Granular computing and soft computing have begun to play important roles in bioinformatics, e-business, security, machine learning, data mining, high-performance computing and wireless mobile computing in terms of efficiency, effectiveness, robustness and uncertainty. IJGCRSIS aims to publish the latest research and development results and experiences in the areas of granular computing, rough sets, fuzzy sets, soft computing and intelligent systems.

The major objective of IJGCRSIS is to stimulate new multidisciplinary research and the development of cutting-edge methods, techniques and tools to solve problems in the above areas. The goal is to help readers understand state-of-the-art techniques/algorithms/methods in soft computing and granular computing.

IJGCRSIS provides a forum to help academics, practitioners, post-graduates and policy makers, working in the area of granular computing, soft computing (rough sets, fuzzy sets, evolutionary computing, neural networks, genetic computing, etc.), to disseminate information and to learn from each other's work. The intended audiences are granular and soft computing researchers/practitioners, and specialists in academia and industry.

IJGCRSIS publishes original research papers (long and short papers, exploratory papers), review papers, technical reports, case studies, conference/workshop reports, application notes, book reviews, commentaries, and news. Special Issues devoted to important topics in granular computing and soft computing, selected from top related conferences such as IEEE GrC, RSFDGrC, IEEE FUZZ, IEEE NN, WCCI and related workshop etc., will occasionally be published.

**Subject coverage**

- Granular and rough computing theory and applications
- Computing with words
- Data mining
- Machine learning
- Fuzzy set theory and applications
- Evolutionary computing
- Neural networks
- Web informatics, Web intelligence and mining
- Data security
- Granular computing in bioinformatics

**Submission of papers**

Papers, case studies, etc. in the areas covered by IJGCRSIS are invited for submission. Authors may wish to send an abstract of proposed papers in advance. Notes for intending authors can be found at: https://www.inderscience.com/papers

All papers must be submitted online.

**Authors of accepted papers will receive a PDF file of their published paper. Hard copies of journal issues may be purchased at a special price for authors from subs@inderscience.com**

All editorial correspondence (but not subscription orders) should be addressed to:

**Professor Xiaohua (Tony) Hu**  
Drexel University  
College of Information Science and Technology  
Philadelphia PA 19104, USA  
Email: thu@cis.drexel.edu

or

**Professor T.Y. Lin**  
San Jose State University  
Department of Computer Science  
San Jose, CA 95192, USA  
Email: tylin@cs.sjsu.edu

A copy of the submitted paper and submission letter should also be sent via email to the IEL Editorial Office at:  
Email: editorial@inderscience.com  
Fax: (UK) +44 1234-240515  
Website: www.inderscience.com

Neither the editor nor the publisher can accept responsibility for opinions expressed in the International Journal of Granular Computing, Rough Sets and Intelligent Systems nor in any of its special publications.

**Subscription orders**

IJGCRSIS is published in four issues per volume. A Subscription Order Form is provided in this issue.  
Payment with order should be made to:  
Inderscience Enterprises Ltd. (Order Dept.),  
World Trade Centre Building 11,  
29 Route de Pre-Bois, Case Postale 856,  
CH-1215 Genève 15, Switzerland.  
You may also FAX to:  
(UK) +44 1234 240 515  
or Email to subs@inderscience.com

**Electronic PDF files**

IJGCRSIS papers are available to download from website: www.inderscience.com  
Online payment by credit card.

**Advertisements**

Please address enquiries to the above-mentioned Geneva address or  
Email: adverts@inderscience.com
## Contents

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Preface</td>
<td>Tsau Young Lin and Xiaohua (Tony) Hu</td>
</tr>
<tr>
<td>1</td>
<td>Assessment of candidate information models for granular computing</td>
<td>Steven A. Demurjian</td>
</tr>
<tr>
<td>21</td>
<td>Granular computing I: the concept of granulation and its formal model</td>
<td>Tsau Young Lin</td>
</tr>
<tr>
<td>43</td>
<td>Data mining in rough relational databases by attribute generalisation</td>
<td>Theresa Beaubouef and Frederick E. Petry</td>
</tr>
<tr>
<td>64</td>
<td>Time series data analysis in multiple granularity levels</td>
<td>Mehmet Sayal and Ming-Chien Shan</td>
</tr>
<tr>
<td>81</td>
<td>Drawing reasonable conclusions from information under similarity</td>
<td>Ronald R. Yager</td>
</tr>
<tr>
<td>105</td>
<td>A new fuzzy $k$-modes clustering algorithm for categorical data</td>
<td>Michael K. Ng and Liping Jing</td>
</tr>
</tbody>
</table>
**IJGCRSIS Subscription Order Form**

**Volume 1, 2009**

*(THIS FORM MAY BE PHOTOCOPIED)*

**Subscription Price and Ordering Information:**
The *International Journal of Granular Computing, Rough Sets and Intelligent Systems* is published four times a year (in one volume of four issues), in English.

**Subscription for hard copy or online format** (one simultaneous user only) € 494 per annum (including postage and handling).

**Subscription for hard copy AND online format** (one simultaneous user only) € 672

Airmail option € 40 per volume extra.

Prices for multi-simultaneous users are available on request.

Subscription orders should be addressed to the publishers:
Inderscience Enterprises Ltd (Order Dept.), World Trade Centre Building 11, 29 route de Pre-Bois, Case Postale 856, CH-1215 Genève 15, Switzerland.

- **Payment with order:**
  Cheques or bankers drafts should be sent with order, made payable to: **Inderscience Enterprises Ltd.**
  Credit card payments will be accepted and will be converted to £ Sterling at the prevailing rates.

  *For rush orders, contact:*
  Fax: (UK) +44 1234 240 515
  Website: www.inderscience.com or Email to subs@inderscience.com

**Please tick if you would like details of other Inderscience publications**  

<table>
<thead>
<tr>
<th>Item</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subscription to Volume 1, 2009</strong></td>
<td>€ ...............</td>
</tr>
<tr>
<td><strong>Please dispatch my order by air mail (add € 40 per Volume):</strong></td>
<td>€ ...............</td>
</tr>
<tr>
<td><strong>I enclose total payment of</strong></td>
<td>€ ...............</td>
</tr>
<tr>
<td><strong>Name of Subscriber</strong></td>
<td>..........................................................</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>..........................................................</td>
</tr>
<tr>
<td><strong>Company/Institution</strong></td>
<td>..........................................................</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>..................................................................................</td>
</tr>
<tr>
<td><strong>Fax</strong></td>
<td>.............................................</td>
</tr>
<tr>
<td><strong>Email</strong></td>
<td>..........................................................</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>..........................................................</td>
</tr>
<tr>
<td><strong>Signature</strong></td>
<td>..........................................................</td>
</tr>
</tbody>
</table>

**I wish to pay by credit card:***

- **I authorise you to debit my account with the amount in GBP sterling equivalent to** € ...............  
- **Three digit security number (on reverse of card):** ...............  
- **Card No.** .................................................. **Expire Date** ..................................................  
  **Signature** .................................................. **Date** ..................................................  

Please tick if you would like details of other Inderscience publications
Editorial Advisory Board (continued)

Professor Bo Zhang
Tsinghua University
Department of Computer Science
Beijing 100084, P.R. China

Professor Ning Zhong
Maebashi Institute of Technology
Department of Life Science and Informatics
460-1 Kamisadori-Cho
Maebashi-City 371-0816, Japan

Members of the Editorial Board

Professor Valentina E. Balas
Aurel Vlaicu University of Arad
Faculty of Engineering
Department of Automation and Applied Informatics
77 B-dul Revolutiei, 310130 Arad, Romania

Professor Tzung-Pei Hong
National University of Kaohsiung
Department of Computer Science and Information Engineering
No. 700, Kaohsiung University Road
Nan-Tzu District, Kaohsiung 811
Taiwan, ROC

Dr. David W.L. Cheung
University of Hong Kong
Department of Computer Science
Pokfulam Road, Hong Kong (China)

Professor Lakhmi C. Jain
Professor of Knowledge-Based Engineering
Founding Director of the KES Centre
University of South Australia
Division of Information Technology, Engineering and the Environment
School of Electrical and Information Engineering, Mawson Lakes Campus
Adelaide, South Australia 5095, Australia

Professor Jian-Hua Dai
Zhejiang University
College of Computer Science and Technology
38# Zheda Road
Hangzhou 310027, P.R. China

Professor Zhoujun Li
Beihang University
Department of Computer Science and Engineering
1550, Pudong Avenue
Wuhan Hubei, 430079, P.R. China

Professor Antonio Di Nola
University of Salerno
Department of Mathematics and Computer Science
84084 Fisciano (SA), Italy

Dr. Churn-Jung Liau
Academia Sinica
Institute of Information Science
Taipei 115, Taiwan, ROC

Professor Jerzy Grzymala-Busse
University of Kansas
Department of Electrical Engineering & Computer Science
3014 Eaton Hall, 1520 West 15th Street
Lawrence, KS 66045-7621, USA

Professor Guilong Liu
Beijing Language and Culture University
School of Information Science
Beijing 100083, P.R. China

Professor Tingting He
Central China Normal University
Department of Computer Science
152 Luoyu Road
Wuhan Hubei, 430079, P.R. China

Professor Qing Liu
Nanchang University
Department of Computer Science
Nanchang, Jiangxi 330031, P.R. China

Professor Caro Lucas
University of Tehran, Department of Electrical and Computer Engineering
PO Box 14395, 515, Tehran, Iran

Prof. Dr. Hans Hellendoorn
University of Technology Delft
Delft Center for Systems and Control
Mekelweg 2, 2628 CD Delft, Netherlands

Dr. Sadaaki Miyamoto
University of Tsukuba
Department of Risk Engineering
Ibaraki 305-8573, Japan
Members of the Editorial Board (continued)

Dr. Hiroshi Nakajima
Omron Corporation
Core Technology Center
9-1 Kizugawadai
Kizugawa-City, Kyoto, 619-0283, Japan

Professor Michael Ng
Hong Kong Baptist University
Department of Mathematics
Kowloon Tong, Hong Kong (China)

Professor Vesa A. Niskanen
University of Helsinki
Department of Economics & Management
PO Box 27, 00014 Helsinki, Finland

Dr. B.K. Panigrahi
Assistant Professor
Indian Institute of Technology, Delhi
44-Vaishali Apartment, IIT Delhi Campus
Hauz Khas, New Delhi – 110016, India

Professor Xiajiong Shen
Henan University, College of Computer and Information Engineering
85 Minglun Street, Kaifeng, Henan 475001
Hong Kong (China)

Professor Roman Slowinski
Poznan University of Technology
Laboratory of Intelligent Decision Support Systems, Institute of Computing Science
Piotrowo 2, 60-965 Poznan, Poland

Professor Shusaku Tsumoto
Shimane University
Department of Medical Informatics
School of Medicine
89-1 Enya-cho, Izumo-city
Shimane 693-8501, Japan

Professor Xiaofeng Wang
Shanghai Maritime University
College of Information Engineering
1550, Pudong Avenue
Shanghai 200135, P.R. China

Professor Anita Wasilewska
Stony Brook University
Computer Science Department
Stony Brook, NY 11794, USA

Professor Jiali Xia
Jiangxi University of Finance & Economics
UFSoft School of Software
NanChang 330013, Jiangxi, P.R. China

Professor Keming Xie
Taiyuan University of Technology
College of Information Engineering
No.79, West Yingze Street
Taiyuan, Shanxi, 030024, P.R. China

Professor JingTao Yao
University of Regina
Department of Computer Science
Regina, Saskatchewan S4S 0A2, Canada

Professor Dongyi Ye
Fuzhou University
College of Mathematics and Computer Science
Fuzhou 350108, Fujian, P.R. China

Dr. Yanqing Zhang
Georgia State University
Department of Computer Science
P.O. Box 3994
Atlanta GA 30302-3994, USA

Professor Shuigeng Zhou
Fudan University
Department of Computer Science and Engineering, 220 Handan Road
Shanghai 200433, P.R. China

Professor William Zhu
University of Electronic Science and Technology of China
School of Computer Science and Engineering, North Jianshe Road
Chengdu, 610054, SiChuan, P.R. China
CALL FOR PAPERS

International Journal of Innovative Computing and Applications (IJICA)

Website: www.inderscience.com
ISSN (Online): 1751-6498      ISSN (Print): 1751-648X

IJICA proposes and fosters discussion on all new computing paradigms and corresponding applications to solve real-world problems. It will cover all aspects related to evolutionary computation, quantum-inspired computing, swarm-based computing, neuro-computing, DNA computing and fuzzy computing, as well as other new computing paradigms.

Objectives
IJICA provides a lively forum for the communications of original research. Its objective is the timely dissemination of original research work on innovative computing paradigms and their applications.

Readership
The audience of IJICA consists of researchers, academics, graduate students, professionals and system engineers working and interested in learning about new trends in computing paradigms.

Contents
IJICA publishes original peer-reviewed papers, which include research papers, review papers, case studies, book reviews. Special Issues devoted to important topics on innovative computing paradigms and applications will occasionally be published.

Subject Coverage
Suitable topics include, but are not limited to:

Evolutionary Computation
- Evolvable hardware
- Evolutionary algorithms
- Evolution strategies
- Multi-objective evolutionary algorithms
- Applications
- Genetic Programming
- Program encoding
- Variants of genetic programming
- Genetic systems programming
- Quantum Computing
- Quantum algorithms
- Quantum logic
- Nanocomputing
- Complexity
- Quantum cryptography and other applications
- DNA Computing
- Encoding DNA strands
- DNA-based algorithms
- Bio-inspired systems
- Swarm-based Computing
- Swarm-based algorithms
- Memetic algorithms
- Artificial life
- Ant colony optimisations
- Efficient meta-heuristics
- Particle swarm optimisation
- Neuro-Computing
- Neural networks
- Learning methods
- Machine learning
- Stochastic computing
- Software simulation
- Hardware emulation
- Fuzzy Computing
- Fuzzy logic and fuzzy sets
- Fuzzy controllers
- New Trends

Specific Notes for Authors
Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere.

