

# Factor Space, the Theoretical Base of Data Science

Pei-Zhuang Wang · Zeng-Liang Liu · Yong Shi ·  
Si-Cong Guo

Received: 1 July 2014 / Revised: 15 August 2014 / Accepted: 10 September 2014 /  
Published online: 28 October 2014  
© Springer-Verlag Berlin Heidelberg 2014

**Abstract** This paper introduces factor space theory, which provides a general coordinate system to describe the real world and a theoretical base for data science. Based on the theory, factorial databases is presented, which carries a new kind of statistics to do intelligent analysis for coming tide of Big Data.

**Keywords** Factor space · Factorial databases · Background relation · Factorial neural networks · Factor vane · Sample cultivation · Information fusion

**Mathematics Subject Classification** 90C05

## 1 Introduction

Big Data stylishly leads the current tide, various parlances dazzle people in delightful surprise with confusion. However, we are acutely aware that the core task in the

---

P.-Z. Wang (✉) · S.-C. Guo  
College of Intelligence Engineering and Mathematics, Liaoning Technical University, Fuxin 123000,  
Liaoning, China  
e-mail: peizhuangw@126.com

Z.-L. Liu  
National Defense University PLA China, Beijing 100091, China  
e-mail: lgchinaphd@tom.com

Y. Shi  
Research Center of Fictitious Economy and Data Science, Chinese Academy of Science,  
Beijing 100080, China  
e-mail: yshi@ucas.ac.cn

S.-C. Guo  
e-mail: lnzxtaw32@126.com

tide is promoting the intelligence in Big Data. As the journal's preface emphasized [1], data science 'should have its own scientific contents such as axioms, laws and rules, which are fundamentally important for experts in different fields to explore their own interests from Big Data'. Even though there are remarkable achievements in this area, data science still lacks theoretical base on intelligence. As Tsien Hsueshen emphasized [2], 'To develop intelligent engineering, most important task is building the mathematical theory towards intelligence!' This paper aims to introduce Factor space, which provides a general coordinate system for description of things in the world, which is the very mathematical base for data science.

Factor space [3] was published in the same year coincidentally with the formal conceptual analysis [4] and rough sets [5]. The three branches were the pioneers in intelligence mathematics, but the former one had focused on genetic analysis for uncertainty several years.

Factor space is a bridge connecting randomness and certainty. Both ends can be ex-transferred each other according to the varying of the dimension of factors [6]. Based on the idea, intentionally or unintentionally, Kolmogorov presented the fundamental space  $\Omega$ , a factor space, in the axiomatic definition of probabilistic field. He drives randomness into an inevitable framework, took a march of mathematics toward random phenomena. Without the idea of factor space, probability could not realize modernization in the thirties of the last century.

Factor space is a bridge connecting fuzziness and certainty also. This bridge and the bridge mentioned above shows a duality: The fuzziness on the ground, the universe  $U$ , can be viewed as the randomness in the sky, the power  $P(U)$  of  $U$ . Based on the idea of factor space, Wang presented the theory of Fuzzy Shadows [7] to treat fuzzy set as the covering function of a random set, which provides a firm base to fuzzy sets theory and has been applied in fuzzy controllers [8] and several areas. As a summary, the book "Fuzzy System Theory and Fuzzy Computer" [9] was published in 1997.

The trace of factor space on intelligence was shown in the books 'Mathematical theory of Knowledge Representation' [10], 'Theory and Applications of Factorial Neural Networks' [11], 'Attribute method in Thinking and Intelligence Science' [12] etc. Some represented papers can be found in [13–27].

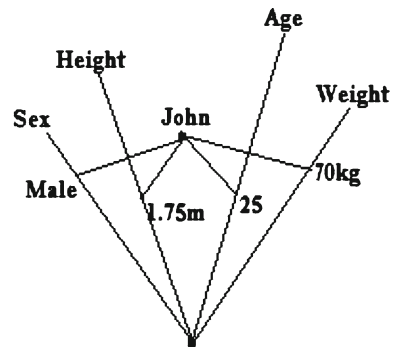
Factor space has common goals with formal concept and rough sets, the authors of this paper emphasize the importance of background relation (called the formal background by R. Wills) and take deep study on the relation. Factor space provides a population theory to information systems in rough sets. All branches will cooperate to establish firm mathematical bases for data science.

Organization of the paper: Sect. 2. Introduction to factor and factors space; Sect. 3. Knowledge representation; Sect. 4. Factorial databases; Discussions including a brief conclusion and main tasks are given in Sect. 5.

Limited to time, all proofs of propositions are omitted.

## 2 What is Factor and Factor Space?

Gene is the key of biology, which forms, generates, and identifies all living objects. There exists a key opening the door to recognize all things in the universe, which is

**Fig. 1** Factor state space

the generalized gene, we call it the *Factor*. The name of gene was called *Mendelian factor* originally; a factor is a fact-or, where ‘fact’ stands for any thing and ‘-or’ is the matter who describes, determines, and identifies all things. The Chinese translation of factor, YINSU, mostly fix to the mentioned meaning, factor is the best name for generalized gene.

A gene likes a switch with two or more states plugged in a node upon chromosome, each state determines a biological property/quality. Gene is the quality-root of living beings. A factor switches a series of states. For example, *Color* is a factor, which switches three basic states: Red, Yellow and Blue. Factor is the quality-root of things.

From the view of mathematics, a factor  $f$  is essentially a mapping, which defined on a domain  $O$ ; every object is mapped into  $X_f$ , the range of  $f$ . A state of  $X_f$  can be an attribute, a feature, a characteristic, or a degree etc., they form a dimension with respect to  $f$ . A factor is the name of the dimension, which hooks a series of attributes. Factor is the attribute of attributes.

Any concrete object has complex quality, which can’t be recognized except taking ‘photo’ from a specified angle/aspect. Factor is the angle of analysis. Without factor, no analysis can be taken.

An object can be analyzed by many factors, and get a record, for example,  $Height(John) = 1.75\text{ m}$ ,  $Weight(John) = 70\text{ kg}$ ,  $Age(John) = 25$ ,  $Sex(John) = \text{Male}$ , etc. Taking a synthesis after analyses, the Cartesian product of those dimensions form a coordinate system, it is the *factor state space*, which is a coordinate system with dimensions named by factors. John has been mapped as a point in the coordinate system. Factor space provides a general coordinate system to describe all things in the universe (see Fig. 1).

