big data abstraction might work, we survey the state of the art in metric space indexing. Indexing, or searching, is one of the fundamental tasks of data management and analysis. A lot of work has been done on metric space indexing. We believe what has been done for indexing provides excellent hints for other data management and analysis tasks. Open issues of big data abstraction in theory and application are also discussed.

The rest of this article is organized as follows. A discussion of customized and universal solutions is presented in the following section. After that we propose big data abstraction, and survey the state of the art in metric space indexing. In the final section, open research issues are discussed.

UNIVERSALIZATION: WHY AND HOW?

In this section, we first show the necessity of universalization by comparing customized and universal solutions, then show the basic approach to achieving universality by reviewing the history of data management systems, and last discuss the current status of big data management with respect to variety.

UNIVERSALIZATION: WHY?

Facing various data types, customized solutions build one system for each data type. Since the system is tailored for a single data type, its performance can be expected to be high. However, its range of applicability is relatively narrow, and its price is thus relatively high. As a result, the performance-price ratio will be relatively low, and less profit has to be expected.

Universal solutions, on the contrary, build one system to support a wide range of data types. After fine tuning, the performance of universal systems is generally acceptable, except for some performance-critical applications. Because of its wide applicability, a universal system can be sold to many customers at relatively low prices. As a result, universal systems are more cost effective and more profitable.

Customized solutions are more suitable for performance-critical applications, while universal solutions achieve better balance between performance and price. Usually, buyers of AAL data management systems prefer universal solutions because of their low prices, given that the performance is acceptable. Likewise, providers of AAL data management systems tend to develop universal solutions to gain more customers and profit. Consequently, universal solutions are more popular than customized solutions in practice. The relationship between customized and universal solutions is similar to that between tailor-made and factory-made clothes. The next question is how to achieve universality.

UNIVERSALIZATION: HOW?

Let us look back on the history of data management systems (Fig. 1), which always show an evolutionary trend from customization to universalization.

Data category	Data type
Behavioral habit data	Sleep time, frequency of wake up, restroom time and frequency, shower time, eating time, walking speed, time in and out
Physiological information	Blood pressure, blood lipids, blood oxygen, temperature, pulse, BMI, weight, bone density, respiratory rate
Healthcare information	Gene sequence, protein sequence, medical image
Environmental data	Surveillance video, noise level, pollution density, weather conditions

Table 1. Common data types in AAL communication systems.

In the early 1960s (Fig. 1a), with the increasing application of computers in enterprise management, large businesses began to build their own enterprise information systems, where common data were numbers: employee IDs, product prices, and so on. Since these systems were only used inside businesses, many of them were built, and a lot of resources were consumed. In the 1970s, the B-tree index was designed and integrated into relational DBMSs. B-tree supports search of numeric values, whether natural numbers, integers, or real numbers. That is, 1D data served as an abstraction of natural numbers, integers, or real numbers, and B-tree worked for all these data types since they are all special cases of 1D data. Furthermore, the integration of SQL made relational DBMSs even easier to use. Thus, some businesses were attracted to relational DBMSs. As the number of customers increased, the price of relational DBMSs dropped. Consequently, businesses gradually replaced their own information systems with relational DBMSs for acceptable performance at a much lower price. This was the first evolution of data management systems from customization to universalization (Fig. 1a).

Figure 1b shows the second stage of evolution when manmade satellites were launched. To manage spatial information acquired by satellite, individual spatial data management systems were built. Spatial data are usually represented by feature vectors and matched by similarity defined by distance functions. Again, a lot of efforts were spent on building individual systems. Later, in the 1980s, multidimensional indexing such as R-tree and kD-tree were designed and integrated into relational databases. Multidimensional indexing supports similarity search of multidimensional data with Euclidean distance or alike. Furthermore, SQL was also extended to support spatial data type and similarity query. As a result, individual spatial data management systems were gradually replaced by spatial DBMSs. This was the second evolution of tje data management system from customization to universalization (Fig. 1b).

Studying the above two evolutions, one can summarize the basic approach to achieve universality into three steps:

1. Find a universal data types that cover various data types.
2. Find a universal distance function that covers various distance functions.