All papers are refereed through a double blind process. A guide for authors, sample copies and other relevant information for submitting papers are available at www.inderscience.com/papers

All papers must be submitted online.
To submit a paper, please go to Online Submission of Papers.
Editor in Chief: Dr. Nadia Nedjah
Preface

Tsau Young Lin

Department of Computer Science,
San Jose State University,
San Jose, California 95192, USA
E-mail: tylin@cs.sjsu.edu

Xiaohua (Tony) Hu

College of Information Science & Technology,
Drexel University,
Philadelphia, PA 19104, USA
E-mail: xh29@drexel.edu

Biographical notes: Tsau Young Lin received his PhD in Mathematics from Yale University and is a Professor of Computer Science at San Jose State University and a Fellow in Berkeley Initiative in Soft Computing, University of California. He is the President of the International Granular Computing Society and Founding President of the International Rough Set Society. He shares the Editor-in-Chief with Xiaohua (Tony) Hu for the International Journal of Granular Computing, Rough Sets and Intelligent Systems. He has/had served on various roles in reputable international journals and conferences. His interests include data/text/web mining, data security and granular/rough/soft computing. He received the Best Contribution Awards from IEEE ICDM01 and International Rough Set Society (2005) and a Pioneer Award from IEEE GrC 2008.

Xiaohua (Tony) Hu is currently an Associate Professor and the Founding Director of the Data Mining and Bioinformatics Lab at the College of Information Science and Technology, one of the best information science schools in USA (ranked as #1 in 1999 and #3 in 2009 in information systems by US News & World Report). He is now also serving as the IEEE Computer Society Bioinformatics and Biomedicine Steering Committee Chair and the IEEE Computer Society Granular Computing Steering Committee Co-Chair. He is a Scientist, Teacher and Entrepreneur. He joined Drexel University in 2002, founded the International Journal of Data Mining and Bioinformatics (SCI indexed) in 2006 and International Journal of Granular Computing, Rough Sets and Intelligent Systems in 2008. Earlier, he worked as a Research Scientist in the world-leading R&D centres such as Nortel Research Center, GTE Labs and HP Labs. In 2001, he founded the DMW Software in Silicon Valley, California. His research ideas have been integrated into many commercial products and applications.
1 Introduction

In this preface, we will give some brief historical notes. We will address it from two views: academic and activities.

The term granular computing (GrC) was proposed by T.Y. Lin in the Fall of 1996 to label his research area (Zadeh, 1998), which was the computable part of Zadeh’s granular mathematics (GrM). The essential idea was outlined in Zadeh (1997). A newer version will appear soon in the Encyclopedia of Complexity and Systems Science, Springer, 2009. However, the concept was much earlier; granularity was explored explicitly in Zadeh (1979). In an obvious way, the most influential words are his informal definition given at the keynote in Zadeh (1996): ‘information granulation involves partitioning a class of objects (points) into granules, with a granule being a clump of objects (points) which are drawn together by indistinguishability, similarity or functionality’.

The last three notions have been abstracted and generalised to ‘constraints or conceptual forces’. Observe that they may not exert uniformly on all objects. So the objects, which are drawn into a granule, may not be homogenous. Each object in the granule has its own role; in general, they cannot be swappable or commutative. Hence, a granule is a tuple, not necessarily a subset. Further, these granules (tuples) may be regrouped into a collection of relations, often with their roles as relational schema.

This observation leads to T.Y. Lin’s (final) formal GrC model in category theory, which is the final stop of his incremental approach (Lin, 1992, 1997, 1998a, 1998b, 1999a, 1999b, 2003, 2006). We will illustrate the idea in two earlier models and one recent model.

The first GrC model, called neighbourhood system (NS), is a pre-GrC concept that was arisen from totally different context, namely, approximate retrieval (Lin, 1989a). This model formalises the ancient intuition, infinitesimals. In this model, each neighbourhood represents a unit of uncertainty.

The third GrC model, called binary neighbourhood system (BNS), was also a pre-GrC concept and arisen about the same time, but from different context, Chinese wall security policy (Lin, 1989b, 2003). In this particular context, a neighbourhood is a unit of knowledge (known information). BNS can also be viewed from another context, namely, Heisenberg uncertainty principle; in this case, a neighbourhood is a unit of uncertainty.

The last one is ninth GrC model that formalises ancient practices, which granulate daily objects into granules, such as human body into head, neck and so forth. This model defines granules, not by a type I fuzzy sets, but by qualitative fuzzy sets, each of which is characterised by a set (neighbourhood in first GrC model) of membership functions.

Next, we turn to the activities. To promote the concept, in 1997, T.Y. Lin organised and chaired the special interest group on GrC within BISC at UC-Berkeley (Berkeley Initiative in Soft Computing at the University of California-Berkeley). In the same year, he also started the RSDMGrC series of conference, which changed to RSFDGrC. Since then, many special sessions were organised, for example, in WCCI (1998, 2002, 2006), NAFIPS (nearly every year since 1999), etc. In 2005, Xiaohua (Tony) Hu joined the forces with T.Y. Lin and held the IEEE International Conference on Granular Computing in China with the help of Qing Liu. This is one of the major milestones in GrC. This series of conferences has been a very successful one. ‘Noble laureates’ are among the keynotes: L. Zadeh (‘EE-Nobel’), S. Ohsuga (Japanese Fifth Generation Computing), S. Smale (‘Math-Nobel’) and R. Karp (‘CS-Nobel’ in BIBM-GrC-WI-IAT Joint Session). Many distinguished scientists and key industry persons, such as D. Quam (IBM’s key person on cloud computing), are among the speakers. More recently, the multinational
GrC Society also has been established. In the light of these successes, we felt it is time to launch an international journal on GrC.

Acknowledgements

Thanks to Professor Xiaohua (Tony) Hu for his successful negotiation for getting Inderscience to publish for us.

References


Assessment of candidate information models for granular computing

Steven A. Demurjian

Department of Computer Science and Engineering,
School of Engineering,
University of Connecticut,
371 Fairfield Way, U-2155, Storrs,
Connecticut, 06269-2155, USA
Fax: 8604864817    E-mail: steve@engr.uconn.edu

Abstract: Granular computing (GrC) is an emerging discipline of information theory that strives to allow reasoning and analysis based on varying levels of information granularity (from fine to coarse). In GrC, the entire information universe can be organised based on many different criteria, allowing the information to be abstracted, aggregated, classified, generalised and so on, based on various characteristics (e.g., data similarity, operational usage, etc.). As a result, GrC relies on information models to describe the universe, the elements of the universe and the composition of each element. In this paper, candidate information models for GrC are explored and assessed, including: the attribute-based data model, the relational data model, the functional data model and the extensible markup language. This includes a description of these models in terms of their concepts, data definition and data manipulation and an analysis of the suitability in support of GrC.

Keywords: granular computing; GrC; data (information) model; attribute-based data model; ABDM; data (information) table; extensible markup language; XML; relational data model; RDM.


Biographical notes: Steven Arthur Demurjian is a Full Professor of Computer Science and Engineering at the University of Connecticut with research interests on secure-software engineering, security for collaborative web portals with biomedical applications and security-web architectures. He has over 130 archival publications, in the following categories: one book, two edited collections, 41 journal articles and book chapters and 87 refereed conference/workshop articles.
1 Introduction

In information theory, one approach to support the reasoning and analysis of information based on varied levels of conceptualisation is the emerging discipline of granular computing (GrC) (Lin, 1997, 2005, 2006; Zadeh, 1998), a term jointly coined by Lin and Zadeh and having a basis in research conducted by Lin on neighbourhoods (e.g., granules) in databases (Lin, 1989b) and computer security (Lin, 1989a). Information granulation is an approach that partitions the entire information universe into granules (Lin, 1998a, 1998b; Zadeh, 1996, 1997). This partitioning can be based on many different criteria: aggregating objects that demonstrate similarity of features or usage by classification (e.g., defining a relation such as a course table in a university application); abstracting away details via generalisation (e.g., creating an object-oriented class person that would be the root of a hierarchy); distinguishing among different characteristics of objects by specialisation (e.g., students or faculty sub-classes of person); summarising the characteristics of similar objects into a coarser conceptualisation (e.g., collecting multiple computer science faculty instances into a single object) and so on. In all of these situations, the information universe is described in its entirety and from that initial characterisation, it is possible to define elements and components of elements.

In support of GrC, one of the key initial considerations is to define the information universe via an appropriate data representation. For GrC, this data representation has most frequently been based on data tables as formally defined for rough sets (Pawlak, 1991), where a data table has many different equivalent nomenclatures, including: a knowledge representation system, an attribute-value system and an information table. For a data table, columns are labelled as attributes (of the objects) and rows are labelled as instances (of the objects). Formally, the data table is characterised as a pair that contains a finite non-empty universe \( U \) and a finite non-empty set of primitive attributes \( A \). Additionally, for each \( a \) in \( A \), there is a non-empty set of values and a function that maps the universe \( U \) to these values. Collectively, these data tables based on rough sets have been used as the basis to define foundational models for GrC (Lin, 2005, 2006) and to achieve privacy protection for medical data by partitioning attributes into identifying, easily known and sensitive categories (Wang et al., 2004).

The main objective of this paper is the assessment of candidate information models for GrC to support the data table (and dependencies that may exist both within a table and across multiple data tables). Specifically, a variety of models that are old and new are examined, including: the attribute-based data model (ABDM) (Hsiao and Harary, 1970), the relational data model (RDM) (Codd, 1970), the functional data model (FDM) (Shipman, 1981) and the extensible markup language (XML, 2008). The ABDM, proposed in early 1970, was touted for its ability to formally define ‘real-world’ records of information, to support their retrieval and has been shown capable of capturing data from relational, hierarchical, network and FDMs (Demurjian and Hsiao, 1987, 1988). The RDM proposed in 1970 along with the current SQL 1992 (SQL2) has evolved into the dominant database model for commercial and open source database systems, with extensions to support object-oriented and other features underway (SQL3). The FDM and its associated Daplex programming language, proposed in 1981, offers unique programming-like features and its functional formal underpinnings can provide a means to reason about data and constraints and moreover, to capture semantics. Finally, the XML has emerged as an information-representation standard format, allowing information to be modelled and more easily exchanged among programs, websites,
databases, etc. All of these models offer different capabilities in support of data tables and GrC.

The remainder of this paper contains two sections and a conclusion. In Section 2, the four information models, ADBM, RDM, FDM, and XML, are examined; the intent is to describe the concepts underlying each model, along with the data definition and manipulation capabilities. Using this as a basis, in Section 3, the suitability of the four information models in support of GrC by considering them against four different criteria are investigated; the objective is to assess the four models in detail to arrive at a thorough understanding of the way that they compare with one another in their support for GrC. Finally, Section 4 offers concluding remarks and summary recommendations regarding the assessment.

2 Candidate data models

In this section, detailed background on the four candidate information models to be assessed for their suitability for GrC are reviewed, namely: the ADBM (Hsiao and Harary, 1970), the RDM (Codd, 1970), the FDM (Shipman, 1981) and the XML (2008). There is a three-fold focus for each information model: describing the model and its concepts, detailing the data definition capabilities and reviewing the data manipulation capabilities. To serve as a common context for the discussion, a university database is utilised as an example. In the university database, faculty are tracked by name, identifier, office phone and department (e.g., computer science, mathematics, English, etc.) and students are tracked by name, identifier, grade point average (GPA) and campus address (e.g., dormitory). For each course in the catalogue, there is a unique course number (e.g., a combination of department and number such as CS123, MATH233, etc.), the course title and description, the department that offers the course and the list of prerequisites for the course. The courses offered are also tracked, by course number/section number pairs (e.g., courses may be offered in multiple sections) and the FacultyID teaching each offering (by faculty identifier) and the term (e.g., semester, quarter, etc.) of the offering (e.g., Fall2007, Spring2008, Summer2008, etc.). Likewise, the students enrolled by the course number/section number pair for each term are tracked.

2.1 The relational data model (RDM)

Relational data is organised into tuples of relations (Codd, 1970). A database is a collection of relations. The attributes of a relation are distinct. The tuples of a relation have the property that no two tuples are identical. In addition, one or more attributes of the relation may be defined as the primary key of the relation; a relation can have multiple candidate keys, one of which is chosen as the primary key. The primary key of the relation is used to uniquely identify the tuples of the relation. To establish dependencies across two or more relations, a foreign key can be defined consisting of one or more attributes of one relation that reference the primary key of another relation. Foreign keys are employed to establish referential integrity constraints across two relations where the foreign key of one relation references the primary key of another relation (with self-reference allowed). Non-primary key attributes are allowed to have null values; this includes a foreign key. A definition of the university database in tabular
Briefly, let us describe the data relationships of this database. First, each student of the Student relation is uniquely identified by a StudentID (primary key), has a Name, a GPA and a CampusAddress. Next, each faculty member of the Faculty relation also has a unique FacultyID (primary key), a Name, an office phone number and a Department affiliation. Third, each course of the Courses relation (catalogue) is uniquely identified by a Course# (primary key), has a Title, a Description and a Department and may have zero or more prerequisite courses (PCourse# – foreign key). Fourth, the OfferedCourses relation tracks the courses offered by sections taught by faculty each term, with these four attributes forming a compound primary key. Fifth, the EnrolledStudents relation tracks the students enrolled in sections of courses for each term, again by a compound primary key. In terms of referential integrity, the foreign key PCourse# self references the relation Course; note that a course without a prerequisite will have a null foreign key. The two create statements are the RDM data definition language, also shown in Figure 1, which illustrate the main concepts for creating database tables in SQL. For Faculty, there are VARCHAR and CHAR types, when null values are not allowed for attributes (Name and FacultyID), the identification of a primary key (FacultyID) and the requirement of unique values (Department). The Courses CREATE TABLE definition illustrates the foreign key definition for PCourse#.