Not only does factor space extend the field of vision, but also bring the flexibility to the coordinate system: For a given task, factor space decreases its dimensions as low as possible! There needs to introduce some operations on factors. For example, *Color*, *Aroma*, *Taste* are three simple factors in foods, and *Color–aroma*, *Color–taste*, *Aroma–taste*, *Color–aroma–taste* are complex factors. We can say that factor *Color* is simpler than *Color–aroma*, and denote that  $Color \leq Color\text{--}aroma, \dots$ , the partial order relation ‘ $\leq$ ’ relation can be defined on factors. Based on the poset, we can further define lattice operations  $\vee$  and  $\wedge$  on the set  $F$  of factors. For example,  $Color \vee Aroma = Color\text{--}aroma, \dots$ ,  $Color\text{--}aroma \wedge Color\text{--}taste = Color, \dots$ , where,

we call ‘ $\vee$ ’ the *conjunction* and ‘ $\wedge$ ’ the *disjunction* since ‘ $\vee$ ’ combines simple factors to become a complex factor and ‘ $\wedge$ ’ decomposes complex factors to simple ones.  $(F, \vee, \wedge)$  forms a lattice. Moreover, we can say that the factor *Color–aroma–taste* is complete, if no more factors are needed for food-evaluation. We call it the complete factor, denoted  $\mathbf{1}$ , relatively, we can add the empty factor  $\mathbf{0}$ , which stands for none of aspect, and there is no state to be mapped. For example,  $\mathbf{0}(\text{John}) = \text{none}$ . We can further define the complementary  $^c$ .  $\text{Color}^c = \text{Aroma} \vee \text{Taste}, \dots$  Finally,  $(F, \vee, \wedge, ^c, \mathbf{1}, \mathbf{0})$  forms a Boolean algebra.

The following definition is cited from paper [10] with a little change:

**Definition 2.1** A factor space defined on universe of discussion  $U$  is a family of sets  $\psi = (\{X(f)\}_{f \in F}; U)$  satisfying:

- (1)  $F = (F, \vee, \wedge, ^c, \mathbf{1}, \mathbf{0})$  is a complete Boolean algebra;
- (2)  $X(\mathbf{0}) = \{\emptyset\}$ ;
- (3) For any  $T \subseteq F$ , if  $\{f|f \in T\}$  are *irreducible* (i.e.,  $s \neq t \Rightarrow s \wedge t = \mathbf{0}(s, t \in T)$ ), then

$$X(\{f|f \in T\}) = \prod_{f \in T} X(f) \left( \prod \text{ stands for Cartesian product} \right);$$

- (4)  $\forall f \in F$ , there is a mapping with same symbol  $f : U \rightarrow X(f)$ .

$F$  is called the set of *factors*,  $f \in F$  is called a factor on  $U$ .  $X(f)$  is called the *state space* of factor  $f$ . Set  $F_o = \{f \in F|f \neq \mathbf{0}, \text{ there is no } g \in F; \mathbf{0} \neq g < f\}$ . Any  $f \in F_o$  is called an *atom factor* of  $F$ .

Factor  $f$  is said to be *simple* than  $g$  or  $g$  is said to be *complex* than  $f$  if  $f \leq g$ . For any  $F' \subseteq F$ , easily denote  $F' = \vee\{f|f \in F'\}$  by a same symbol. A complex factor can be written as the conjunction of simpler factors or the set of those simpler factors. We have that  $F = P(F_o) = \{g|g \subseteq F_o\}$ .

### 3 Knowledge Representation

Knowledge is the accumulation of brain’s right reflection upon the real world. People get knowledge from distinguishing. To distinguish object  $u$  from object  $v$ , there comes comparing. Any two objects have both common properties and different properties; they are either the same or different, and they are neither the same nor different. A factor indicates which aspect we compare from. John and Mary are the same with respect to their nationality, but they are different with respect to sex. *Nationality* and *Sex* indicate two different aspects, they are two factors. Without factor’s indication, comparing can not be taken. Mathematically speaking, a comparison is a judgment: ‘Are  $u$  and  $v$  different under factor  $f$ ?’ They are different under  $f$  if and only if their images are different under mapping  $f$ . i.e.,  $f(u) \neq f(v)$ . Two objects are called *distinguishable* if they are different under  $f$ . A group of objects are called distinguishable if any two of them are distinguishable.

**Basic Problem 1** Given a finite set of objects  $U$  and a group factors  $f_1, \dots, f_n$  defined on  $U$ , how to reduce the number of factors to distinguish objects in  $U$ ?

**Proposition 3.1** *If there are two objects  $u, v \in U$ , for any  $f_i$  they are indistinguishable, then basic problem is no answer, and the problem is called un-compatible.*

**Proposition 3.2** *If a group of objects are distinguishable under  $f$  and  $f \leq g$ , then they are distinguishable under  $g$ .*

**Note.** If a group of objects are distinguishable under  $f$  but indistinguishable under  $g$ , then they are distinguishable under  $f \setminus g = f \wedge g^c$ . Of course,  $f \wedge g^c \leq f$ .

Proposition 3.2 shows that the more complex factor, the easier to distinguish. We hope to select factor as simple as possible. Can we select a simplest factor to do  $U$ -distinguishing? No, even though a group of objects are distinguishable under both  $f$  and  $g$  and  $h = f \wedge g \neq \emptyset$ , they may be indistinguishable under  $h$ . So that, there may not exist a unique simplest factor  $f^*$ , under which, the  $U$ -objects are distinguishable. We are forced to find out reduced factors defined as follows:

**Definition 3.1** Given a group of objects  $U$ . Set  $D = \{f \in F \mid U \text{ is distinguishable by } f\}$ , we call a factor  $f \in D$  is a *irreducible* or *reduced* factor for distinguish  $U$  if there is no  $g \in D$  ( $g \neq \mathbf{0}$ ) such that  $g < f$  ( $g < f$  means that  $g \leq f$  and  $g \neq f$ ).

For  $u, v \in U$ , set  $D_{(u,v)} = \{f \mid f \in F_o, f(u) \neq f(v)\}$ , where  $F_o$  is the set of atom factors. Any  $f \in D_{(u,v)}$  is a atom factor. An atom factor is irreducible, and disjunction of two atom factors is  $\mathbf{0}$ . To distinguish all objects in  $U$ , we need to take a congress, selecting one representative  $f_{(u,v)}$  from each group  $D_{(u,v)}$ . The conjunction  $f = \bigvee f_{(u,v)}$  of those representatives must make  $U$  distinguishable. It is easy to prove that  $f$  is a reduced factor for  $U$ -distinguishing. Elections of representative result different reduced factors, we have the following consequence:

**Proposition 3.3** *The set of all reduced factors for distinguishing  $U$  under  $F$  is*

$$D = \{\bigvee \{f_{(u,v)} \in D_{(u,v)} \mid u, v \in U\}\} \tag{3.1}$$

*We leave the concrete algorithm of factor reduction in Sect. 4.*

**Note.** It is noted that the symbol ‘ $\vee$ ’ is not called conjunction but disjunction in rough sets. Attribute reduction (i.e., factor reduction) is a hot topic in rough sets until now. Since it concerns with factor and factors’ operation, especially, concerns with atom factor, such a simple problem becomes complex, even confusion. Related review can be seen in [28].

### 3.1 Classification

When the result of comparing is indistinguishable, there comes the class relation. Given a factor space  $\psi = (\{X(f)\}_{f \in F}; U)$ , for  $g \leq F$ , set

$$R(g) = \{(u, v) \in U \times U \mid u \text{ and } v \text{ are indistinguishable under } g\}$$

which is called the *class relation* on  $U$  under  $g$ .  $R(g)$  is an equivalent relation, and the quotient set is  $C = U_{/R(g)} = \{\mathbf{a}\} = \{u \mid g(u) = \mathbf{a}\} \mid \mathbf{a} \in X(g)\}$ .