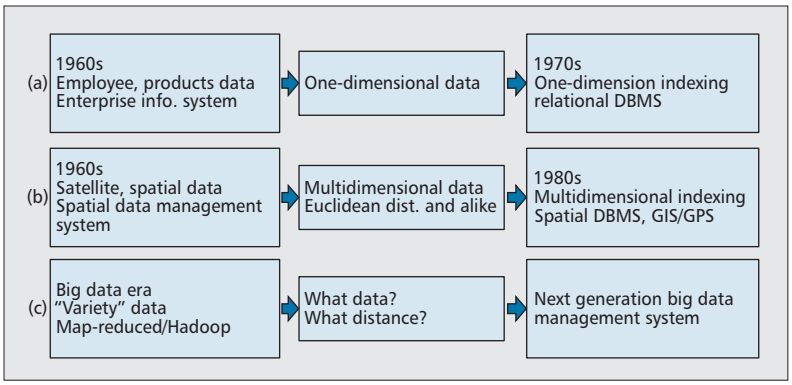

Figure 1. History of data management systems.

3. Build a system based on the properties of the universal data type and the universal distance function.

Discussion

In the second evolution, the multidimensional data type serves as the abstraction for spatial data types. A data type must satisfy two conditions to be covered by this abstraction:

- Data must be in feature vector form.
- Similarity of data must be defined by Euclidean distance or the like.

However, in the big data era, many data types and corresponding distance functions do not satisfy the above two conditions, such as text with edit distance, protein sequence with global alignment, or MMR images with Hausdorff distance. A new abstraction that can cover more AAL data types is in great need.

Map-Reduce is a very popular programming model to tackle big data applications these days. However, one should keep in mind at least two issues about map-reduce.

Who Developed It? — Map-Reduce was originally developed by Google, who possesses a great amount of both intelligence and resource to develop it. Building a big data system under the umbrella of the Map-Reduce model requires great programming skills and huge resource investment. Therefore, it is not for end users.

What Is It For? — Map-Reduce was originally developed to scan logs. After years of development, its functionality is still very limited. It is not an accurate claim that Map-Reduce has outperformed traditional DBMSs, but it is better only in limited application environments.

Therefore, we can conclude that the current status of AAL big data management is very similar to the early stage of building individual systems of the former two evolutions, and a new abstraction for big data is necessary to carry on the third evolution of data management systems from customization to universalization.

Big Data Abstraction

We propose the use of metric space as a universal abstraction for AAL data types, and to build a universal big data management and analysis system based only on the properties of metric space.

Metric Space

Informally, metric space is a set with a distance function, satisfying the triangle inequality, defined by its elements.

Definition — A metric space [2] is a pair (S, d), where S is a nonempty set and d is a real-valued distance function with the following properties:

For all $x, y \in S$, $d(x, y) \geq 0$ and $d(x, y) = 0$ iff $x = y$. (Positivity)

For all $x, y \in S$, $d(x, y) = d(y, x)$. (Symmetry)

For all $x, y, z \in S$, $d(x, y) + d(y, z) \geq d(x, z)$. (Triangle inequality)

Metric space requires only a metric distance function. An interpretation of the data in a coordinate system is not necessary. Two immediate advantages of using metric space as a universal abstraction are:

- Metric space is more universal than multidimensional space. Since Euclidean distance satisfies positivity, symmetry, and triangle inequality, multidimensional data with Euclidean distance form a special case of metric space. Some data types that cannot be abstracted into multidimensional space can be abstracted into metric space.
- A universal programming model can be built on metric space. To perform big data management and analysis, users only need to define their own data type and associated metric distance function, which can be plugged into the universal model as a black box.

Metric Space's Range of Application

Common AAL data types that can be abstracted into metric space are listed in Table 2. Except for examples 1 and 2, the examples cannot be directly abstracted into multidimensional space. Until now, only customized solutions have been developed for them.

For data types that cannot be directly abstracted into metric space, there are some alleviations. First, there are some mathematical approaches to convert non-metric distance functions to metric ones. Second, there are universal approaches for distance functions satisfying only some of the metric properties (e.g., semi-metric and pseudo-metric). Third, as long as some kind of inference can come from the distance function, universal approaches can be developed. An example is the protein identification problem with mass spectra. For three data objects x, y and z, an upper bound of $d(x, y)$ can be determined given $d(x, z)$ and $d(y, z)$, and a metric space index was adapted to support similarity queries of mass spectra [3].