To illustrate the data manipulation language in RDM, consider Figure 2 where sample SQL statements are given for INSERT, DELETE, UPDATE and SELECT, the second SELECT statement representing a join. The INSERT statement creates a faculty member by inserting values. The DELETE statement removes all faculty in the CompSci department. The query is qualified via the WHERE clause which is a Boolean expression to constrain the impact of the query; this is also used in UPDATE and SELECT queries. The UPDATE statement modifies all of the Faculty in CompSci to be in the Math department. The SET statement performs the modification and can be any valid expression. The first SELECT statement retrieves faculty names and departments for
faculty in either the CompSci or Math departments. Finally, the second SELECT retrieves faculty Names and course Titles for all courses offered by the CompSci department.

**Figure 2**  SQL database operations

```
INSERT INTO FACULTY
VALUES ('John Smith', 12121212>, 5551111>, 'CompSci'>);

DELETE FROM Faculty WHERE Department = 'CompSci';

UPDATE Faculty SET Department = 'Math'
WHERE Department = 'CompSci';

SELECT Name, Department
FROM Faculty
WHERE Department = 'CompSci' OR Department = 'Math';

SELECT Name, Title
FROM Faculty, Courses
WHERE Faculty.Department = 'CompSci' AND
Faculty.Department = Courses.Department;
```

2.2 The attribute-based data model (ABDM)

In ABDM, the initial step in defining a database is the specification of a collection of the attributes that are in the database (Hsiao and Harary, 1970). The database records in ADBM consist of sets of attribute-value pairs. An attribute-value pair is a member of the Cartesian product of the attribute name and the value domain of the attribute. As an example, <Department, CompSci> is an attribute-value pair having CompSci as the value for the Department attribute. Database records have the property that for a given set of attribute-value pairs, no two pairs in the set have the same attribute name. In ADBM, data is considered in the following constructs: database, file, record, attribute-value pair (keyword) attribute-value range, record body, directory, directory keyword and non-directory keyword. Informally, a database consists of a collection of files. Each file contains a group of records which are characterised by a unique set of keywords. A record is composed of two parts. The first part is a collection of attribute-value pairs (keywords). The second part of the record, which is optional, is for unformatted textual information and is referred to as the record body. A record contains at most one attribute-value pair for each attribute defined in the database.

Examples of file definition equivalent to the Faculty and Courses relations and create table statements from Figure 1 are given in Figure 3; the first grouping defines the tables – the ABDM data definition language, the second grouping illustrates actual records of attribute-value pairs. For the data definition in ABDM, the types (string, i-value, etc.) are associated with each of the attribute names. For the actual stored data,
the angle brackets, <>, enclose an attribute-value pair, i.e., keyword and value. The curly brackets, {}, include the record body. The record is enclosed in parentheses. The first attribute-value pair of all records of a file, by convention, is the same. In particular, the attribute is File and the value is the file name (akin to a relation name in RDM). Similar records for Student, Courses, CourseOfferings and EnrolledStudents can also be defined; referential integrity is achieved by using the unique record numbers that are given to every instance.

**Figure 3** Attribute-based files for university database

```
(File, Faculty), <Name, string>, <FacultyID, i-value>,
   <Ophone, string>, <Department, string>)

(File, Courses), <Course#, i-value>, <Title, string>,
   <Description, string>, <Department, string>, <PCourse#, i-value>)

(File, Faculty), <Record Number, 123>, <Name, John Smith>,
   <FacultyID, 12121212>, <Ophone, 5551111>, <Department, CompSci>,
   {This is an assistant professor up for tenure in 2012})

(File, Courses), <Record Number, 987>, <Course#, CS100>
   <Title, Introduction>, <Description, An Intro Course>,
   <Department, CompSci>, <PCourse#, Null>)
```

In ABDM, the indexing criteria for a given database are also a critical part of the definition of the database. In particular, for indexing purposes, there are two different types of attribute-value pairs in a record (or a file), with all of the indexing data maintained in the directory of the database. Certain attribute-value pairs of a record (or a file) are called the directory keywords of the record (file), because either the attribute-value pairs or their attribute-value ranges are kept in a directory for identifying the records (files). Those attribute-value pairs which are not kept in the directory are called non-directory keywords. In the example record above, the File, Record Number and FacultyID attributes would be directory keywords, while the other attributes would be non-directory keywords. The identification of database records is by either directory or non-directory keywords. When directory keywords are used, the search space is clearly defined using the indexing criteria. When non-directory keywords are used, the entire file of records must be searched. Note that null values are not allowed for directory attributes.

Data manipulation in ABDM is via the attribute-based data language (ABDL) that supports five primary database operations, INSERT, DELETE, UPDATE, RETRIEVE and RETRIEVE-COMMON (akin to a join). The INSERT request is used to insert a new record into the database. The qualification of an INSERT request is a list of keywords with or without a record body being inserted. DELETE, UPDATE and RETRIEVE have a query (Boolean expression), with UPDATE having a modifier (changes all faculty in the CompSci department to the Math department). The RETRIEVE has a target list (Name, Department) of attributes and values to be returned. RETRIEVE-COMMON merges two files by common attribute values and is a transaction of two retrieve requests that are processed serially. The first retrieve finds the name of all faculty members who are in the CompSci department; the second retrieve finds the title of all courses offered in
the CompSci department. The middle statement links these two result sets on the department, with the Name (from the Faculty file) and the Title (from the Course file) the result set.

Figure 4  ABDL database operations

<table>
<thead>
<tr>
<th>Action</th>
<th>Conditions</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSERT</td>
<td>(&lt;File, Faculty&gt;, &lt;Record Number, 123&gt;, &lt;Name, John Smith&gt;, &lt;FacultyID, 12121212&gt;, &lt;Ophone, 5551111&gt;, &lt;Department, CompSci&gt;, {This is an assistant professor up for tenure in 2012})</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>(File = Faculty) and (Department = CompSci)</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>(File = Faculty) and (Department = CompSci)</td>
<td></td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>((File = Faculty) and ((Department = CompSci) or Department = Math)) (Name, Department)</td>
<td></td>
</tr>
<tr>
<td>COMMON</td>
<td>(Department, Department)</td>
<td></td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>((File = Courses) and (Department = CompSci)) (Title)</td>
<td></td>
</tr>
</tbody>
</table>

2.3  The functional data model (FDM)

An entity in a functional database is always associated with a collection of distinct functions that can be applied to the entity to return either individual data values or one or more objects (Shipman, 1981). The term object is used to refer to the actual data values (values for the functions) for an entity. Thus, an object can be considered as an instantiation (occurrence, in the earlier terminology) of the entity. An object is analogous to a tuple in the RDM or a record in ABDM. The functions of an entity are applied to the entity to return a particular value that is associated with that entity, i.e., to return a portion of the object. There are two types of functions, scalar-valued functions and entity-valued functions. Each type of function may be either single-valued (returning one value) or set-valued (returning zero or more values). Scalar-valued functions return one or more typical database values (i.e., string, integer and float). Entity-valued functions return one or more entity objects as their values.

Additionally, in the FDM, relationships between the entities may be defined that lead to one or more generalisation hierarchies for the database. Generalisation of Smith and Smith (1977) is an abstraction technique that is used to organise commonalities from multiple entities into a single entity (e.g., the name and identifier for students and faculty can be organised in a person entity). Because of the presence of generalisation hierarchies, there are two different categories of entities, entity subtypes and entity supertypes. An entity subtype exists at the inner and leaf nodes of a generalisation hierarchy and inherits all of the characteristics of its ancestors, i.e., inherits all of the
functions of its ancestors. An entity supertype or, more simply, an entity type, is at the root of a generalisation hierarchy and has the property that no two instantiations (occurrences) of the entity type return the same values for all of the entity’s functions. Additionally, one or more single-valued, scalar-valued functions in an entity type may be defined to be the key of the entity. Functional data is organised into generalisation hierarchies of entities. A database is a collection of entities organised into generalisation hierarchies. References among entities are accomplished when a function is entity valued, with self-referencing possible. In Figure 5, the data language definition of the FDM version of the university database using the Daplex language is presented.

**Figure 5** Daplex entities for university database

```sql
DATABASE University IS
  TYPE Person; SUBTYPE Student; SUBTYPE Faculty;
  TYPE Courses; TYPE CoursesOffered;

  TYPE Person IS
    Name : STRING(1..30); Identifier : STRING(1..8);
  END ENTITY;

  TYPE Courses IS
    Course# : STRING(1..8); Title : STRING(1..20);
    Descrip : STRING(1..100); PCourse# : SET OF Courses;
  END ENTITY;

  TYPE CoursesOffered IS
    Course# : STRING(1..8); Section# : INTEGER;
    TERM : STRING(1..10);
  END ENTITY;

  SUBTYPE Student IS Person
    Takes : SET OF CoursesOffered; GPA : FLOAT;
    CampusAddress : STRING(1..100);
  END ENTITY;

  SUBTYPE Faculty IS Person
    Teaches : SET OF CoursesOffered;
    Ophone : STRING(1..7); Department : STRING(1..30);
  END ENTITY;

  UNIQUE Course# WITHIN Courses;
  UNIQUE Section# WITHIN Formats;
  UNIQUE Identifier WITHIN Person;
  UNIQUE Course#, Section#, Term WITHIN CoursesOffered;

END UNIVERSITY;
```
The FDM version of the university database has subtle differences from its RDM counterpart. Note that Student and Faculty in the RDM version have been re-organised into a generalisation hierarchy with Person (root entity type with commonalities) and Student and Faculty (children or entity subtypes). Courses are similar, but the foreign key PCourse# has been replaced by an entity-valued function. The other two differences are: CoursesOffered no longer includes the faculty identifier and has been replaced by a Teaches entity-valued function in Faculty and, EnrolledStudents has been replaced by a Takes entity-valued function in Student. The end of the definition of the database includes the various uniqueness constraints (akin to primary keys).

To illustrate data manipulation in FDM, we concentrate solely on the retrieve operations (FOR EACH), as given in Figure 6. The two examples correspond to their SQL and ABDL counterparts in Figures 2 and 4, respectively; the first FOR EACH is a basic SQL SELECT or ABDL RETRIEVE and the second FOR EACH is an SQL join or ABDL RETRIEVE-COMMON. The first FOR EACH iterates over the Faculty entity to print the Name and Department of faculty members in either CompSci or Math. The second FOR EACH has two nested loops: the outer loop iterates over Faculty entity constrained by the CompSci department, while the inner loop iterates over the Courses entity constrained by the CompSci department. The query prints the faculty Name and course Title if there is a match. UPDATE and DELETE in Daplex works in a similar fashion, with iteration over the entity (single FOR EACH loop) and either DESTROY (remove the entity if the WHERE clause is matched) or modify by an expression.

Figure 6  Daplex database operations

2.4 The extensible markup language (XML)

The XML (2008) has emerged as a standard for information modelling and exchange for web-based applications, database interoperability, common software tool formats, patient record data, etc. XML allows information content to be hierarchically organised and tagged to highlight important/relevant content. The tags capture not only the content, but can be leveraged to represent the meaning of the information (semantics). In a web-based setting, XML is the successor to HTML to allow information content to be hierarchically organised and tagged to highlight important and relevant content. The tags can be
exploited to capture both information content and the meaning of the information (semantics). In addition, XML allows the definition of templates to capture the known structure information for an application by the creation of XML schema files called document type definitions (DTDs). The resulting XML document instances that are created contain both information content (data) as well as semantic notations (tags) that indicate the meaning of the information.

XML provides a flexible means to store and transmit data between different information systems and platforms and has emerged as a dominant means for interoperability on the World Wide Web (as a successor to HTML) and as a standard for information format and exchange (e.g., the OpenDocument project for office applications). Both HTML and XML use tags to identify data/elements. However, while the HTML tags are predefined and specify the way the data within the tags is to be displayed, the XML tags are user-defined and can be employed to identify the data structure in a hierarchical fashion. For instance in HTML, `<b>` or `</b>` are all predefined tags to display the data/characters between them in a bold font. In XML, you will not have those kinds of predefined tags. All of the tags are defined by the users to indicate the structure of the data elements. An XML document for faculty data is given in Figure 7.

Figure 7  Relational tables for university database

```xml
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Faculty>
  <Name> John Smith </Name>
  <FacultyID> 12121212 </FacultyID>
  <Ophone> 5551111 </Ophone>
  <Department> CompSci </Department>
</Faculty>
```

In the XML fragment, the tags `<Faculty>`, `<Name>`, `<FacultyID>`, etc., are defined, based on the application and/or modelling requirements. These tags capture data content (with attribute names) as well as data dependencies; they do not provide any clues on formatting or displaying the data elements between them. In addition to these structural capabilities of an XML file, each of the elements can be further quantified with tags to track semantic aspects of the data. For example, the Name could be augmented with a tag that indicates whether it is a married or maiden name and the Department could be tagged with a title (e.g., professor, associate professor, head, etc.). If the XML structure contained numerical data (e.g., GPA), then the additional tags may denote the scale of the data (e.g., out of 4.0, 100, etc.).