**Definition 3.2** Set a mapping

$$\begin{aligned}\chi(g) : U &\rightarrow C = U_{/R(g)} \\ u &\rightarrow [g(u)],\end{aligned}$$

which is called a *factor classification* of  $U$  under  $g$ . When  $g$  is a simple factor  $f$ ,  $\chi(f)$  is called a *single factor's classification*, when  $g = f_1 \vee \dots \vee f_n$ ,  $\chi(f_1 \vee \dots \vee f_n)$  is called a *multiple factors' classification*.

**Proposition 3.4** *The division  $C(f_1 \vee \dots \vee f_n)$  is the product of divisions  $C(f_1), \dots, C(f_n)$ :*

$$C(f_1 \vee \dots \vee f_n) = C(f_1) \times \dots \times C(f_n), \quad (3.2)$$

where  $C(f_1) \times \dots \times C(f_n) = \{\mathbf{a} = [a_1] \cap \dots \cap [a_n] \mid [a_1] \in C(f_1), \dots, [a_n] \in C(f_n)\}$ .

### 3.2 Concept-Forming

What is an object? An object is an investigated grain with variable granularity. In the beginning, it is the whole university with maximal granularity. We can say that when a person was in the moment of his born, the brain has a zero-concept with whole universe as its extension. The process of object-distinguishing, from rough to fine granularity, is the accumulation of knowledge. It indeed is the process of concept-forming, which generates a group of son-concepts from a father-concept by means of taking a classification on the extension of father-concept under certain factor  $F$ .

Concept is the basic unit of knowledge. Knowledge is an embeddable network with the insert part: A father concept  $\alpha$  breeds to a group of son-concepts.

Concept has two forms of description: Intension and extension. Intension is a statement introducing what properties the concept has. Extension is the range of objects having those properties.

Intension essentially tells us what division on the state spaces is taken.

Given a factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  with  $F = f_1 \vee \dots \vee f_n$ . Consider the power of  $X(F)$ ,  $P(X(F)) = (P(X(F)), \cup, \cap, {}^c)$  forms a Boolean algebra.

**Definition 3.3** For any  $a_1 \in X(f_1), \dots, a_n \in X(f_n)$ ,  $\mathbf{a} = (a_1, \dots, a_n)$  is called a *quality-atom* in  $\psi$ . Denote the set of all quality-atoms as  $\mathbf{A}$ , which is an equivalent representation of  $X(F)$ . We can naturally define  $(P(\mathbf{A}), \vee, \wedge, \sim)$  by isomorphism and call it the *statement Boolean algebra* on  $\psi$ , where  $P(\mathbf{A})$  is the power of  $\mathbf{A}$ , and the operations  $\vee, \wedge, \sim$  are isomorphism to  $\cup, \cap, {}^c$  respectively.

**Definition 3.4** A quality-atom  $\mathbf{a}$  is real if there exists an object  $u \in U$  such that  $F(u) = \mathbf{a}$ . The set  $R$  of real quality-atoms is called the *background relation* of factor set  $F = \{f_1, \dots, f_n\}$ , denoted  $R = R(F)$ .

Adopting the symbol  $R$  aims to emphasize that background relation is the real attribute configuration in between factors.

It is obvious that

$$R = F(U) = \{\mathbf{a} = (a_1, \dots, a_n) \mid \exists u \in U; F(u) = \mathbf{a}\} \tag{3.3}$$

Consider the inversed mapping of  $F$  on  $R$

$$F^{-1} : R \rightarrow P(U)$$

$$\mathbf{a} = (a_1, \dots, a_n) \mapsto [\mathbf{a}] = \{u \in U \mid F(u) = \mathbf{a}\}.$$

Extend the domain of mapping  $F^{-1}$  from  $R$  to its power  $P(R)$

$$F^{-1} : P(R) \rightarrow P(U)$$

$$\vee \{\mathbf{a}^k\} \mapsto \cup \{[\mathbf{a}^k]\}.$$

**Proposition 3.5** *The extended mapping  $F^{-1}$  is isomorphisc between  $(P(R), \vee, \wedge, \sim)$  and  $(P(U), \cup, \cap, \subset)$ .*

**Definition 3.5** For any  $A \subseteq R$ , Set  $B = F^{-1}(A)$ .  $\alpha = (A, B)$  is called a *concept*,  $A$  and  $B$  are called its *intension* and *extension* respectively. The Boolean algebra  $(\mathfrak{J}, \vee, \wedge, \sim)$  generated by all concepts with their operations is called the *concept Boolean algebra*. For any  $\mathbf{a} \in R$ ,  $\alpha = (\mathbf{a}, [\mathbf{a}])$  is called an *atom concept*, and the set of all atom concepts is denoted as  $\mathfrak{J}_o$ .

Common expressions do not adopt any concepts in the Boolean algebra except public concepts. A public concept is stated by disjunctive normal form, which is a super rectangle in  $R$ .

**Definition 3.6** Given a factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  with  $F = f_1 \vee \dots \vee f_n$ , a concept  $\alpha = (A, B) \in \mathfrak{J}$  is called a *basic concept* if

1. Its intension is rectangle within  $R$ :  $A = A_1 \times \dots \times A_n \subseteq R$ ;
2. The rectangle can not be extended within  $R$ . i.e., there is no  $B = B_1 \times \dots \times B_n \subseteq R$  such that  $B \supseteq A$ .

It is obvious that all basic concepts form a semi-lattice.

**Definition 3.7** Given  $B \subseteq U$  and  $A \subseteq X(F)$ ,  $(A, B)$  is called an *involution* if  $F(B) = A$  and  $F^{-1}(A) = B$ .

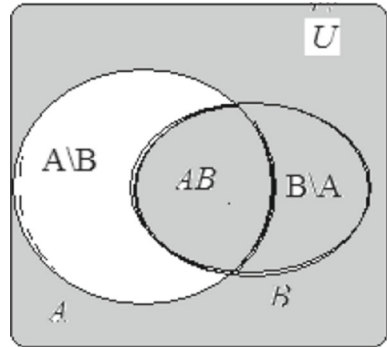
**Proposition 3.6** *For any basic concept  $\alpha = (A, B) \in \mathfrak{J}$ , involution.  $(A, B)$  is an involution.*

**Note.** Involution criterion was presented by Wills in formal concept analysis. It is easy to prove that Definition 3.7 is consistent with his definition on involution. Here, we need to be alert that the concepts in Wills papers are not general concepts, but basic concepts. Furthermore, attributes  $a, b, \dots$  must not have different attribute name. If there are some attributes having the same attribute name, then they do not obey the involution criterion.

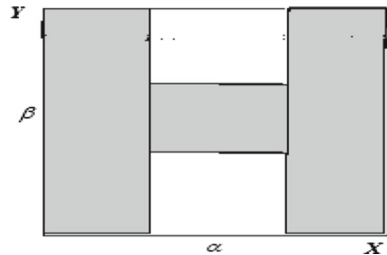
**Basic Problem 2** Given a finite set of objects  $U$  and a group factors  $f_1, \dots, f_n$  defined on  $U$ , how to reduce the number of factors to get all atom concepts?