The great universality of metric space is also one of the disadvantages of metric space abstraction. Domain-specific information is discarded. The triangle inequality of the distance function is the only property that can be leveraged. The key point is to recognize the pattern encapsulated by the distance function.

Metric-space-based big data abstraction is in its early stage of development. Among the fun-

damental tasks of metric space big data management and analysis, search is the only one that has received intensive research. To show the basic idea of how data management and analysis can be done in metric space, in the next section, we survey the state of the art in metric space indexing that supports similarity queries.

Metric Space Indexing: A Universal Indexing for Similarity Queries

Given a database of data objects, a distance function as the similarity measurement, and a query object, a similarity query finds all data objects that are similar, determined by the distance function, to the query object.

Figure 2 illustrates the basic idea of how triangle inequality can be leveraged to answer similarity queries in metric space. Assume an image database consists of three cartoons of Mickey, Minnie, and Pluto, respectively [4]. Another cartoon of Mickey is used as a query [4], and we want to find all similar cartoons to it in the database. Since the distance calculation is usually costly for complex data types such as image, one goal is to minimize the number of distance calculations during the search. During preprocessing, pair-wise distances of the three cartoons in the database are calculated and stored. When the query comes, d(Mickey, query) is first calculated. Since the query is also a cartoon of Mickey, we can assume that d(Mickey, query) is small, say 1. Then, from the triangle inequality, it can be derived that $149 \le d(\text{query, Minnie}) \le 151$ and $199 \le d(\text{query, Pluto}) \le 201$. Therefore, neither Minnie nor Pluto is a query result. In a word, using triangle inequality, the similarity query is answered with only one distance calculation.

In the following, we first introduce the concept of an index, then survey common tree structured metric space indices and discuss their problems. Next, the pivot space model, a theoretical framework for metric space indexing, is introduced.

Index

A database index, or simply index, is a data structure to improve the efficiency of data lookup in a database. Answering similarity queries usually consists of two steps.

Offline Construction — Given the data set, construction builds an index data structure offline. The tree structure is one of the most popular metric space indexing structures. In their top-down construction, tree structure metric space indexing methods build index trees by recursively applying two basic steps: pivot selection and data partitioning. In pivot selection, a small number of reference points, called pivots, are selected from the database. In data partitioning, data points are partitioned by their distances to the pivots.

Online Search — Based on the offline built index data structure, similarity queries are answered online. The search process basically descends the index tree from the root to the leaf nodes. At each internal index node, some computation is performed based on the query and the information previously stored in the node. Some children are determined to be unable to contain any query results and can be pruned. Children that cannot be pruned are further visited in the same way. At each leaf index node, similarly, computation is performed to decide which data objects can be pruned and which data objects are query results without calculating their distances to the query. For the remaining data objects, their distances to the query are calculated to find the query results.

	Data type	Distance function
1	Number (one-dimensional)	Absolution value of difference
2	Vector (multidimensional)	Euclidean distance or alike
3	Text	Edit distance
4	Protein sequence	Global alignment (weighted edit distance)
5	Image	Hausdorff distance
6	Video	Percentage of similar frames

Table 2. Examples of metric space.

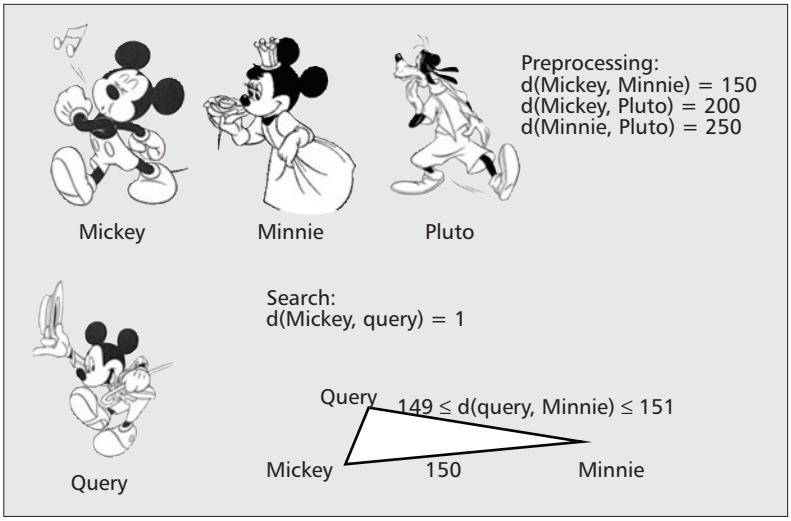

Figure 2. Triangle inequality: the principle of metric space abstraction.