The overall appearance and structure of an XML file can also be defined and validated by an XML schema file. An XML schema file defines the tags or attributes that can be used and where they appear in an XML file. There are two types of XML schema files: the DTD and the XML schema (a more sophisticated construct). A sample XML schema for the XML fragment of Figure 7 is defined in Figure 8; this is analogous to data definition in a database. This XML schema file defines that in the corresponding XML file, the Faculty element is a complex type (akin to a relation in RDM, file in ABDM and entity type in FDM) and must have one Name, one FacultyID, one Ophone and one Department element. Uniqueness is achieved via the ‘unique’ keyword attached to the
FacultyID element. With a XML schema, an XML parser can be used to validate whether the elements within a XML file are valid or not with respect to the XML schema. Conceptually, a XML schema is analogous to type declaration for a RDM relation, an ABDM file or a FDM entity type (subtype).

Data manipulation in XML is achieved via XQuery (2007), a query language for XML documents. XQuery is utilised in XML to write queries that can extract information from XML documents. To illustrate, in Figure 9, we provide various XQuery code fragments; assume that the XML data given in Figure 7 are in XML documents called ‘faculty.xml’ for Faculty and ‘courses.html’ for Courses, to contain, respectively faculty and courses for the university application. The first group of lines in Figure 9 is the XQuery code to access the faculty.xml file for the Names and Departments. The second group of lines in Figure 9 uses the FLWOR language, which is modelled on SQL, to do the same query. The third group is a nested query that accesses both the faculty.xml and courses.xml documents and is equivalent to the join in Figure 2, the RETRIEVE-COMMON in Figure 4 and the nested FOR EACH in Figure 6.

Figure 8  Relational tables for university database

```xml
<?xml version="10" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Faculty">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element unique name="FacultyID" type="xs:string"/>
        <xs:element name="Ophone" type="xs:string"/>
        <xs:element name="Department" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 9  XQuery XML operations

```xml
for $f in doc("faculty.xml")/Faculty/Department [Department = 'CompSci' or Department = 'Math']/Name
where $f.Department = 'CompSci' or $f.Department = 'Math'
return $f/Name $f/Ophone
```

```xml
for $f in doc("faculty.xml")/Faculty/
where $f.Department = 'CompSci' or $f.Department = 'Math'
return $f/Name $f/Ophone
```