Since this is a finite sample of factor space, we leave the problem in Sect. 4.

**Fig. 2** Truth field of  $\alpha(u) \rightarrow \beta(u)$



**Fig. 3** Truth field of  $\alpha(x) \rightarrow \beta(y)$



### 3.3 Causal Reasoning

Based on concepts, reasoning is another important topic in knowledge representation. Reasoning/inference is a logical terminology. Sets theory describes concept from extension, which is isomorphic to first order predicate logic. In mathematical logic, a *proposition* is a sentence, which can be judged either Yes or No. A proposition is called a *predicate* if it has the form  $\alpha(u)$  (read ‘ $u$  is  $\alpha$ ’), where  $\alpha$  is a concept and  $u$  is an object in universe of discussion  $U$ .  $\alpha(u)$  is true if and only if  $u \in B$  (extension of  $\alpha$ ).  $B$  is the true-field of  $\alpha$ . The operations of two concepts  $\alpha = (A, B)$  and  $\beta = (A', B')$  are isomorphic to the operations of predicates: The true-field of  $\alpha \vee \beta$  and  $\alpha \wedge \beta$  are  $B \cup B'$  and  $B \cap B'$  respectively. The true-field of  $\sim \alpha$  is  $E^c$ .

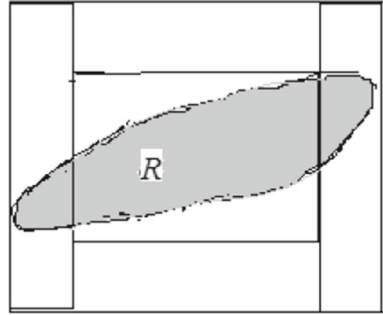
The inference  $\alpha(u) \rightarrow \beta(u)$  (if  $u$  is  $\alpha$  then  $u$  is  $\beta$ ) is a predicate with true-field  $T = (B \setminus B')^c = B \cup B'^c$ . It likes digging a ‘moon’ from  $U$  (see Fig. 2).  $\alpha(u) \rightarrow \beta(u)$  is a tautology if and only if  $T = U$ , i.e.,  $B \subseteq B'$ . The essence of inference is ‘obtained’.

Factor space theory extends the inference form from  $\alpha(u) \rightarrow \beta(u)$  to  $\alpha(x) \rightarrow \beta(y)$  (see [29]):

In a factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  with  $F = f \vee g$ , where  $f$  may be a complex factor  $f = f_1 \vee \dots \vee f_n$ . To simplify, set  $X = X(f)$  and  $Y = X(g)$  and  $U = X \times Y$ . Consider the true-field of the inference  $\alpha(x) \rightarrow \beta(y)$  (if  $x$  is  $\alpha$  then  $y$  is  $\beta$ ), where  $x \in X$ ,  $y \in Y$ ,  $B \subseteq X(f)$ ,  $B' \subseteq X(g)$ . The extension of  $\alpha$  in  $U$  is  $B \times Y$ , and the extension of  $\beta$  in  $U$  is  $X \times B'$ . The true-field of  $\alpha(x) \rightarrow \beta(y)$  is  $T = ((B \times Y) \setminus (X \times B'))^c = (X \times B') \cup (B^c \times Y) = (B \times B') + (B^c \times Y)$ . Its shape resembles the alphabet H (see Fig. 3).



**Fig. 4**  $T^* = T \cap R \supseteq R$



**Definition 3.8** We call factors  $f$  and  $g$  *independent* if  $(f \vee g)(U) = f(U) \times g(U)$ . A group of factors are called *independent* if any two of them are independent.

If  $F = \{f_1, \dots, f_n, g\}$  is a set of independent factors, then all concept Boolean algebra is isomorphic to  $(P(X(F)), \cup, \cap, ^c)$ , and then knowledge representation will be mediocre. When  $F$  is a set of independent factors, it is not possible to make an inference  $\alpha(x) \rightarrow \beta(y)$  to be a tautology unless  $E = \emptyset$ . It means that there is no valuable causal reasoning if factors are independent.

But, when  $f_1, \dots, f_n$  are not independent, the situation becomes complex. Factors are interdependent, they are reciprocal causation, interact as both cause and effect. The decision variable  $g$  occurring in existing models of decision making is also a factor. However, we call  $f$  and  $g$  the *conditional* and *resulted* factors respectively. A factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  is called a *causality factor space* if  $F = f \vee g$  consists of conditional and resulted factors. The tableau representing a sample  $S$  in causality factor space is called a *causality analysis table*. Any decision making essentially is a causal reasoning in a factor space.

Generally,  $R = F(U) \neq X(F)$ , and we must delete all fictitious quality-atoms. Then, the real state space of  $F$  is not  $X(F)$  but  $R$ ! The true-field of  $\alpha(x) \rightarrow \beta(y)$  is not  $T$  but  $T^* = T \cap R = [(B \times B') + (B^c \times Y)] \cap R$ .  $\alpha(x) \rightarrow \beta(y)$  is a tautology if and only if  $T^* = R$ , i.e.,  $[(B \times B') + (B^c \times Y)] \supseteq R$  (see Fig. 4).

Set  $I = \{\alpha(x) \rightarrow \beta(y) | T(\alpha(x) \rightarrow \beta(y)) = R\}$ , which is the set of all tautologies on the factor space. It is obvious that

**Proposition 3.7** If  $\alpha(x) \rightarrow \beta(y) \in I$ ,  $E(\alpha') \subseteq E(\alpha)$  and  $E(\beta') \supseteq E(\beta)$ , then  $\alpha'(x) \rightarrow \beta'(y) \in I$ , where  $E(\gamma)$  is the extensions of concept  $\gamma$ .

Denote  $v(\alpha(x) \rightarrow \beta(y))$  as the *information value* of tautology  $\alpha(x) \rightarrow \beta(y)$ . We know that a wider extension of antecedent gives rise to a narrower extension of succedent, and a higher information value of tautology:

$$v(\alpha(x) \rightarrow \beta(y)) \geq v(\alpha'(x) \rightarrow \beta'(y)) \quad \text{if } E(\alpha') \subseteq E(\alpha) \text{ and } E(\beta') \supseteq E(\beta).$$

Set  $\underline{I} = \{\alpha(x) \rightarrow \beta(y) \in I \mid \text{there is no } \alpha'(x) \rightarrow \beta'(y) \text{ in } I; v(\alpha'(x) \rightarrow \beta'(y)) > v(\alpha(x) \rightarrow \beta(y))\}$

**Definition 3.9**  $\alpha(x) \rightarrow \beta(y)$  is called a *valuable tautology* if  $\alpha(x) \rightarrow \beta(y) \in \underline{I}$ . A valuable tautology is called an *extracted rule* if  $\beta$  is an atom concept.

**Basic Problem 3** Given a factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  with  $F = \{f_1, \dots, f_n, g\}$ , how to get all valuable tautologies?