Common Metric Space Indexes

Based on the way data is partitioned, there are two kinds of metric space indices: the vantage point tree (VPT) and general hyper-plane tree (GHT) [5, 6].

During the offline construction of a VPT, the data space is partitioned into disjoint regions recursively. In each recursion, a vantage point, or pivot, is first selected as a reference point. Then the distances from all remaining points to the vantage point are calculated, and the median, m, of the distances is determined. Next, data is partitioned such that points with distances smaller than or equal to m move to the left

Many traditional metric space indexes only show their superiority to others experimentally. A theoretical framework is needed to analyze and compare different indexes and to predict the performance of new indexes.

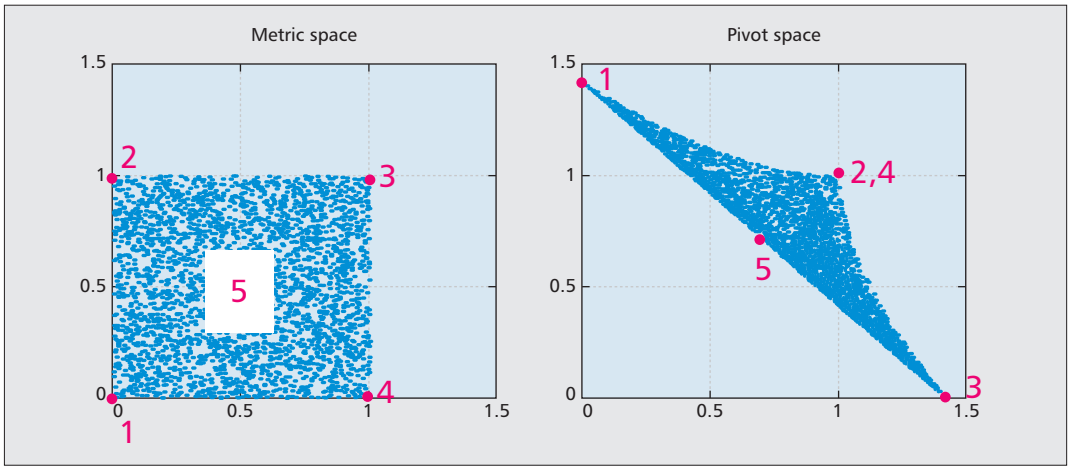

Figure 3. The pivot space of points from unit square with Euclidean distance.

child, and points with distance larger than m move to the right child.

In each recursive construction step of GHT, two points, c_1 and c_2, are selected as centers, or pivots. Each remaining point is assigned closer to the center. That is, points closer to c_1 than to c_2 are assigned to the left child, while points closer to c_2 than to c_1 are assigned to the right child.

Since metric space has no coordinate system, mathematical tools of multidimensional space are not directly applicable to metric space. Consequently, most of above work is based on heuristics. Theoretical analysis is usually overlooked. Nevertheless, a large amount of traditional metric space indices only show their superiority to others experimentally. A theoretical framework is needed to analyze and compare different indices and to predict the performance of new indices. The pivot space model introduced next aims to solve the above problems.

Pivot Space Model: A Theoretical Framework for Metric-Space Indexing

A first goal of the pivot space model is to impose coordinates to data in metric space so that mathematical tools of multidimensional space can be applied for theoretical analysis. To do so, a mapping from metric space to multidimensional space is defined [7].

Let R^m denote a general real coordinate space of dimension m. Let (S, d) be a metric space where $S = \{x_i \mid i = 1, 2, \ldots, n\}$ is the database, and d is a metric distance function. Let $P = \{p_j \mid j = 1, 2, \ldots, k\}$ be a set of k pivots. $P \subseteq S$. Duplicates are not allowed.