```xml
for $c in doc("courses.xml")/Course
where $f.Department = $c.Department
return $c/Course# $c/Department
```
Finally, note that XML has other tools that may be useful for GrC. Extensible Stylesheet Language Transformations (XSLT, 1999), a language for the transformation of XML documents, can take an XML instance and apply a set of rules to translate the format into a new XML document in another format. For GrC, this could be utilised to transform one information model to another; the conversion of data between two different information models, where each information model is represented as a separate XML document with a separate XML schema. Both XSLT and XQuery use the XML Path Language (XPath, 1999) which provides the primitives for their higher level capabilities.

3 Assessing candidate data models

This section, analyses, assesses, compares and contrasts the four candidate information models as given in Section 2 (ABDM, RDM, FDM and XML) for their suitability in support of GrC. To facilitate the comparison, criteria are utilised that are relevant for comparing data models (Demurjian and Hsiao, 1987, 1988), in conjunction with various concepts proposed in Zadeh (1997). The criteria chosen are:

- Formalism and theory: This criterion is used to assess the degree to which a formal model exists for the data model that is capable of supporting GrC.
- Expressive power: This criterion is used to evaluate the ability of each data model to support the definition, usage and analysis of granules.
- Data definition and manipulation: This criterion is used to evaluate the ability of each data model to capture data and meta-data (attribute names, attribute types, value ranges, minimum and maximum values, etc.) and manipulate the information (retrieve, insert, update and delete).
- Extensibility and utility: This criterion is used to determine the usability of the data model, from many different perspectives.

In the remainder of this section, these four criteria are explored against the four models, as a means to assess their suitability for GrC; this is covered in Sections 3.1 to 3.4. To provide a collective view of the assessment, Section 3.5 collates the results of the assessment into tabular form, accompanied by a summary discussion.

3.1 Formalism and theory

All of the data models given in Section 2 have a certain degree of formalism/theory upon which they are based. GrC relies on an information model that has a formal basis, in order to capture the required information within a data table (Pawlak, 1991) and to employ theory of rough sets and fuzzy logic for granular analysis. As a result, it is vital that an underlying data model is rich enough to support the formalisms necessary for GrC. To begin, the ABDM (Hsiao and Harary, 1970) was proposed by David K. Hsiao (founder of ACM Transactions on Database Systems) and Frank Harary (noted graph theorist) and has a formal basis that has been explored by other researchers in the 1970s (Rothnie, 1974), 1980s (Demurjian and Hsiao, 1988) and 1990s (Lin, 1992), with applicability in many different contexts including paging environments for OS, data-model transformations for federated databases and polyinstantiation for security,
Assessment of candidate information models for granular computing

respectively. Likewise, the RDM (Codd, 1970) has a rich history and formal basis using set theoretic concepts and is the dominant approach to date for GrC researchers Demchenko et al. (2005); Lin (2005); Wang et al. (2004).

The FDM (Shipman, 1981) was proposed in an era in the early-to-mid 1980s when semantic data models (successors to the entity-relationship data model) were advocated to break away from relatively flat approaches (e.g., relational, hierarchical and network data models and databases systems) to an approach that was semantically rich and able to model the data as it occurs in the ‘real world’. In fact, in a workshop in 1993 later published as special journal issue (Gray et al., 1999), the overriding theme emphasised the functional model and its potential for a unifying paradigm, providing formalisms based on functional and logic programming that could be used to capture both relational data and object-oriented data in a formal way. Lastly, XML (2008) has emerged as a de facto standard across many different disciplines, being used in a wide variety of contexts (web semantics, patient health records/standards, database interoperability, information exchange format, etc.). XML (2008) is based on detailed specifications and has a grammar in Extended Backus-Naur Form. As a result, there are XML parsers that have been built for many purposes to parse XML files and XML schemas (see Section 2.4 again) in different ways. Such a formal-grammar basis means that XML can be utilised in a very rigorous manner to reason using context-free grammar concepts from automata theory. In summary, it is clear that from a formal basis, each of these four data models can support the information/data table needs of GrC, offering different strengths in their support.

3.2 Expressive power

The expressive power needed to support information granules can be impacted by many factors; we choose three for our comparison:

- **grouping capability** which involves the reason that objects are grouped into granules (equivalence classes) and can be based on object relationships, similarity, proximity, semantics, etc. (Zadeh, 1997)

- **linguistic representation of granules** which assigns a linguistic value to each granule that can be accomplished using a naming convention or by providing sample representative objects (Zadeh, 1997)

- **constraint and type checking** which involves the ability to enforce intra-granular and inter-granular dependencies (constraint checking) while simultaneously insuring that the granule always adheres to the defined structure (type checking).

From an information model perspective, these three granulation factors are directly related to the features and characteristics of the data models as presented in Section 2 and includes: aggregation (records in ABDM, relations in RDM, types and subtypes in FDM and XML schemas in XML), generalisation (inheritance in FDM), identity and uniqueness (directory attributes in ABDM, primary keys in RDM, uniqueness in FDM and unique in XML) and relationships (record references in ABDM, foreign keys in RDM, entity-valued functions in FDM and hierarchical structure in XML).

Specifically, for the grouping capability factor, the assembly of objects into granules (equivalence classes) is aggregation, which is provided by all four data models.
Aggregation is simply a result of the creation of the equivalence class and the aggregation can be based on similarity, proximity, etc. The process of grouping objects into granules is part of GrC, but once the granulation has been completed, then any of these four models have sufficient aggregation to represent a granule. If granules are related to one another hierarchically (or the grouping occurs based on a hierarchical relationship), then either FDM or XML is appropriate for this resultant aggregation. For the linguistic representation of granules factor, again, the same aggregation capability applies; all four models provide a naming convention and the examples as given in Section 2 for ABDM and XML illustrated data instances (sample representative objects).

Finally, for the constraints and type checking factor, each model offers different capabilities. For ABDM, uniqueness of records, record references across objects (instances), support for attribute value pairs (used by many granulation researchers) and directory attributes (akin to keys), all provide the basis for constraint checking; the definition of the attributes (name and type) in each record provides type checking. For RDM, no duplicate tuples per relation, primary keys, foreign keys and referential integrity are relevant for constraint checking; a relational table definition with its attributes and data types provides type checking. For FDM, the types and subtypes provide explicit support for type checking (i.e., these types and subtypes are akin to programming languages types and subtypes in Ada), while uniqueness constraints preserve properties (constraint checking). For XML, the XML schema is a template that an XML instance must follow (type checking) and in its definition, provides the ability for dependencies (constraint checking); as such, for different granules, there would be different XML schemas and XML provides the ability to parse and map from one XML schema to another via XSLT (1999) and XQuery (2007). Thus, XML is much more powerful than the other models, particularly in type checking; the use of XML can facilitate the ability to transform one granule to another while still preserving content and semantics.

### 3.3 Data definition and manipulation

The third criterion involves data definition and data manipulation capabilities, which are considered separately. For data definition, all four of the models provide the definition of a schema. Of the four, RDM and FDM are the most precise in their definitions: in Figure 1, RDM supports primary keys, referential integrity (foreign keys), uniqueness of column values and not null specification and in Figure 5, FDM is very specific in its entity type and subtype definitions, provides uniqueness and allows the definition of enumeration types (not shown). As a result, from this perspective, both are well suited for the definition of information models for GrC. Conversely, ABDM and XML are not as complete in their definition, the actual instance data of these two approaches unites the names with the values: in Figure 3, ABDM has attribute-value pairs (<Name, John Smith>) in each record that is stored in the File and in Figure 7, XML has tags (<Name>John Smith</Name>) that identify each element of the XML schema (Figure 8). This information is directly accessible (comes along with) when retrieve queries are issued. For RDM, the only way to access this meta-data is to perform meta-data SQL queries against the meta-data kept by a relational database system. Given the inexpensive cost of secondary storage, keeping attribute (element) names along with the attribute values is an acceptable overhead for most applications, particularly in the case of GrC.
For data manipulation, all four of the models are very comparable in their capabilities. As seen in Figures 2, 4, 6 and 9 (for RDM, ABDM, FDM and XML, respectively), the retrieve queries were comparable in their structure. RDM and ABDM were declarative in nature, with their use of SELECT and RETRIEVE. FDM was procedural, writing loops that iterative over entity objects. XML via XQuery was in fact a combination of the two – being influenced by SQL (declarative) while having a link to a more procedural approach. The big advantage in this regard for information modelling of GrC when one considers access the data, seems to be in the future potential of XML. In addition to XQuery, XML has XSLT to allow transformation among XML schemas. Suppose that a GrC application had two XML documents: one XML document was based on an XML schema to represent the required information from one perspective and the second XML document was based on a different XML schema to represent the required information for a slightly different perspective. In this situation, XSLT can be used in a number of different ways. First, it is possible to translate back and forth between the two XML documents for the GrC application, allowing information to be easily exchanged. Second, one could define a third XML schema to serve as a common (aggregate) format and translate from the two other documents to this common schema. Finally, the movement of XML towards the semantic web and Web 2.0 has the most potential for GrC, to allow not just static structure of an information model to be stored, but its meaning.

3.4 Extensibility and utility

The final criterion considered is extensibility and utility, which is focused on the usability of the various models, both today and in the future. This can be examined from many different perspectives:

- **Database platform support:** While Section 3.1 considered the theoretical basis for GrC, when one moves from theory to practice in GrC, a seamless transition to the corresponding database management system for a chosen data model would be very useful. In that regard, RDM has the definite advantage, with a wide variety of commercial (Oracle, SQL Server, Informix, etc.) and Open Source (MySQL, PostgreSQL, Ingres, etc.) products available. XML is also a strong player in this regard, since it is a dominant technique for information exchange (data interoperability) among databases, particularly across networked and distributed database solutions.

- **Future potential:** XML has the greatest potential for future usage in GrC as an information model, due to its widespread and growing usage. In fact, there are many open source XML databases that are starting to be released, including: Apache Xindice, Senda XML DBMS, X-Hive/DB, etc. Many of these are in their earliest release stages. For RDM, SQL3, which will include object-oriented extensions, is still being considered from a standards perspective. These object-oriented capabilities are critical to allow granules to be formed in a representation similar to the way that they occur in the ‘real world’, rather than having the granule be flattened into a relational table. Once approved, the relational database vendors will have to implement SQL3 and release new versions. One can easily hypothesise that if XML database systems may make SQL3 obsolete before it even is released.
• Data model translations: The ABDM, as given in Section 2.2, has been shown to be capable of subsuming the features of RDM and FDM (Demurjian and Hsiao, 1988) at both a model and system level. This means that a single model can be used in place of other models without a loss of information.

Overall, the clear leader from a model perspective is ABDM and its ability to subsume RDM and FDM model capabilities. From a practical perspective, while RDM may have the edge in commercial and open source database platforms, the future seems pointed towards XML for sharing and database usage.

3.5 Summary discussion

To more easily assess and discuss the suitability of the four candidate information models in support of GrC, Table 1 summarises the material presented in Sections 3.1 to 3.4. In the table, there are separate groupings for each of the four criteria. To compare, we utilise qualitative values such as strong, very strong, supported, possible, unknown and emerging. For those combinations of criteria (and factors) and model where there was not any clear or evident support, the table entry was left blank.

Table 1 Summary of four models vs. four criteria

<table>
<thead>
<tr>
<th>Model</th>
<th>RDM</th>
<th>ABDM</th>
<th>FDM</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formalism and theory</td>
<td>Very strong</td>
<td>Very strong</td>
<td>Very strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Expressive power</td>
<td>Grouping</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Linguistic</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Checking</td>
<td>Strong</td>
<td>Supported</td>
<td>Very strong</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Available</td>
<td>Very strong</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>Schema</td>
<td>Supported</td>
<td>Strong</td>
<td>Very strong</td>
</tr>
<tr>
<td></td>
<td>Storage</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Current</td>
<td>Supported</td>
<td>Possible</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Future</td>
<td></td>
<td>Possible</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>DB platform</td>
<td>Very strong</td>
<td>Emerging</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Future</td>
<td>Possible</td>
<td>Very strong</td>
<td></td>
</tr>
</tbody>
</table>

To begin, consider formalism and theory (Section 3.1), as given in Table 1. All four models are very strong with established formal underpinnings. ABDM is based on graph theory, RDM utilises set theoretic concepts, FDM has a functional model as its formalism and XML is specified by a context free grammar. Each model offers the level of formalism that is acceptable for GrC in representing granules. The second criteria, expressive power, had three different factors in Section 3.2: grouping capability (aggregation), linguistic representation of granules (naming convention for granules) and constraint/type checking (enforce intra- and inter-granular dependencies). For grouping, all four models offer a degree of aggregation in different manners (Supported in Table 1); it is difficult to differentiate between them or to rank one above the other. For linguistic representation, the ability to create tables in RDM, files in ABDM, entities in FDM and schemas in XML are all named constructs that are appropriate for representing a granule. Finally, for constraint/type checking, there is a clear distinction among the four models.
ABDM is the weakest of the four, supporting constraints to a limited degree through the uniqueness of record numbers (see Figure 3 again). RDM is somewhat stronger, providing a wealth of definition capabilities in schema creation to define relations among tables and constrain values and dependencies (see Figure 1). FDM is very strong, leveraging programming language typing for its schema definition (see Figure 5) and type checking (from Ada) that is enforced at both compile and runtime. XML is also very strong with the enforcement of an XML document against a defined XML schema (see Figure 8) and the ability to define dependencies across documents (and schemas). If given a choice, the reliable standard (RDM) or emerging technology (XML) are the ones most appropriate when considering the second criteria for GrC.

The third criteria, data definition and data manipulation, has similarities with the second criteria. For data definition, at the schema level, RDM, FDM and XML are very strong in their ability to define objects; ABDM has limited support. This mirrors the previous discussion on constraint/type checking for the second criteria. However, when one considers the data itself, ABDM and XML are very strong; both store the name (attribute for ABDM and element for XML) along with the data for every instance in the database; this greatly simplifies the retrieval and processing of information with an acceptable storage overhead for most applications. In terms of GrC, ABDM and XML have an edge in this regard; RDM requires querying of meta-data to retrieve column names, while there are no commercially available FDM database systems (hence, unknown). For data manipulation, at a pure language level, all four models support the ability to query and modify information through their respective languages (current/supported in the table). However, if you look to the future, the edge is to XML with its analysis and parsing tools (XSLT and XPath – see Section 2.4 again) with RDM in the possible category pending the outcome of SQL3 and its acceptance and deployment in database products. This overlaps with the fourth criteria, Extensibility and utility, particularly in regards to the DB platform and future. RDM is very strong as the dominant commercial database management system, with XML emerging; ABDM and FDM are not a factor in this regard. For the future, the same trend is evident, with RDM possible (pending SQL3) and XML very strong (widely accepted standard and emerging database platforms). Lastly, ABDM is the only model shown capable of capturing both relational and functional data; thus, it has the edge in this regard.