This is the rule extraction problem in decision making or classification. To answer the problem, there is a fundamental theorem.

**Theorem 3.1** Any valuable tautology is determined by background relation  $R$ :

$$\alpha(x) \rightarrow \beta(y) \in \underline{I} \text{ if and only if } E(\beta) = \{y | \exists x \in E(\alpha); (x, y) \in R\}.$$

By means of the theorem, we can get the succedent  $E(\beta)$  of a valuable tautology for a given antecedent  $E(\alpha)$  by following trilogy:

1. Doing cylinder extension from  $E(\alpha)$ , get  $E(\alpha) \times Y$ ;
2. Intersecting the cylinder with  $R$ , get  $(E(\alpha) \times Y) \cap R$ ;
3. Projecting to  $y$ , get  $E(\beta) = \{y | \exists x \in E(\alpha); (x, y) \in R\}$ .

The concrete algorithms for basic problem 4 are discussed in Sect. 4.

### 3.4 Background Base $B(R)$

From Theorem 3.1, we know that background relation  $R$  is the most important concept in factor space. Given a factor space  $\psi = (\{X(f)\}_{f \in F}; E)$ , how to get the background relation  $R$ ? It is not difficult to get  $R$  by copying all factorial tableaux in practice. There is no need to record the information of objects from the leftist column, only record where they image to. Each sampling tableau cutting out leftist column is a part of  $R$ ,  $R$  is the union of those parts. Even though it is not difficult to get  $R$ , the number of  $R$ -points is time-consuming to manage in big databases. We need to do information compression.

$R$  is a topological concept, no matter how to divide the states  $a_1, a_2, a_3$  in  $R = \text{Height}(U)$ , its character and function in knowledge structure relies on the order of  $a_1, a_2, a_3$  only, no main relation with their concrete division. We can set  $X^*(\text{Height}) = \{1, 2, 3\}$ , replacing  $a_1, a_2, a_3$  by 1,2,3 respectively.  $X^*(\text{Height})$  is a integer state space, called the *offset* of  $X(\text{Height})$ . There comes the *offset transformation*. We omit the definitions and propositions in detail. Offset transformation embeds a state space into a Euclidian space. The transformation is very easy when all states are degrees.

$R$  is an existence, existence is a connection, if 2.1 m and 2.3 m belongs to  $R = \text{Height}(U)$ , why would there not be a man whose height is 2.2 m in the world? Even none in present, there would be one in the future. Furthermore,  $R$  is convex in practice usually. We can make the following assumption:

**Assumption 3.1** The background relation  $R^*$  is integer-convex.

**Definition 3.10** Under the Assumption 3.1, we call the set of vertexes of  $R^*$  the *background base* of  $R$ , denoted as  $B(R)$ .

Since  $R^*$  is convex, which can be generated from  $B(R)$ . The base is much smaller than  $R$ . From  $R$  to  $B(R)$ , an information compression is made. It can be seen as the first mining.

### 4 Factorial Databases

**Definition 4.1** Given a factor space  $\psi = (\{X(f)\}_{f \in F}; E)$  with a name # of concept  $\alpha = (E, F)$ , a sample  $S$  (a group of sample points)  $\subseteq E$  with  $T = (\#; S, F)$  is called a *factorial database*.

$T$  is a general record system. When it becomes a tableau, the tableau is the same as the tableau of information system in rough sets except emphasizing the name of universe. Putting a clear name in each tableau is necessary for information fusion.

The first task of factorial databases is finding out the background relation  $R$ .

#### 4.1 Explanation of $R$ and $B(R)$

Comparing intension and extension, intension is more essential than extension. Indeed, extension is infinite and variable. For example, the extension of Man includes living men, dead or not yet born, it is infinite; the extension of Young Man is variable. John is a young man at someday, while he may not be young some years later. Therefore, strictly speaking, extension is unrealizable, it makes that  $R$  is not calculate-able. But,  $R = F(E)$  is a subset in  $X(F)$ , it is an intension. It consists of finite quality-atoms, and it is stable: For any sample  $S$ , no matter what operations are taken, it does not cut off  $R$ . It is always true that  $F(S) \subseteq R$  and  $R = \cup\{F(S)|S \text{ occurs in a sample}\}$ . Given a factor space  $\psi = (\{X(f)\}_{f \in F}; E)$ ,  $G_o = F(S_o)$  is the image of first sample  $S_o$  in the beginning, it may not be the real background  $R$ . After a series of sampling operations,  $R = F(S_o) \cup \dots \cup F(S_t)$  becomes stable gradually, we call it is nature. A nature background relation  $R$  is not fictitious symbol but a real entity in factorial databases.

#### 4.2 How to Determine the Background Base $B(R)$ ?

Let  $S^* = \{n_k = (n_{k1}, \dots, n_{kn})\}_{(k=1, \dots, m)}$  be an integer-valued sample. To determine its base  $B(S^*)$ , firstly, along each axis, we select the maximum or minimum of  $S^*$ . i.e., For  $t \in \{1, \dots, n\}$ , set

$$*t = \operatorname{argmin}_k \{n_{kt}\}; t^* = \operatorname{argmax}_k \{n_{kt}\} \tag{4.1}$$

**Proposition 4.1** For any  $t \in \{1, \dots, n\}$ ,  $n_{*t}$  and  $n_{t^*}$  are two vertexes of  $S^*$  from lower and upper upon factor  $f_t$ .

We call  $n_{*t}$  and  $n_{t^*}$  the  $t$ -lower vertex and  $t$ -upper vertex respectively. All of them are called the *natural vertexes*.

Apart the natural vertexes, to find out other vertexes of  $S$ , we need to delete inner points of  $S^*$ . According to convex analysis, An inner point  $n = (n_1, \dots, n_n) \in S^*$  requires that  $n$  is a convex combination of  $\{n^t\}_{(t \in T)} \subseteq S^* \setminus \{n\}$ . i.e.,  $\exists \lambda^t \geq 0$  for  $t \in T$  with  $\sum \lambda^t = 1; n = \sum \lambda^t n^t$ . To judge if a point is inner along the way, we will face an N-hard problem. Therefore, we present a new way as follows:

Denote  $\mathbf{n}^* = 1/m \sum \{\lambda^t \mathbf{n}^t | \mathbf{n}^t \in S^*\}$ , where  $m$  is the number of points of  $S^*$ , which is the center of  $S^*$ .