Given the set of pivots, each point x in S can be mapped to a point x_p in the non-negative orthant of R^k. The jth coordinate of x_p represents the distance from x to p_j:

$$x_p = (d(x, p_1), \ldots, d(x, p_k)).$$

The **pivot space** [7] of S, $F_{P,d}(S)$, is defined as the image of S:

$$F_{P,d}(S) = \{x_p \mid x_p = (d(x, p_1), \ldots, d(x, p_k)), x \in S\}.$$

Figure 3 gives an example of pivot space. The original data consists of 3000 points uniformly distributed in the unit square. With Euclidean distance, these points form a metric space. Two points, with coordinates (0,0) and (1,1), are selected as the pivots. Since the number of pivots is 2, the pivot space is also 2D. For a point, (a, b), in the original metric space, its coordinate on the x-axis in the pivot space is its Euclidean distance to the first pivot, (0, 0), and its coordinate on the y-axis in the pivot space is its Euclidean distance to the second pivot, (0, 1). Four special points (corners and the center) in the original metric space are selected, and their images in the pivot space are marked.

A **complete pivot space** [7] is a pivot space with all points selected as pivots. It has been proved that the mapping from metric space to a complete pivot space is isometric [7]. That is, the mapping is one-to-one, and the metric space distance of any pair of points equals the L_∞ distance of their images in the complete pivot space. Therefore, instead of the original metric space, the complete pivot space can be searched to answer similarity queries. Since the complete pivot space is a multidimensional space, the problem turns into a multidimensional indexing problem, to which many mathematical tools can be applied.

Furthermore, it has been proved that the partition boundaries of both VPT and GHT are straight lines (or hyper-planes) in the pivot space, with different slopes. In other words, VPT partition and GHT partition are essentially rotations of each other [8]. Moreover, it has been shown theoretically and experimentally that VPT outperforms GHT [8].

Open Issues and Future Work

Based on the survey of metric space indexing, one can deduce a possible paradigm for metric-space-based big data management and analysis tasks other than indexing. That is, we can first map data from metric space to pivot space, and then apply traditional multidimensional space mathematical tools to the pivot space.

Open research issues and future work include, but are not limited to, the following.

Given the isometric mapping, can complete pivot space replace metric space? The pivot

space model defines an isometric mapping from metric space to the complete pivot space. Under this mapping, the pair-wise distances are preserved. For indexing, the complete pivot space can be searched instead of the original metric space. For other tasks (e.g., clustering and classification), can metric space be replaced by the complete pivot space? More theoretical analysis is to be performed on this topic.

If the above is "yes," can we work on the complete pivot space directly? Since all points are selected as pivots, the dimension of the complete pivot space equals the size of data, which is usually huge for big data applications. High dimensionality causes problems for many data management and analysis tasks, such as indexing. It is necessary to identify which tasks can be performed on the complete pivot space directly, and which cannot.

If the above is "no," how should dimension reduction for the complete pivot space be done? According the pivot space model, dimension reduction for the complete pivot space can only select existing dimensions and cannot create new dimensions for indexing. More theoretical analysis is to be done to determine for which data management and analysis tasks that dimension reduction for the complete pivot space model can create new dimensions.

How can performance be improved algorithmically? It is difficult to build a universal system generally performing well. Since metric space abstraction discards domain-specific information and only leverages the triangle inequality of the distance function, it is of key importance to refine the algorithms to achieve acceptable performance.

How can it be done in a parallel or distributed manner? Another way to improve performance is to exploit multiple processors. Parallel and distributed techniques are of great value.

How can metric distance functions be defined for more data types and applications? As discussed earlier, not all data types and distance functions can be abstracted into metric space. Defining proper metric distance functions for data types and applications is of critical importance.

Conclusions

This article focuses on the variety challenge of big data problems in AAL communications systems. First we show that a universal approach is very effective in overcoming variety. Then we show that universality can be achieved by abstraction. Next, metric space is proposed as a universal abstraction for AAL big data. To demonstrate how a metric space data management and analysis system can be built, we survey the state of the art in metric space indexing and introduce the pivot space model. Last but not least, a few important open research issues, which form the direction of future work, are discussed.