The question to ask at the end of this assessment process is what are the available choices for GrC researchers seeking an information model? If we consider the third and fourth criteria, for GrC, the choice for researchers seems to be the established standard (RDM) vs. the emerging technology (XML). This choice is governed by whether the researchers are seeking to not only reason about information granules, but are also interested in transitioning to storage of granules for practical experimentation. Likewise, if we focus on the first and second criteria, all of the models are very comparable in their ability to capture granules and reason about their structure and dependencies; FDM and XML have a slight edge over the other two models. Our intent was not to provide a definitive answer to the question, but in the assessment to offer the issues and considerations to allow an informed decision to be made when choosing an information model for GrC.
4 Conclusions

In this paper, candidate information models and their suitability for GrC have been assessed. This paper considered four different models, presented in Section 2: the ABDM (Hsiao and Harary, 1970), the RDM (Codd, 1970), the FDM (Shipman, 1981) and the XML (2008). These models were explored in detail, with a uniform presentation that included: a description of the concepts and ideas underlying each model; an examination of the data definition capabilities of each model and an exploration of the data manipulation capabilities of each model. Two of these models (RDM and XML) are dominant from a database system perspective today, but all four models have different capabilities to offer when modelling, reasoning and analysing for GrC.

To place the four models in their proper perspective, in Section 3, the models were explored using four different criteria: formalism and theory, expressive power, data definition and manipulation and extensibility and utility: For the first criterion (Section 3.1), each of the four models have a strong formal basis and long history of usage; to varying degrees all are suitable for a formal basis for GrC. For the second criterion (Section 3.2), we compared based on three factors: grouping capability, linguistic representation of granules and constraint and type checking. In the analysis, ABDM, RDM and FDM are very comparable in what they offer, but XML has the potential to transcend all three, with both its current capabilities and its emerging characteristics and features. For the third criterion (Section 3.3), the focus was on data definition and manipulation capabilities. At the data definition level, there were strong ties between ABDM and XML in their pairing of attribute names with values. From a data manipulation perspective, with a focus on querying, XML had an edge with its declarative and procedural language. Finally, for the fourth criterion (Section 3.4), there were different strengths for different models: for ADBM, its strength was its ability to subsume RDM and FDM at the data model level (Demurjian and Hsiao, 1988); for RDM, its strength was in the deployed database platforms (commercial and open source) and the future potential of SQL3 with object-oriented capabilities (which will improve granule representation) and for XML, the usage in relational databases for information exchange, coupled with the arrival of XML database systems, may have the greatest potential for the future.

This leaves us with three recommendations that are the outcome of our assessment of the four data models. First, from a theory perspective, while all four data models are appropriate to some degree for GrC, ABDM and RDM have to be strongly considered due to their long history of usage and formal basis, with FDM an alternative when there is a desire to have a functional/logic based basis to GrC. Second, from a practical perspective, if one is to transition from GrC theory to actual usage, RDM is the choice today, but XML has an excellent chance to be the choice of the future. Third, if one is concerned with capturing semantics (meaning) in addition to structure, then XML is the clear choice; Web 2.0 and the semantic web offer strong potential to satisfy syntactic modelling needs augmented with rich semantics to capture the meaning of the stored information.
References


Granular computing I: the concept of granulation and its formal model

Tsau Young Lin

Department of Computer Science,
San Jose State University,
San Jose, California 95192, USA
E-mail: tylin@cs.sjsu.edu

Abstract: The term granular computing (GrC) was coined by Lin and Zadeh (derived from Zadeh’s granular mathematics) in Fall, 1996. Since then, nine GrC models have been proposed. In GrC2008 keynote, the category theory based model (8th GrC model) was proposed to be ‘the (final) GrC model’; however, the announcement was not in time to be included in the proceedings. So this paper is the first printed presentation.

This paper explains the concept of granulation in three ways: it is defined

1. inductively by classical example
2. informally by Zadeh
3. formally by the category-based GrC model.

It outlines how this model can be specified into nine models by specifying the abstract category to the category of

1. sets/type I fuzzy sets
2. functions/generalised functions such as, random variables (measurable functions), measures/probabilities
3. turing machines/clusters of computers
4. qualitative fuzzy sets.

The concept of qualitative fuzzy sets is new and quite novel; we have explained it via neighbourhood systems.

Keywords: granular computing; GrC; category theory; neighbourhood system; Heisenberg uncertainty principle; discretionary access control; social network; cloud computing; simplicial complex; binary relation; infinitesimal.


Biographical notes: Tsau Young (T.Y.) Lin received his PhD in Mathematics from Yale University. He is a Professor of Computer Science at San Jose State University and a fellow in Berkeley Initiative in Soft Computing, University of California. He is the President of International Granular Computing Society and the Founding President of International Rough Set Society. He shares the Editor-in-Chief with Tony Xiaohua Hu for the International Journal of Granular Computing, Rough Sets and Intelligent Systems. He has served on various roles in reputable international journals and conferences. His interests include data/text/web mining, data security and granular/rough/soft computing. He received the best contribution awards from ICDM01 and International Rough Set Society (2005) and a pioneer award from GrC 2008.
1 Introduction

Granulation seems to be a natural methodology deeply rooted in human mind. Daily ‘objects’ are routinely granulated into sub ‘objects’. For example, human body and the surface of earth are often granulated into sub ‘objects’, such as head, neck, ... and plateaus, hills, plains ... respectively. The boundaries of these sub ‘objects’ are intrinsically fuzzy, vague and imprecise. Flawlessly formalising such a concept has been difficult. Early mathematicians have idealised/simplified granulation into partitions (= equivalence relations) and have developed the partition into a fundamental concept in mathematics, for example, congruence in Euclidean geometry, quotient structures (groups, rings, etc.) in algebra, the concept of ‘a.e.’ (almost every where) in analysis. Nevertheless, the notion of partitions, which absolutely does not permit any overlapping among its granules, seems to be too restrictive for real world problems. Even in natural science, classification does permit small degree of overlapping; there are beings that are both appropriate subjects of zoology and botany. So a more general theory, namely, granular computing (GrC) is needed:

- What is GrC?

It has been a shifting paradigm, since the inception of the idea. Lately, however, the concept seems to have reached its steady state. The goal of this paper is to present this ‘final’ formal model; of course, nothing can be really final.

To trace the evolution of the idea, let us recall how the term was coined. In the academic year 1996–1997, when I arrived at UC-Berkeley for my sabbatical leave, Zadeh suggested granular mathematics (GrM) to be the area of research. To narrow down the scope, I proposed the term GrC to label the area (Zadeh, 1998). Therefore, at the very beginning, GrC is the computable part of the GrM.

- What is GrM?

In his paper, Zadeh (1979) already had implicitly explained his view. In 1997, he outlined his main idea in the seminal paper (Zadeh, 1997), where he said, ‘Basically, TFIG… inspiration… humans…… its foundation and methodology are mathematical in nature’.

Our views are the same, but I adopted an incremental approach: by mapping neighbourhood system (NS) to Zadeh’s intuitive definition (Zadeh, 1996), I took NS as the GrM and regarded it as the first GrC model (Lin, 1998a, 1998b, 1999).

- What is NS?

Totally from different context, namely, approximate retrieval, in 1988–1989, I generalised the notion of topological neighbourhood systems (TNS) to that of NSs by simply dropping the axioms of topology (Bairamian, 1989; Lin, 1988, 1989a; 1989b; 1992; Lin et al., 1990; Chu and Chen, 1992). In this view, each neighbourhood is a unit of uncertainty, see Section 10. But in later part of the same year (1989), when I considered the Chinese wall computer security policy (CWSP) model, a neighbourhood was regarded as a unit of information (known knowledge) (Lin, 1989a); see Lin (2003, 2009) for recent results. In other words, in early GrC period, a granule is a unit of knowledge or lack of knowledge (uncertainty).

Since then, nine working models have been built and it seems that we have reached the steady state. So one of them (8th GrC model) is selected to be the ‘final’ model.
This paper ‘defines’ the concept of granulation in three ways:

1. ‘inductive’ definition: granulation is defined by a set of examples
2. Zadeh’s informal definition
3. the ‘final’ formal model.

This ‘final’ model is a category theory based model (8th GrC model) that fits Zadeh’s intuition and can be specified to those nine models and classical examples.

The paper is roughly organised as follows: after the introduction, we present a soft introduction to the delicate nature of GrC. Then a set of ‘classical’ examples is presented as ‘inductive definition’ of GrC. The main part, which is the formal ‘final’ definition of GrC, is then presented. In this section, we have included Zadeh’s intuitive definition. Rest of the paper is devoted to relate this final model to nine models and ‘classical’ examples. Much more precise details will be in the forth coming papers.

2 Some delicate nature of GrC

In common practices, very often, we loosely regard any collection of objects as a set. Strictly speaking, this loose view needs some qualification. It actually implicitly implies that the interactions among objects are formally ignored. Recall that in set theory, all points are discrete, namely, there are no interactions among elements. However in GrC, these interactions are the main concerns. So the collection of granules, called granular structure (GrS), should never be regarded as a set of granules unless granules are mutually disjoint. In fact, there are three states for GrS: a granule can be considered

1. in isolation, called isolated state or internal state because in this case only the internal structure is relevant
2. as a sub-object of the universe, called embedded state or conceptual state because the full concept of granule is under consideration
3. as a point in the collection of granules called quotient state or external state because the only concerns are the interactions among granules.

Now, let us recall some practices in mathematics.

Example 1: Examples from algebraic geometry: Let \( Z \) be the ring of integers, that is, we are considering not only the set of integers but also its two operations, addition and multiplication. A prime ideal is a subset that is closed under addition and multiplication by any element of the ring. Let \( p \) be a prime number, then the prime ideal generated by \( p \) is a subset

\[ \{ ..., -2p, -p, 0, p, 2p, ... \}, \]

which is obviously closed under addition and multiplication by any integer. In algebraic geometry, the collection of prime ideals is often denoted by \( \text{Spec}(Z) \). If every prime ideal is regarded as a point, then, \( \text{Spec}(Z) \) is a set, however, if the interactions among these prime ideals are considered, \( \text{Spec} (Z) \) is a topological space under Zariski topology.
This topological space is the quotient structure of GrC, the collection of granules in quotient state.

Here are some more ‘elementary’ examples.

**Example 2**: Let \( U = \{e_0, e_1, e_2, e_3\} \) be a finite set.

1. From partition theory: Let \( \beta \) be the collection \( \{\{e_0, e_1\}, \{e_2, e_3\}\} \) of subsets. As all subsets are mutually disjoints, \( \beta \) is an honest classical set.

Next, we consider a very common example, which often has not been carefully considered. Let \( \beta \) be the collection of all subsets of \( U \). In this case, there are interactions among these subsets as there are overlapping among them. Let us examine how \( \beta \) has been handled.

2. In set theory: For casual users, \( \beta \) is often regarded as a set and is called the power set. This is valid only if we disregard the interactions among these subsets.

However,

3. In algebra: \( \beta \) is a Boolean algebra, when we consider the interactions in terms of ‘intersection’ and ‘union’.

4. In lattice theory: \( \beta \) is a lattice, if we do consider the ‘union’, ‘intersection’ and ‘inclusion’ together.

5. In algebraic topology: If we do consider the interactions in terms of ‘inclusion’ only, then \( \beta \) is a partial ordered set and has a nice geometrical representation: We take \( U = \{e_0, e_1, e_2, e_3\} \) to be a set of linearly independent points in a Euclidean space; then \( \beta \) can be interpreted as a simplicial complex as follows:

   1. \( U = \{e_0, e_1, e_2, e_3\} \) is a collection of vertices
   2. \( \beta \) is a collection of simplexes
      a. four singletons are the four zero-simplexes: \( \Delta_i = \{e_i\} \), \( i = 0, 1, 2, 3 \)
      b. six subsets of two elements are the six one-simplexes: \( \Delta_{ij} = \{e_i, e_j\} \), \( i < j = 0, 1, 2, 3 \)
      c. four subsets of three elements are the four two-simplexes: \( \Delta_{ijk} = \{e_i, e_j, e_k\} \), \( i < j < k = 0, 1, 2, 3 \)
      d. one subset of four elements is the three-simplexes

This simplicial complex is called the closed tetrahedron.

In this section, we had presented four mathematical GrS structures for the collection of all subsets (‘power set’).

### 3 Inductive definition of granulation

The following examples collectively define ‘inductively’ the concept of granulation.
3.1 Commutative granules

E1 Ancient practice: granulation of daily objects

Many daily objects are routinely granulated into sub ‘objects’. For example, human body is granulated into head, neck, …; the surface of earth has been granulated into hills, plateaus, planes, … This class of examples are intrinsically fuzzy, vague and imprecise, more precisely, on the boundaries of granules. There are easy solutions, but not adequate: one can easily write down a membership function to represent a granule, such as head, neck or body. However, each expert may come up a distinct membership function and therefore may have a distinct theory of GrC on human body. This is not a satisfactory theory, as there is no unified view on all of these distinct expert dependent theories.

In this paper, a new qualitative fuzzy set theory is used to model this class of examples; they are referred to as 9th GrC model; more details will be in forthcoming papers.

E2 Ancient mathematics: intuitive granulation of the space and time

The space and time has been granulated into granules of infinitesimals by early scientists. Of course, mathematically, the notion of infinitesimal granules does not really exist. Nevertheless, it has noisily played a very important role in the history of mathematics. This intuitive notion led to the invention of calculus by Newton and Leibniz. Actually the idea was much more ancient; it was in the mind of Archimedes, Zeno, etc. Yet, the solutions were in modern time. It led to the theory of limit (18th century), topology [early 20th century (Munkres, 2000)] and non-standard analysis, which formally realised the original intuition [mid-20th century (Robinson, 1966)].

The modern theories of this ancient intuition have inspired two models, first GrC model and second GrC model; they have been referred to as NSs and partial coverings (PCovs) respectively in pre-GrC time.

E3 Classical case: partition (equivalence relation)

This class of examples have been well studied in mathematics and recently in computer science by rough set community.

E4 Granules of uncertainty from quantum mechanics

Heisenberg uncertainty principle states that, in general, neither the momentum nor the position of a particle can be determined simultaneously with arbitrary great precision. In other words, a great precision of momentum can determine only a probabilistic neighbourhood of positions and vice versa.

E5 Granules of knowledge from computer security

In many computer systems, discretionary access control (DAC) model assigns each user a set of users (friends) who can access his files. However, we also consider a set of users (foe), who cannot access his files. This is called explicitly denied list in military security. Examples [E4] and [E5] are real world examples. The idea has been simplified into 3rd GrC model. It was called binary NS in pre-GrC terminology. Mathematically, it is
equivalent to a binary relation (BR). Geometrically, a BR is a graph or network. They are the major data structures in computer science. The example [E4] is also in the category of random variables; we are expecting to see it playing heavy roles in future papers.

3.2 Non-commutative granules

Next, we give examples of non-commutative granules (not necessarily commutative), which are generalisations of PCovs and BRs (second/third GrC model).

E6 Granules of knowledge in data mining

One of the important concepts in data mining (association rules) is the frequent itemset. It has two views:

1. A frequent itemsets is a collection of constant sub-tuples in a given relation.
2. A relational table (which is called an information table in rough set community) can be viewed as a knowledge representation of a universe which is a set of entities. In this view, each attribute (column) defines an equivalence relation on the set of entities; this was observed by Z. Pawlak in late 1982 and Tony Lee in early 1983 (Lee, 1983; Pawlak, 1991). An attribute value can be regarded as the name of such an equivalence class. In this view, a frequent itemset is an intersection of equivalence classes. This intersection is a granule (Lin, 2000; Louie and Lin, 2000).

E7 Granules of knowledge in web/text mining