**Proposition 4.2** *A point  $\mathbf{n} = (n_1, \dots, n_n) \in S^*$  is an inner point of  $S^*$  if there are two points  $\mathbf{a}, \mathbf{b} \in S^* \setminus \{\mathbf{n}\}$  such that  $(\mathbf{a} - \mathbf{n}, \mathbf{n}^* - \mathbf{n}) < 0$ ,  $(\mathbf{b} - \mathbf{n}, \mathbf{n}^* - \mathbf{n}) < 0$  and  $(\mathbf{a}, \mathbf{b}) \geq 0$ , where  $(\mathbf{x}, \mathbf{y})$  is the inner product of  $\mathbf{x}$  and  $\mathbf{y}$ .*

Algorithm **InnerPointJudgment** $[S^*, \mathbf{n}]$  (Inner point judgment $[S^*, \mathbf{n}]$ )

Input  $S^*$  and  $\mathbf{n}=(n_1, \dots, n_n)$

Calculate  $\mathbf{n}^*$ ;

If there are two points  $\mathbf{a}, \mathbf{b} \in S^* \setminus \{\mathbf{n}\}$  such that  $(\mathbf{a} - \mathbf{n}, \mathbf{n}^* - \mathbf{n}) < 0$ ,  $(\mathbf{b} - \mathbf{n}, \mathbf{n}^* - \mathbf{n}) < 0$  and  $(\mathbf{a}, \mathbf{b}) \geq 0$ , then make judgment:  $\mathbf{n}$  is an inner point of  $S^*$ ; Else, No judgment.

The complexity of the algorithm is  $O(mn)$ , where  $m$  is the number of points in  $S^*$ ,  $n$  is the number of dimension of state space, i.e., the number of reduced factors.

Algorithm **BackgroundBase** $[S^*]$  (Background base $[S^*]$ )

Input  $S^*$ .

$N := \emptyset$ ;

1. For  $t = 1, \dots, n$ , Calculating  $*t$  and  $t^*$ ;  $N := N \cup \{\mathbf{n}^*t, \mathbf{n}_t^*\}$ ;
2. For each  $\mathbf{n}=(n_1, \dots, n_n) \in R^* \setminus N$ , do **InnerPointJudgment** on  $\mathbf{n}$ ; If  $\mathbf{n}$  is an inner point of  $S^*$ , then delete  $\mathbf{n}$  from  $S^* : S^* := S^* \setminus \{\mathbf{n}\}$

Finally, get output:  $B(S^*) := S^*$ . The complexity of the algorithm is  $O(m^2n)$

Related paper can be seen in [30].

Inversely, we can generate  $S^*$  from  $B(S^*)$

**Proposition 4.3** *Given a background base  $B$ , we can get background  $R$  such that  $B = B(R)$ .*

Now we can see how important background base is: Background base  $B(R)$  is equivalent to background relation  $R$ , it is information compression of  $R$  without information loss.

### 4.3 Basic Problems in Factorial Databases

**Basic Problem 1** Given a sample  $S = \{u_1, \dots, u_m\} \subseteq U$  and a group factors  $f_1, \dots, f_n$  defined on  $U$ , how to reduce the number of factors to distinguish any two objects in  $S$ ?

Any two objects in  $S$  are distinguishable if and only if all indistinguishable classes are singletons.

Treating  $X(f) = \{a_1 < \dots < a_n\}$  as a poset with total order  $<$ .

Algorithm **IndexPermutation** $[S, f]$  (Index permutation $[S, f]$ )

Input  $S = \{u_1, \dots, u_m\}$

(1)  $:= 1$ ; (2)  $:= 2, \dots, (m) := m$ ;

For  $j = 1$  to  $m$   
 For  $k = 1$  to  $j - 1$   
 If  $f(j) < f(k)$  then take permutation for  $(j)$  and  $(k) := (j); (j) := (k); (k) := *$ ;  
 Output 1. New order:  $(1), \dots, (m)$ ;  
 2. Sub-classes:  $C_1, \dots, C_r$ .  
 The number of comparing operations is  $m(m - 1)$ .  
 The result of the algorithm is dividing classes according to the order of indices:  
 (see paper [31])

$$S = (u_{(1)}, \dots, u_{(n_1)}; u_{(n_1+1)}, \dots, u_{(n_1+n_2)}; \dots; u_{(n_1+\dots+n_{r-1}+1)}, \dots, u_{(m)}),$$

$$f(u_{(1)}) = \dots = f(u_{(n_1)}) = a_1, \dots, f(u_{(n_1+\dots+n_{r-1}+1)}) = \dots = f(u_{(m)}) = a_r.$$

The number of objects in sub-classes are  $n_1, n_2, \dots, n_r$ . Set

$$d = 1 - [n_1(n_1 - 1) + \dots + n_r(n_r - 1)]/m(m - 1)$$

**Definition 4.2**  $d$  is called the *distinguishing number* of  $f$  on  $S$ .

**Algorithm DistinguishingReduction** $[S; f_1, \dots, f_n]$  (Distinguishing reduction $[S; f_1, \dots, f_n]$ )

Input  $S; F = \{f_1, \dots, f_n\}$

$C := S$ ;

Set of reduced factors  $R := \emptyset$ ;

Step 1. Select factor  $f$  from  $F$  such that  $d$  is maximal on  $C$ ;

Do *IndexPermutation* $[C, f]$ ;

If  $C = C_k$  is divided into a group of son-classes, i.e.,  $r > 1$ , then set the first son-class as new class  $C_k := C_{k1}$ ; Return to step 1; Else, there is no son-class of  $C$ , which is not dividable, then set the next younger brother-class  $C_{k+1}$  as the new class  $C := C_{k+1}$ ; Return to step 1; The process stop when there is no divisible sub-class existed.

**Basic Problem 2** Given a finite set of objects  $E$  and a group factors  $f_1, \dots, f_n$  defined on  $E$ , how to reduce the number of factors to get all atom concepts?

There are algorithms similar to algorithms in basic problem 1.

**Definition 4.3** Let  $k$  be the number of sub-classes of  $E$  in the classification under factor  $f$ , which is called the *classifying number* of  $f$ .

**Algorithm ClassifyingReduction** $[E; f_1, \dots, f_n]$  (Classifying reduction $[E; f_1, \dots, f_n]$ )

Input  $E; F = \{f_1, \dots, f_n\}$

$C := E$ ;

Set of reduced factors  $R := \text{empty}$ ;

Step 1. Select factor  $f$  from  $F$  such that *the* classifying number  $k$  is maximal on  $C$ ;  
 Do *IndexPermutation*[ $C, f$ ];  
 Record basic concepts corresponding to sub-classes;

If  $C = C_k$  is divided into a group of son-classes, i.e.,  $r > 1$ , then set the first son-class as new class  $C_k := C_{k1}$ ; Return to step 1; Else, there is no son-class of  $C$ , which is not divisible, then record the atom concept corresponding to  $C$ ; Set the next younger brother-class  $C_{k+1}$  as the new class  $C := C_{k+1}$ ; Return to step 1; The process stops when there is no dividable sub-class existed.

**Basic Problem 3** In a causality factor space  $\psi = (\{X(f)\}_{f \in F}; U)$  with  $F = \{f_1, \dots, f_n, g\}$ , for a finite sample  $S \subseteq U$ , how to extract rules with minimal number of reduced factors?

There is the *classify-in* method. ‘Classify’ indicates that conditional factors classify  $U$  into classes finer and finer such that  $g$ -classes swallow  $f$ -classes more and more. Since the essential of reasoning is ‘contained by’, once a  $f$ -class  $A \subseteq B$  ( $g$ -class), a tautology  $A \rightarrow B$  is extracted. Class  $A$  sacrifices itself to extract a rule  $A \rightarrow B$ , this is the idea of *classify-in* method’. This idea comes from paper [32] and improved to get good tests by Huadong Wang [33]. Comparing with decision tree methods ID3 and C4,5 on cancer diagnosis. Bao et al. [34] presents a slightly different method to deal with the basic problem for quantitative factors.