Since much less attention has been paid to variety than to volume and velocity, this article provides a novel perspective on designing AAL big data management and analysis systems.

Acknowledgment

This research was supported by the following grants: China 863: 2012AA01A309; NSF-China: 61170076, U1301252, 61471243; Shenzhen Foundational Research Projects SGLH20131010163759789, JCYJ20130401111833183, JCYJ20140418095735561.

References

[1] D. Laney, "3D Data Management: Controlling Data Volume, Velocity and Variety," Gartner, Feb. 2001.
[2] J. Matousek, *Lectures on Discrete Geometry*, Springer-Verlag, 497, 2002.
[3] S. R. Ramakrishnan *et al.*, "A Fast Coarse Filtering Method for Protein Identification by Mass Spectrometry," *Bioinformatics*, vol. 22, no. 12, 2006, pp. 1524–31.
[4] http://www.tom61.com/shaoertuku/jianbihuatupian/2010-12-18/811.html, Oct. 2014.
[5] E. Chavez, *et al.*, "Searching in Metric Spaces," *ACM Computing Surveys*, vol. 33, no. 3, 2001, pp. 273–321.
[6] P. Zezula *et al.*, *Similarity Search: The Metric Space Approach*, Springer, 2006.
[7] R. Mao, W. Miranker, and D. P. Miranker, "Pivot Selection: Dimension Reduction for Distance-Based Indexing," *J. Discrete Algorithms*, Elsevier, 2012, pp. 32–46.
[8] R. Mao *et al.*, "On Data Partitioning in Tree Structure Metric-Space Indexes," *Proc. 19th Int'l Conf. Database Systems for Advanced Applications*, Apr. 21–24, 2014, Bali, Indonesia, pp. 141–55.

Biographies

Rui Mao received his B.S. (1997) and M.S. (2000) in computer science from the University of Science and Technology of China, and another M.S. (2006) in statistics and his Ph.D. (2007) in computer science from the University of Texas at Austin. After three years at Oracle USA Corporation, he joined Shenzhen University in 2010, where he is now an associate professor in the College of Computer Science and Software Engineering. His research interests include universal data management and analysis, and high-performance computing. He has about 50 publications, and his work on the pivot space model was awarded the SISAP 2010 Best Paper award.

Honglong Xu received a B.S. degree in computer science from Shenzhen University in 2010, and is now a Ph.D. student in communication engineering at Shenzhen University.

Wenbo Wu received a B.S. degree in computer science from the University of Texas at Austin, and is now a Ph.D. student of statistics at the University of Georgia.

Jianqiang Li received his B.S. degree in mechatronics from Beijing Institute of Technology, China, in 1996, and his M.S. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2001 and 2004, respectively. He worked as a researcher at the Digital Enterprise Research Institute, National University of Ireland, Galway, in 2004–2005. From 2005 to 2013, he worked at NEC Labs China as a researcher, and the Department of Computer Science, Stanford University, as a visiting scholar in 2009–2010. He joined Beijing University of Technology in 2013 as Beijing Distinguished Professor. His research interests are in Petri nets, enterprise information systems, business processes, data mining, information retrieval, semantic web, privacy protection, and big data. He has over 40 publications and 37 international patent applications.

Yan Li received his B.S. in electronic engineering and information science from the University of Science and Technology of China in 2001, and his Ph.D. from Paris 11 University in 2007. He joined Shenzhen University in 2008, and is now an associate professor in the College of Computer Science and Software Engineering. His research interests include low-noise signal processing and mixed signal IC design.

Minhua Lu received her B.S. in electronic engineering and information science from the University of Science and Technology of China in 2001, and her Ph.D. degree in biomedical engineering from the Hong Kong Polytechnic University, China, in 2007. She joined the Department of Biomedical Engineering, Shenzhen University in September 2007, and is now an associate professor and associate head of the department. Her main research interests include biomedical ultrasound imaging, medical instrumentation, tissue elasticity imaging, and image processing. She has published more than 60 technical papers, and holds 3 invention patents.

Based on the survey of metric-space indexing, one can deduce a possible paradigm for metric space based big data management and analysis tasks other than indexing. That is, we can first map data from metric space to pivot space, and then apply traditional multi-dimensional space mathematical tools to the pivot space.