High frequent sets of cooccurring keywords (fsck) in document set or the web can be regarded as an abstract ordered simplex. Moreover, the apriori principle of frequent item sets turns out to be the closed condition of simplicial complex (Spanier, 1966). Recall that a simplicial complex (of keywords) consists of two objects: one is a finite set of vertices (keywords). Another one is a family of ordered/oriented/unordered subsets, called ordered/oriented/unordered simplexes that satisfy the closed condition, namely, any subset of a simplex is a simplex. This is an important mathematical structure in algebraic/combinatorial topology. Currently, it is finding its way to web technology (Lin and Chiang, 2005; Lin et al., 2006; Lin and Hsu, 2008).

E8 Granules of knowledge from social networks

The collection of committees in a human society (a set of human beings) is a granulation. Observe that each member may play distinct role in a committee, so the members cannot exchange their roles freely. We can view the collection of roles as a relational schema. If we do so, then a committee is a tuple. Observe that different types of committees have different schema. The set of committees under the same schema forms a relation. A collection of $n$-ary relations for various $n$ is a granulation of the society.

The example [E6], [E7] and [E8] are modelled into fifth GrC model.
3.3 Granules of advanced objects

Roughly, the examples [E1] to [E8] given above are granulation of data. Now, we will turn to more complicated objects.

First, we observe that all examples can be fuzzified (type I fuzzy set), which can be fully characterised by a membership function; mathematically, they are bounded functions. So functions can be granules. Moreover, the example [E4] in Heisenbeerg uncertainty principle is actually a granulation of random variables (measurable functions). Hence, it should be nature to extend the consideration of membership functions and measurable functions to general functions. So we include the following example into our collections.

**E9 Functional granulation: the family of radial-basis-functions is a granulation in some function space (Park and Sandberg, 1991)**

More generally, a collection of functions (e.g., radial basis functions) that satisfies the universal approximation property will be regarded as functional granulation. In fact, we will extend it to measures/probabilities and generalised functions.

This class of examples led to sixth GrC model.

**E10 Computers or clusters of computers in grid/cloud computing are granules**

These are hardware examples.

Traditionally, ‘how to solve it’ (Polya, 1957) has not been any part of formal mathematics, however, ‘how to compute’ is an integral part of computing. So GrC includes some mathematical/computational practices.

**E11 A granule can be a subprogram in a program or a lemma in a mathematical proof, when the proof is computable**

Formally, within the computable domains, such a granule is a sub-Turing machine. The so called ‘divide and conquer’ in computer science is actually a granulation of Turing machines. This idea is modelled in 7th GrC model.

A lemma in a mathematical proof is a granule conceptually; however, we should not include it in GrC unless the mathematical proof itself is computable. However, in this class, we may extend the idea of 7th GrC model to GrM.

Let us summarise this section by Zadeh’s intuitive definition.

- ‘information granulation involves partitioning a class of objects (points) into granules, with a granule being a clump of objects (points) which are drawn together by indistinguishability, similarity or functionality’ (Zadeh, 1996; Lin, 1998a).

We will paraphrase into

**Definition 1: Informal definition**

- ‘information granulation involves partitioning a class of objects (points) into granules, with a granule being a clump of objects (points) which are drawn together by some constraints or forces, such as ‘indistinguishability, similarity or functionality’.”
4 Formal definition of granulation

In GrC2008, I proposed to use the category theory based model (eighth GrC model) as the ‘final’ formal model for GrC. To state it, it requires some category theory.

As the category is not a ‘common sense’ in computer science, we will introduce some simpler models as stepping stones.

4.1 Some simpler models

By interpreting Zadeh’s clump of objects as a set we formally define

**Definition 2:** Second GrC model. Let \( U \) be a classical set, called the universe. Let \( \beta = \{ F_k | k \in K \} \) be a family of subsets. Then the pair \((U; \beta)\), is called the 2nd GrC model. If we use pre-GrC terminology, \( \beta \) should be called a partial covering (PCov). However, most of those papers addressed only on (full) covering. PCov is a special case of NS.

In this model, we have implicitly assumed that the constraints or forces (in Zadeh’s definition) are uniformly exerted on each member of the granule. So a granule is a set.

If the constraints or forces are not uniform, one can regard them as a schema (of a relational database). In this case granules are tuples of some relations. These are modelled in 5th GrC model.

Let us introduced a convention:

- **Convention for index.** In computer science, the index often runs through a countable set. In this case, we often denote it as follows: \( k = 1, 2, \ldots, k, \ldots \) without naming an index set. As this paper will include GrM, we will take the following convention: the lower case letter, say \( k \), denotes the parameter that runs though an index set, \( K \), whose name is the corresponding cap letter.

To make the understanding of the 5th GrC model easier, we explain the generalisation process as follows:

1. \( U \) is generalised to \( U = \{ U_j^h | h \in H; j \in J \} \)
2. \( \ldots \ldots \ldots \)
3. \( F^k \subseteq U \) is generalised to \( R^k \subseteq \prod \{ U_{L_k}^h | k \in H, L_k \subseteq J \} \). By convention, \( k \) is a fixed constant in \( H \) and \( l_k \) is a parameter run through \( L_k \), which is a subbag of \( J \).
4. \( \ldots \ldots \ldots \)

**5th GrC model**

1. Let \( U = \{ U_{j_h}^h | h \in H, j_h \in J_h \} \) be a given family of classical sets, called the universe.

Note that distinct indices do not imply the sets are distinct. More precisely, it is a re-indexing (allowing repetitions) on a family of sets.
2 Let \( \prod \{ U^k_i | k \in H; L_k \subseteq J_k \} \) (\( k \) is a fixed constant) be a family of Cartesian products of various lengths \( n \), where \( n = |L_n| \) is the cardinal number of \( L_n \) that may be infinite. Here ‘\( \subseteq \)’ means a sub-bag. Recall that a bag is a set that allows the repetition of some elements (Garcia-Molina et al., 2002).

3 Recall that an \( m \)-ary relation is a subset \( R^m \) of a product space in the previous item.

4 Let \( \beta = \{ R^m | m \in M \} \) be a given family of \( n \)-ary relations given in previous item for various \( n \); note that \( n \) can be infinite.

Then the pair \( (\mathcal{U}, \beta) \) is the formal definition of 5th GrC model. We may call it relational GrC model.

This form of 5th GrC model is more convenient for the generalisation. However, for applications, we will give another form in Section 7 below.

### 4.2 The category theory

First, we would like to observe that 8th GrC model is abstractly the same as the category of relational databases (Lin, 1990). In other words, from the point of views of mathematical structures the category of data and knowledge are the same. However, their meanings are very different.

Let us set up some language for category theory.

**Definition 3:** A category consists of

1. A class of objects
2. For every ordered pair of objects \( X \) and \( Y \), a set \( \text{Mor}(X, Y) \) of morphisms with domain \( X \) and range \( Y \); if \( f \in \text{Mor}(X, Y) \), we write \( f : X \to Y \)
3. For every ordered triple of objects, \( X, Y \) and \( Z \), a function associating to a pair of morphism \( f : X \to Y \) and \( g : Y \to Z \) their composite \( g \circ f : X \to Z \). These satisfy the following two axioms
   a. **Associativity.** If \( f : X \to Y \), \( g : Y \to Z \) and \( h : Z \to W \), then \( h \circ (g \circ f) = (h \circ g) \circ h : X \to Z \)
   b. **Identity.** For every object \( Y \) there is a morphism \( I_Y : Y \to Y \) such that if \( f : X \to Y \), then \( I_Y \circ f = f \) and if \( h : Y \to Z \), then \( h \circ I_Y = h \).

If the class of objects is a set, the category is said to be small. Here are some examples.

1. The category of sets: the objects are classical sets. The morphisms are the maps.
2. The category of fuzzy sets: the objects are fuzzy sets (of type I). The morphisms are the maps.
3 The category of crisp/fuzzy sets with crisp/fuzzy BRs as morphisms: the objects are classical sets. The morphisms are crisp/fuzzy BRs. In crisp case, this is the category of entity relationships models.

4 The category of power sets: the object $U_X$ is the power set $P(X)$ of a classical set $X$. Let $U_Y$ be another object, where $Y$ is another classical set. The morphisms are the maps, $P(f): U_X \to U_Y$ that are induced by maps $f: X \to Y$.

5 The category of topological spaces: the objects are classical topological spaces. The morphisms are the continuous maps.

6 The category of neighbourhood system spaces (NS-space): the objects are NS-spaces; see first GrC model. The morphisms are the continuous maps.

4.3 The ‘final’ formal GrC model

Let CAT be a given category; we adopt the index convention stated above. By replacing the classical sets in 5th GrC model with objects in CAT, we have the ‘final’ model.

Definition 4: Category theory based GrC model

1 $\mathcal{C} = \{ C^h_j | h \in H, j_h \in J_h \}$, called the universe, be a family of objects in the category CAT.

2 There are Cartesian products $\prod \{ C^k_i | k \in H, L_k \subseteq J_k \}$ of bags of objects, where $k$ is a fixed constant in $H$ and $l_k$ is a parameter run through the index set $L_k$.

3 An $n$-ary relation object $R^n$ is a sub-object of a product object.

4 $\beta$ be a family of $n$-ary relations ($n$ is any cardinal number and could vary).

The pair $(\mathcal{C}, \beta)$ is 8th GrC model. This is the ‘final’ formal GrC model. We may refer to it as the category theory based GrC model [see Lin (2009) for softer treatments].

5 An overview of GrC models

By specifying the general category to various special cases, we have all the models. We give a summary here:

Proposition 1: The 8th GrC can be specified to nine models as follows:

1 models of non-commutative granules
   a by taking CAT to be the category of qualitative fuzzy sets, we have 9th GrC model
   b by taking CAT to be the category of Turing machines, we have 7th GrC model
   c by taking CAT to be the category of fuzzy sets (membership functions), functions, random variables and generalised functions, we have 6th GrC model
Granular computing I: the concept of granulation and its formal model

31

d by specifying the category to be the category of sets, we have 5th GrC model

2 models of commutative granules

a by limiting the product objects to be product of two objects in 5th GrC model, we have 4th GrC model

b by limiting the number of BRs in 4th GrC model to be one, we have 3rd GrC model

c by requiring all $n$-nary relations to be symmetric, we have 2nd GrC model

d the reduction to 1st GrC model is treated in Section 10.

Schematically, we summarise the relationships of GrC models as follows: (the diagram will be different, if it is based on their approximation spaces)

- ‘$\Rightarrow\iff$’ is a two way generalisation but they are not inverse to each other
- ‘$\Rightarrow, \uparrow$ and $\downarrow$’ are one way generalisations

‘GM’ means GrC models and rough set theory (RST) means rough set model.

\[
\begin{align*}
8th \ GM & \Rightarrow \begin{bmatrix} 5th \ GM \Rightarrow \ 2nd \ GM \\ 4th \ GM \subseteq \ 1st \ GM \\ 3rd \ GM \iff \ 3rd \ GM \end{bmatrix} \Rightarrow RST \\
8th \ GM & \Rightarrow \begin{bmatrix} 6th \ GM \\ 7th \ GM \\ 9th \ GM \end{bmatrix}
\end{align*}
\]

6 Reduction to advanced GrC models and examples

For next two models, we will use the language of category theory. Note that we have not committed ourselves to every specific detail yet.

Definition 5: Sixth GrC model is in the categories of functions, random variables and even generalised functions

Type I fuzzy sets can be described by membership functions (Zimmerman, 1991) so a granule is a membership function. Hence, we generalise granules to be real/complex valued functions, random variables (measurable functions) and generalised functions (e.g., Dirac delta functions).

In the case, a granule is a function, we may require that the GrS (a collection of functions) has the universal approximation property, namely, any function in the universe can be approximated by the functions in the collections. The membership functions selected in fuzzy controls do have such properties. In neural networks, the functions generated by the activation functions also have such property (Park and Sandberg, 1991). In the case of probability/measure theory, quantum mechanics may be a good guiding example.
Observe that a tuple can be regarded as a mapping \( f: i \to X_i \) where \( X_i, i \in I \) are classical sets (Halmos, 1950). If \( X_i \) are all real/complex numbers, a tuple is reduced to a function. So 6th GrC model is, in the case of functions, a special case of 5th GrC model.

**Definition 6:** Seventh GrC model is in the category of Turing machines

For examples, a collection of lemmas in mathematical proof (mechanisable), a set of subprograms in a computer program or a computer or cluster of computers in grid/cloud computing are granules in the model. In conclusion, we can see [E9], [E10], [E11] are examples of the ‘final’ formal model.

7 Alternate 5th GrC model

The GrS of 5th GrC model (defined in Section 4.1) is a collection of \( n \)-relations for various \( n \). We may also look at GrS as a collection of \( n \)-tuples. As we have observed previously that the collection of \( n \) objects that are ‘drawn together’ is, not necessary a subset, For example, if the universe is a human society then a group of people may be drawn into a committee with distinct roles, such as the chair, vice chair, secretary, treasurer, etc. As every member has different role, they can not be swapped around freely. So the committee is not a set; it is a group of objects, in which each object has different characteristic This set of characteristics can be regarded as a schema (in the sense of relational database) that consists of distinct roles (interpreted as attributes); this is [E8] and is an excellent example of this view. So we formally define 5th GrC as follows:

**Definition 7:** Two forms of 5th GrC model

1. Let \( \mathcal{U} = \{ U^h | h \in H, j_h \in J_h \} \) be a given family of classical sets, called the universe. Note that distinct indices do not imply the sets are distinct.

2. Let \( \prod \{ U^k | k \in K; L_k \subseteq J_k \} \) (\( k \) is a fixed constant in \( H \) and \( L_k \) is a parameter run through \( L_k \)) be a family of Cartesian products of various lengths \( n \), where \( n = |L_k| \) is the cardinal number of \( L_k \) that may be infinite. Here ‘\( \subseteq \)’ means a sub-bag. Recall that a bag is a set that allows the repetition of some elements (Garcia-Molina et al., 2002).

3. Recall that an \( m \)-ary relation is a subset \( R^m \) of a product space in the previous item.

4. Let \( \beta = \bigcup_{m \in M} \{ R^m \} \) or \( \beta = \{ R^m | m \in M \} \) is a set \( n \)-tuples for various \( n \) or a set of \( n \)-relations; note that \( n \) is not fixed and can be infinite.

Then the pair \( (\mathcal{U}, \beta) \) is the formal definition (with two possible interpretations of \( \beta \)) of the ‘new’ 5th GrC model. The term 5th GrC model will represent either meaning. So we may still call it relational GrC model.
Note that the GrS of 5th GrC model is the collection of tuples in a relational database (= a set of relations), if the sets $U^h_{h^*}$ are attribute domains [and the column is at most countable (Ginsburg and Hull, 1981)]. Observe that a high frequent itemset [E6] is a set of constant subtuples with high cardinality. The collections of such frequent itemsets are the union of sets of constant subtuples; this collection is the GrS of a 5th GrC model in this new form. Also note that if $n$ only varies through finite cardinal number, 5th GrC model (defined previously) is the relational structure (without functions) of the first order logic.

Observe that in text analysis, one often considers the set of keywords and 'forget' the original order of the texts (Lin et al., 2006). In Lin and Hsu (2008), we do consider the order of keyword sets (as ordered simplexes). In this view the ordered simplicial complex of keywords [E7] is an example of 5th GrC model. We are expecting swift development of 5th GrC model, as SQL language and technology of database becomes available tools.

8 2nd GrC models and modern examples

Let us assume that the universe is a single set and all products are products of this single set. Further, all relations are assumed to be symmetric. In this case, all tuples are subsets of the universe. Under such assumptions, the ‘final’ formal model is reduced to:

**Definition 8:** 2nd GrC Model is the pair $(U, \beta)$, where $\beta$ is a family of subsets of $U$. It may be referred to as global GrC model. The $\beta$, sometimes is referred to as a PCov.

Obviously, this model includes the classical case, partitions [E3] as its examples. Note that 2nd GrC model is a special case of 1st GrC model (will be defined below). If we regard the sub-collection of all members of the PCov $\beta$, that contains $p$, as a NS at $p$, then this 2nd GrC model is an example of 1st GrC model.

The modern example [E7], unordered simplicial complexes, is an important example of such a model: A simplicial complex consists of a set of vertices and a family of subsets, called unordered simplexes, that satisfies the closed condition (Spanier, 1966).

[Digression] Perhaps, it is worthwhile to note that

- the closed condition of simplicial complex is the apriori principle in association (rules) mining.

This observation plays an important role in document clustering (Lin et al., 2006).

9 Third and fourth GrC models and modern examples

In this section, we will show that 3rd and 4th GrC models formalise modern example [E4] and [E5]. Recall that [E4] concludes that, a precise measure of the momentum can only determine a (probabilistic) ‘neighbourhood’ of positions and [E5] concludes that in computer security, the DAC model assigns to each user $p$ a family of users, $Y_i$, $i = 1, …,$ who can access $p$’s data. In other words, each $p$ is assigned a granule of friends.
Let $U$ and $V$ be two classical sets. Each $p \in V$ is assigned a subset, $B(p)$, of ‘basic knowledge’ (a set of friends or a ‘neighbourhood’ of positions).

$$p \rightarrow B(p) = \{Y_i, i = 1, \ldots\} \subseteq U$$

Such a set $B(p)$ is called a (right) binary neighbourhood and the collection $\{B(p) \mid \forall p \in V\}$ is called the binary neighbourhood system (BNS).

**Definition 9:** 3rd GrC model is the three-tuple $(U, V, \beta)$, where $\beta$ is a BNS. It may be referred to as a binary GrC model. If $U = V$, then the three-tuple is reduced to a pair $(U, \beta)$.

Observe that BNS is equivalent to a BR:

$$BR = \{(p, Y) \mid Y \in B(p) \text{ and } p \in V\}.$$  