Given a class  $C$  including  $m$  objects, which is divided by factor  $f$  as a group of sub-classes  $\{C_i\}$ , where  $C_i$  includes  $n_i$  objects. We call  $C_i$  a determined class if for any  $u \in C_i$  there is a common state  $g(u) \in X(g)$ . Intuitively speaking, a determined class is swallowed by a  $g$ -class.

**Definition 4.4** Set

$$c = (1/m) \sum \{n_i | C_i \text{ is a determined class}\}$$

which is called the *determined degree* of conditional factor  $f$  on  $C$  for resulted factor  $g$ .

**Algorithm ClassifyInRuleExtraction**[ $S; f_1, \dots, f_n, g$ ] (Classify-in rule extraction [ $S; f_1, \dots, f_n, g$ ])

Input  $S; F = \{f_1, \dots, f_n, g\}$

$C := S$ ;

Set of reduced factors  $R := \emptyset$ ; Rule set  $L := \emptyset$ ;

Step 1. Select factor  $f$  from  $F$  such that *the* determined degree  $c$  is maximal on  $C$ ;

Do *IndexPermutation*[ $C, f$ ];  $R := R \cup \{f\}$ ; Get finer classes

Record basic concepts corresponding to sub-classes;

Extract determined classes as rules; Put those rules into  $L$ ;

If  $C = C_k$  is divided into a group of son-classes, i.e.,  $r > 1$ , then set the first son-class as new class  $C_k := C_{k1}$ ; Return to step 1; Else, there is no son-class of  $C$ , which is not dividable, then record the atom concept corresponding to  $C$ ; Set the next younger brother-class  $C_{k+1}$  as the new class  $C := C_{k+1}$ ; Return to step 1; The process stop when there is no dividable sub-class existed.

Output: All extracted rules

**Applying algorithm *ClassifyInRuleExtraction*[ $E; f_1, \dots, f_n, g$ ] to decision making**

Given a testing point  $i' = (i_1, \dots, i_n)$ , what is the resulted value?

**Solution:** If there is a sample point in  $E$  having same antecedent value, then take its  $g$ -value as the resulted value; Else, declare: No recognition.

The training accuracy = 100 %; the indistinguishableness is high when the sample of background  $R$  is not mature. But, there is no need to worry about the indistinguishable, Getting solution from experts, Put the new point into  $S$  such that  $S$  becomes more mature.

Another way: Viewing each vertex  $i = (i_1, \dots, i_n, j)$  as a rule  $a_{1i_1} \wedge \dots \wedge a_{ni_n} \rightarrow b_j$  (If  $x_1$  is  $a_{1i_1} \wedge \dots \wedge x_n$  is  $a_{ni_n}$  then  $g$  is  $b_j$ )

**Algorithm *BackgroundBaseRuleExtraction***[ $S^*; f_1, \dots, f_n, g$ ] (Background base rule extraction [ $S^*; f_1, \dots, f_n, g$ ])

Input Sample  $S^*$ , factors  $f_1, \dots, f_n, g$

1.  $R^* := S^*$ ; Do Algorithm *Backgroundbase*[ $S^*$ ];
2. Transfer each point  $i = (i_1, \dots, i_n, j)$  to a rule

Output: All rules

**Application algorithm *BackgroundBaseRuleExtraction***[ $S^*; f_1, \dots, f_n, g$ ] to DM

Given a testing point  $i' = (i_1, \dots, i_n; ?)$ , what is the resulted value  $j$ ?

**Solution:** Set  $\underline{S}^* := \{\underline{i} = (i_1, \dots, i_n) | \exists i = (i_1, \dots, i_n, j) \in S^*\}$ . The unknown value  $j$  is the weighted average of  $g$ -values of  $S^*$ -points according to the distances between  $i'$  and  $\underline{S}^*$ -points.

The authors guess that training accuracy is 100 % when  $i'$  is an inner point of  $\underline{S}^*$ ; The solution fails when  $i'$  is not an inner point of  $\underline{S}^*$ . Again, do not worry on failing, Getting solution from experts, and put the new point into  $R$  such that  $R$  becomes more mature.

#### 4.4 Characters of Factorial Databases and Cultivation of Samples

Does data science need statistics? Yes, databases mainly come from and are used by statistical departments. But, intelligent data need new kind of statistics. The paper shows factorial statistics has its own characters as follows:

1. Population of factorial databases is background relation  $R$ , the expected points of  $R$ -distribution is not located at the center (expectation) but the border, where they are not dense but rare.
2. Factorial databases is active, no matter how big the data is, the base of background relation keeps a stable size, automatically realizing the simple–complex bi-direction transformations.
3. Factor space vane is an embedded neural network, factorial databases can do information fusion, from small to great, build a self organized ecological system. This is a great intelligent engineering.

4. Most important character: ample is not only a tool but the object of sample-cultivation.

A good sample approximates to background relation  $R$ . Once it is the background relation  $R$ , it implicates related knowledge perfectly and becomes a knowledge databases. All valuable information absorbs into the databases.

Facing the challenge of intelligence, Codd [35] put relation, a mathematical concept, into databases, and brought up a bright generation for data science. He did not know what kind of relation is the key relation, and left some doubtful points. Factorial databases has found which is the key relation, a new generation of data science is coming.

## 5 Discussions

Section by section, this paper mathematically describes object-distinguishing, concept-forming and causal reasoning the three basic units of thinking process and knowledge accumulation. All of the three topics accept and obey language of factor space both. The brief conclusion is: Factor space is a mathematical theory of intelligence, and factorial databases will play an important role in intelligence engineering.

Limit to time, the paper can not introduce all important tasks and topics researched in the past, present and in the future, we only list advanced tasks as follows:

1. Classification becomes a simple task within a known framework of factor space, while practical classification problems are so difficult! The problem is: How to mine the hidden factors? This is the topic of feature extraction. Some alternative branches such as support vector machine and so on are popular approaches now. Factor space needs its own methods to resolve this problem. The background base mentioned in Sect. 3.4 provides a tool to extract the vertex-character of son-concepts, which will be effect in classification. Another way is using causal reasoning rules to define the intension of son-concepts. Using reasoning back to concept-forming, this is an important approach.
2. Factor neural network presented by Liu [11, 21, 22] provides a embedded framework to building knowledge/information self-organized ecological system in the future;
3. Factor space vane developed by Wang and Li [9, 10, 15, 16] provides mathematical structure for factorial NN, and provides an equipment for non-structure information processing;
4. Yuan [25], Li [9, 10, 15, 16], Zou and Feng [12] presented factorial category theory for information category and information fusion. The language transla-functor (a translator built by functor theory) can be generated for unifying intelligence languages, and also unifying the editions of factorial dictionary/lexicon.
5. Replacing clear states by fuzzy states in the factorial state space, all fuzzy concepts will fold into factorial databases. Realizing the bi-direction transformations between databases and experts, the constructed self-organized ecological system will actively communicate with human being.



6. Variable weights in factor space presented by Wang and Li [9,10,15,16] brings deeper developing for factor space theory. It concerns with the agility of thinking process.
7. Potential optimization on factor space is incubated by Wang. He presents potential gradient in the paper [36], and brings deeper developing for factor space theory. It concerns with the insight and aha experience in thinking process. With regret, he has some typing errors in the paper such as: ‘polynomial:time’ should be ‘polynomial–time’ in the title. In Example 4.1,  $F_0 = (7, 4, 7, 6, 5)$  should be  $F_0 = (7, 4, 7, 9, 5)$ .

**Acknowledgments** The authors wish to thank all friends for their valuable critics, comments and assistances on this paper. This study was partially supported by the Grants (Grant Nos. 61350003, 70621001, 70531040, 90818025) from the Natural Science Foundation of China.

## References

1. Shi Y, Tian YJ (2014) Inaugural volume of the annals of data science: optimization and data science. *Ann Data Sci* 1(1):1–3
2. Hsueshen T (2008) Selected letters of Tsien Hsueshen. National Defence Industry, Beijing
3. Wang PZ, Sugeno M (1992) The factors field and background structure for fuzzy subsets. *Fuzzy Math* 2:45–54
4. Wille R (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ivan R (ed) *Ordered set*. Reidel, Boston, pp 445–470
5. Pawlak Z (1982) Rough sets. *J Comput Inf Sci* 11:341–356
6. Wang PZ (1981) Random differential equations. In: Hao BL et al (eds) *Advance of statistical physics*. Science, Beijing
7. Wang PZ (1985) *Fuzzy sets and falling shadow of random sets*. Beijing Normal University Press, Beijing
8. Wang PZ, Zhang HM (1989) Truth-valued flow inference and its dynamic analysis. *J Beijing Norm Univ* 1:1–12
9. Wang PZ, Li HX (1997) *Fuzzy system theory and fuzzy computer*. Science, New York
10. Wang PZ, Li HX (1994) *A mathematical theory on knowledge representation*. Tianjin Scientific and Technical, Tianjin
11. Liu ZL, Liu YC (1992) *Theory of factorial neural networks*. Beijing Normal University Press, Beijing
12. Feng JL (1990) *Attribute method in thinking and intelligence science*. Atomic Energy, Beijing
13. Kandel A, Peng XT, Cao ZQ, Wang PZ (1990) Representation of concept by factor space. *Cybern Syst* 21(1):43–57
14. Li HX (1995) Looking for the success of fuzzy logic from the mathematical essential of fuzzy control. *Fuzzy Syst Math* 4(9):1
15. Li HX (2000) Factor space theory and its applications in fuzzy information processing: two kinds of factor spaces canes. *Comput Math Appl* 40:835–843
16. Li HX (2014) *Factor space theory*. Collected works on factor space. Liaoning Technical University, Fuxin, pp 57–161
17. Wang PZ, Zhang HM (1989) Truthvalue inference and its dynamic description. *Beijing Norm Univ Nat Sci* 1:1–12
18. Wang PZ, Zhang DZ (1988) Probe on mathematical form of thinking. *J Univ Appl Math* 1(1):85–95
19. Wang PZ (1990) A factor spaces approach to knowledge representation. *Fuzzy Sets Syst* 36(1):113–124
20. Wang PZ (1990) Factor space and knowledge representation. In: Verdegay JL, Delgado M (eds) *Approximate reasoning tools in artificial intelligence*. TUV, Rheinland, pp 97–114
21. Liu ZL, Liu YC (1994) *Theory and applications of factorial neural networks*. Guizhou Normal Science and Technology, Guiyang
22. Liu ZL (2014) *Factor representation of knowledge*. In: *Collected works on factor space*. Liaoning Technical University, Fuxin, pp 171–175

23. Wang PQ, Li XH (2010) A description method on classification of equipment information based on knowledge factor expression theory. *J Natl Univ Defense Technol* 33(5):150–155
24. Chen YY, Chen TY (1984) Eigenvalue expansion method in approximate reasoning. *J Liaoning Norm Univ Nat Sci* 3:1–7
25. Yuan XH, Wang PZ (1995) Factor space and category. *Fuzzy Syst Math* 2:26–33
26. Yu FS, Luo CZ (1999) Granule factors space and intelligent diagnostic expert systems. In: *Advances of electric mathematics. Proceedings of the 7th national conference on electric mathematics. China Science and Technology, Beijing*, pp 24–27
27. Wang HD, Guo SC (2014) Feedback extension envelop and its improvement. *Fuzzy Syst Math* (in press)
28. Yan GJ, Dai N, Wang PZ Information system and factor space (working paper)
29. Liu HT, Wang HD, Wang PZ, Guo SC The base of background relation (working paper)
30. Wang PZ (2014) Factor space and data science. *J Liaoning Eng Technol Univ* (in press)
31. Wang PZ (2013) Factor space and factor data bases. *J Liaoning Eng Technol Univ* 32(10):1–8
32. Wang PZ, Guo S, Bao YK, Liu HT (2014) Causality analysis in factor spaces. *J Liaoning Eng Technol Univ* 33(7):1–6
33. Wang HD, Wang PZ, Guo SC (2014) Improved factor analysis algorithm on factor spaces (working paper)
34. Bao YK, Ru HY, Jin SJ (2014) Factorial analysis, a new algorithm of knowledge mining in causality space. *Liaoning Tech Univ* 33(8):1141–1144
35. Codd EF (1970) A relational model of data for large shared data banks. *Commun ACM* 13(6):377–387
36. Wang PZ (2014) Discussions on Hirsch conjecture and existence of strongly polynomial–time simplex variants. *Ann Data Sci* 1(1):41–71



**Pei-Zhuang Wang** received the BS degree in Mathematics from Beijing Normal University, China in 1957. He serves as Honorary Director of Institute of Intelligence Engineering and Mathematics, Liaoning Technical University, His research interests include optimization and factorial analysis applied in data science.



**Zeng-Liang Liu** received the Dr. degree in Mathematics from Beijing Normal University, China, in 1991, and finished his post-doc in the Department of Computer Science, Beijing University of Aeronautics and Astronautics, China, in 1993. Now, he is the professor of National Defense University PLA China. His main research directions include intelligence system engineering, C4ISR, and network information security.



**Yong Shi** serves as the Executive Deputy Director, Chinese Academy of Sciences Research Center on Fictitious Economy and Data Science and Director, the Key Research Lab on Big Data Mining and Knowledge Management, Chinese Academy of Sciences. He has been the Union Pacific Chair of Information Science and Technology, University of Nebraska at Omaha, USA. Dr. Shi's research interests include business intelligence, data mining, and multiple criteria decision making. He has published more than 22 books, over 200 papers in various journals and numerous conferences/proceedings papers.



**Si-Cong Guo** graduated from Fuxin Mining Institute, in China 1972. He serves as Executive Deputy Director of Institute of Intelligence Engineering and Mathematics at Liaoning Technical University. Main research interests: soft computing in information science and fuzzy analysis.