Conversely, a BR defines a (right) BNS as follows:

$$p \rightarrow B(p) = \{Y \mid (p, Y) \in BR\}$$

So both modern examples give rise to BNS, which was called a binary GrS in Lin (1998a). We would like to note that based on this (right) BNS, the (left) BNS can also be defined:

$$Y \rightarrow D(Y) = \{p \mid Y \in B(p) \text{ for all } p \in V\}.$$  

Note that BNS is a special case of NS (defined in 1st GrC model), namely, it is the case when the collection $NS(p)$ is an empty set or a singleton $B(p)$. So the 3rd GrC model is a special case of 1st GrC model.

The algebraic notion, BRs, in computer science, is often represented geometrically as graphs, networks, forest, etc. So the 3rd GrC model has captured most of the mathematical structure in computer science.

Next, instead of a single BR, we consider the case: $\beta$ is a set of BRs. It was called a [binary] knowledge base (Lin, 1998a). Such a collection naturally defines a NS.

**Definition 10:** 4th GrC model is the pair $(U, \beta)$, where $\beta$ is a set of BRs, It may be referred to as multi-binary GrC model.

This model is most useful in databases; hence it has been called binary granular data model (BGDM), in the case that $\beta$ is a collection of equivalence relations, it is called granular data model (GDM)

**Proposition 2:** The collection of 4th GrC models induces an onto map to the collection of 1st GrC models.

Observe that 4th GrC model induces a 1st GrC model as follows: Let $\beta = \{\beta_i \mid i \in I\}$, then $\{B_i(p) = \{q \mid (p, q) \in \beta_i\}\}$ for all $i \in I$, form a bag of subsets (distinct $i$ may produce
the same \( B(p) \); recall that a bag is a set with repeated elements (Garcia-Molina et al., 2002). Let \( NS(p) = \{B_i(p)\} \) be the set (collapsed from that bag). Then the collection \( \{NS(p) | p \in U\} \) is a NS; see Section 7.

Conversely, the selection of one \( B_i(p) \) from \( NS(p) \) at each point \( p \) defines a BNS and hence, a BR. Consider all possible such selections, we have a 4th GrC model that induces the given 1st GrC model; this shows the onto-ness

10 NS – 1st GrC model

In previous section, we have shown that 4th GrC model defines 1st GrC model, So, it is clear how one can deduce 1st GrC model from the ‘final’ GrC model.

The ancient intuitive notion of infinitesimal granule, \([E2]\) in Section 3, has been formalised in two ways:

1. the formal infinitesimal granule in non-standard world (NSW)
2. TNS in standard world.

We will focus on TNS. It is important to observe that the ancient intuition of infinitesimal granules (with the required properties) is formalised, not by a set, but by a family of subsets, that satisfies the (local version) axioms of topology. This family is denoted by \( TNS(p) \) and is called TNS at \( p \). Nevertheless, in this paper a (modern) granule will refer to a neighbourhood, but, not to the whole NS.

**Definition 11:** The notion of topology can be defined in two equivalent ways:

1. **Global version:** A topology \( \tau \) is a family of subsets, called open sets, that satisfies the axioms of topology: \( \tau \) is closed under finite intersections and arbitrary unions of \( \tau \).
2. **Local version:** A topology, called TNS, is an assignment that associates each point \( p \) a family of subsets, \( TNS(p) \), that satisfies the following axioms of topology:
   - If \( N \in TNS(p) \), then \( p \in N \)
   - If \( N_1 \) and \( N_2 \) are member of \( TNS(p) \), then \( N_1 \cap N_2 \in TNS(p) \)
   - If \( N_1 \in TNS(p) \) and \( N_1 \subseteq N_2 \), then \( N_2 \in TNS(p) \)
   - If \( N_1 \in TNS(p) \) then there is a member \( N_2 \in TNS(p) \) such that \( N_2 \subseteq N_1 \) and \( N_2 \in TNS(q) \) for each \( q \in N_2 \) (that is, \( N_2 \) is a neighbourhood of each of its point).

These two definitions lead us to first and second GrC models (local and global GrC models).

Let \( U \) and \( V \) be two classical sets. Let \( NS \) be a mapping, called NS,

\[
NS : V \rightarrow 2^{(P(U))},
\]

where \( P(X) \) is the family of all crisp/fuzzy subsets of \( X \). \( 2^Y \) is the family of all crisp subsets of \( Y \), where \( Y = P(U) \). In other words, \( NS \) associates each point \( p \) in \( V \), a family \( NS(p) \) of crisp/fuzzy subsets of \( U \). Such a subset is called a neighbourhood (granule) at \( p \) and \( NS(p) \) is called a NS at \( p \).

**Definition 12:** The three-tuple \((V, U, \beta)\) is called 1st GrC model (local GrC model), where \( \beta \) is a NS. If \( V = U \), the three-tuple is reduced to a pair \((U, \beta)\). In addition, if we require NS to satisfy the topological axioms, then it becomes a TNS.

Brief pre-GrC historical notes:

1. In 1988–1989, Lin generalised TNS to the NS by simply dropping the (local version) axioms of topology (Lin, 1988, 1989b) and apply it to approximate retrievals. Each neighbourhood was treated as a unit of uncertainty.

2. In the same year (1989), Lin also examined a BR (conflict of interests) for computer security from the view of NS (Lin, 1989a). Each neighbourhood represents a unit of knowledge.


4. In much earlier, NS was studied in Sierpenski (1952) as a generalisation of topology. Note that however, there are fundamental differences, for example, NS can be empty, while his cannot. Moreover, the concept of closures is different. The term pre-topology also has been used for referring NS and TNS.

5. In early GrC period, Lin, by mapping the NS onto Zadeh’s intuitive definition, used NS as his first mathematical GrC model (Lin, 1998a, 1998b, 1999).

### 10.1 The concept of ‘near’ and context

The following arguments are adopted from my pre-GrC paper (Lin, 1997). The notion of near is rather difficult to formalise. Let us examine the following examples.

1. Is Santa Monica ‘near’ Los Angeles? Answers could vary. For local residents, who have cars, answers are often ‘yes’. For visitors, who have no cars, answers may be ‘no’.

2. Is 1.73 ‘near’ \( \sqrt{3} \)? Again, answers vary; they depend on what was the agreement on the tolerance radius, in other words, in the given context.

Intrinsically, ‘near’ is a subjective judgment. One might wonder whether there is a scientific theory for such subjective judgments.

Mathematical analysis has offered a nice solution.
They simply include all possible contexts into its formalism. Here is the formalism of the second question: given the radius of an acceptable error, say, radius of errors 1/100 (a selected context).

Is 1.73 ‘near’ $\sqrt{3}$?

With this context selection, that is, 1/100 is acceptable error, then 1.73 is near $\sqrt{3}$! On the other hands, if the context (agreement) has changed to 1/1000, then 1.73 is NOT near $\sqrt{3}$! Clearly, in this numerical example, the collection of all possible contexts is the collection of all positive real numbers. We often use $\varepsilon$ to denote the variable that varies through such a context set.

Similarly, let us assume a NS has been assigned to each city in Los Angeles area. For example, based on car driving, public transportation, walking and, etc., we assign a neighbourhood to each city for each context. Under such a concept of NS, the question 1 above can be formulated properly as follows:

Assuming that we have selected context (taking public transportation), is Santa Monica ‘near’ Los Angeles?

Now we can have a definite answer to this question.

So a NS is a good infrastructure for addressing the concept of ‘near’! These analysis leads to the following conclusions.

1. in modelling, a NS is a good infrastructure for providing all possible contexts
2. under this model, in an application, selecting a context means selecting a fixed neighbourhood as a unit of tolerance (uncertainty).

Now, based on such a concept, we reexamine previous examples

*Example 3:* If we have chosen ‘driving half an hour’ as acceptable distance of ‘near’, then Santa Monica is ‘near’ Los Angeles.

*Example 4:* Let the collection of $\varepsilon$-neighbourhoods at each point be the NS of the real numbers $R$. Then $(R, \varepsilon$-neighbourhoods) provides the proper contexts for discussing ‘near’ answers at $p \in R$ (approximate answers), where $\varepsilon$ could take any real value. Let $p$ varies through the real numbers then $(R, \varepsilon$-neighbourhoods) = $\{(p, \varepsilon$-neighbourhoods($p$)| for all $p$} is a first GrC model on $R$.

Now, we re-state the previous example using this first GrC model

1. assuming we have agreed $\varepsilon = 1/100$ is acceptable, then 1.73 is ‘near’ $\sqrt{3}$
2. but, if we have only agreed $\varepsilon = 1/1000$ then 1.73 is not ‘near’ $\sqrt{3}$
3. next, let us consider a deeper question

Is the sequence 1, 1/2, 1/3, …, 1/n, … ‘near’ zero?
By ‘near’ zero we mean: For any given \( \epsilon > 0 \) (a context at zero), there is a number \( N = \lceil 1/\epsilon \rceil + 1 \), such that, \( \epsilon > 1/n \) for all \( n > N \), where \( \lceil \cdot \rceil \) is the integer part of \( \cdot \). Such a concept of ‘near’ for all contexts is said to be absolutely ‘near’.

For readers who familiar with the standard \((\epsilon, \delta)\)-definition of limit can spot the origin of NSs. Such a context free (all possible contexts) answer is precisely the classical notion of limits, \( \lim_{n \to \infty} 1/n = 0 \). Using our language, we may say that limit is the context free answers of ‘near’.

Perhaps we should also point out here that there are no context free answers for the question whether two points are ‘near’.

- A lesson
  1. each granule represents a context, a state or an agreement as what to be considered as ‘near’
  2. GrS \( \beta \) provides the complete contexts/states that can be used in reasoning about ‘nearness’.

11 Qualitative fuzzy sets – 9th GrC model

Fuzzy theories have been driven by applications. Naturally implicitly or explicitly, some contexts or hidden assumptions of their specific applications may have imposed into their theoretical frameworks, for example, the selection of a membership function is a selection of a context.

Based on the lesson from NS, we will develop a fuzzy set theory that provides all possible contexts. So a ‘real world fuzzy set’ should be assigned a ‘complete’ set of membership functions that can represent every admissible context.

Let \( U \) be the universe and \( X = \text{MF}(U) \) be the set of all membership functions on \( U \). Let \( \mathcal{W} \) be a collection of ‘real world fuzzy sets’ on \( U \), for example, a granulation (collection of granules) of human body.

**Definition 13:** A qualitative fuzzy set is a NS \( Q \) on \( \mathcal{W} \), namely, a map

\[
Q : \mathcal{W} \to 2^X
\]

such that \( Q \) assigns each real world fuzzy set a ‘complete’ set of membership functions that represent every admissible context. A qualitative fuzzy set will be denoted by \((w, Q(w))\).

This definition is a theoretical one. It is very similar to the concept of attribute domain in a database system. In a database system, an attribute, say city names, means its attribute domains consists of all city names in the past, present and future. Of course, there is no way to know all possible city names in the future, so the concept of attribute domain really is for theoretical use only; it is not an implementable concept. The notion of real world fuzzy set has a similar nature. Theoretically, it consists of all experts’ membership functions, each selection models the given a real world fuzzy set as a type I fuzzy set.
Definition 14: A context/state of a real world fuzzy set \( w \in W \) is a selection of a membership function in \( Q(w) \).

Definition 15: A 9th GrC model consists of \( \{ U, \{ Q(w) \mid w \in W \} \} \) and we say \( U \) is granulated by qualitative fuzzy sets \( \{ Q(w), w \in W \} \).

Example 5: Let \( U \) be the universe, namely, the human body. Let \( W \) be a real world granulation of \( U \) that consists of the collection of real world fuzzy sets, head, neck and so forth. Then the pair \( \{ U, \{ Q(w) \mid w \in W \} \} \) is a mathematical model of this real world granulation. In real applications, we select (based on experts’ opinion) one membership function from each \( Q \) (head), \( Q \) (neck)… So a granulation with a given context is \( \{ U, \) a membership function for each granule, head, neck, etc.\).

12 Conclusions and future directions

Since the birth of GrC, more than a decade has passed, there have been many directions and many authors. However, this paper proposes the first mathematical model of GrC that can formalise all classical examples. This model is an accumulation of previous nine models. Since this paper is not a survey paper, we shall cite only three early books for general purpose (Lin et al., 2002; Pedrycz, 2001; Inuiguchi et al., 2003), we will focus only on those issues that are touched in this article.

1 GrC and RST

RST has been served as the guiding ‘model’ for GrC developments (Lin, 1992, 1996, 2005, 2006a; Chiang et al., 2005; Lin and Liau, 2005). So there is a lot of similarity. However, we would like to caution the readers that, there are fundamental differences. For example, the fundamental views of uncertainty are quite different; Pawlak used ‘unable to specify’ as the base of uncertainty, while GrC regard a granule as a unit of uncertainty (such as uncertainty in quantum mechanics). Also the approximation theories are different.

2 GrC and databases

As we have pointed out that the categorical structures of databases and GrC are similar. So many database technology should be able to import to GrC. Historically, they are quite close; see brief pre-GrC historical note in Section 10.

Though the mathematical structures are equivalence, their semantic are very different. Nevertheless, we are looking forward to the transfer of database technology to GrC.

3 GrC and data mining

As in the classical example [E6] Section 3, we have pointed out that frequent itemset is genuine granules. So the close interactions have been explored and more is expecting (Lin, 1998a, 2000, 2002, 2004).

4 GrC and fuzzy logic

Most of expositions have been based on classical sets (and fuzzifiable concepts). For more intrinsic fuzzy view, we strongly recommend the readers to read Zadeh’s articles.
5 GrC and clouding computing

Theoretically, cloud computing can be related to the GrC on the category of Turing machines. We are expecting some stronger interactions.

6 Developments of categories

In this paper, a category based model is proposed as the formal model for GrC. It can be specialised into various models to realise all the classical examples. Roughly granulation is a family of sub-objects in product objects. However, all the details have not been explored especially, the categories of functions, random variables (measurable functions), Turing machines and qualitative fuzzy sets; see forthcoming papers.

References


Granular computing I: the concept of granulation and its formal model


Data mining in rough relational databases by attribute
generalisation

Theresa Beaubouef*

Computer Science Department, SLU 10847,
Southeastern Louisiana University,
Hammond, LA 70402, USA
Fax: (985) 549-5532 E-mail: tbeaubouef@selu.edu

Frederick E. Petry

Naval Research Laboratory,
Bldg. 1005, Code 7440.5,
Stennis Space Center, MS 39529, USA
Fax: (228)-688-4853 E-mail: fpetry@nrlssc.navy.mil

Abstract: The rough relational database model was developed for the management of uncertainty in relational databases. In particular, knowledge discovery performed on this database can provide useful information in the form of rules that may not have been known previously. A particular type of knowledge discovery, attribute oriented induction of rules from generalised data, is discussed in this paper, along with the advantages of the rough relational database model for this approach.

Keywords: data mining; rough relational database; attribute generalisation; knowledge discovery; concept hierarchy.


Biographical notes: Since earning her PhD in Computer Science from Tulane University in 1994, Theresa Beaubouef has held positions with the Naval Research Laboratory and Xavier University, and is currently a Professor at Southeastern Louisiana University. Her research interests include the areas of uncertainty in databases, spatial databases and data mining. In 2007, she was awarded with the university’s President’s Award for Excellence in Research.

Frederick E. Petry received his PhD in Computer Science from Ohio State University (1975). He was a Professor at the University of Alabama (Huntsville), Ohio State and Tulane University and is currently with the Naval Research Laboratory. He has over 300 publications including 115 journal articles/book chapters and seven books. His monograph on fuzzy databases is recognised as definitive on this topic. He was an IEEE Fellow (1996), an IFSA Fellow (2003) and an ACM Distinguished Scientist (2008). In 2002, he was the Outstanding Researcher of the Year (Tulane School of Engineering) and received Naval Research Laboratory’s Berman Research Publication awards (2004, 2008).
1 Introduction

Databases are recognised for their ability to store and update data in an efficient manner, providing reliability and the elimination of data redundancy. The relational database model, in particular, has well-established mechanisms built into the model for properly designing the database and maintaining data integrity and consistency. Data alone, however, are only facts. What is needed is information. Knowledge discovery attempts to derive information from the pure facts, discovering high level regularities in the data. It is defined as the nontrivial extraction of implicit, previously unknown and potentially useful information from data (Frawley et al., 1991; Tan et al., 2006).

An innovative technique in the field of uncertainty and knowledge discovery is based on rough sets. Rough set theory, introduced and further developed mathematically by (Pawlak, 1984), provides a framework for the representation of uncertainty. It has been used in various applications, most notably for representation of uncertainty in databases for data mining and improved information retrieval (Beaubouef and Petry, 1994; Srinivasan, 1991).

In a previous work, we developed the rough relational database model (Beaubouef et al., 1995) based on rough set theory. In this paper, we expand our work to include attribute oriented induction (AOI) of characteristic rules based on generalisation of the rough relational database. We follow the approach taken by (Han et al., 1992) for ordinary relational databases. This technique may readily be extended for use with object-oriented, spatial and other complex databases and may be integrated with additional data mining techniques for a comprehensive knowledge discovery approach.

2 Background

For some time now, rough sets (Pawlak, 1991) have been established as a technique for uncertainty management for both data and queries in relational databases in several ways (Beaubouef and Petry, 1994, 2000; Beaubouef et al., 1995) and in (Slowinski, 1992), for generalisation related to rough set indiscernibility. In rough sets an approximation space is defined on some universe U by imposing upon it some equivalence relation that partitions the universe into equivalence classes called elementary sets, based on some definition of ‘equivalence’ as it relates to the application domain. This partitioning can be adjusted to increase or decrease the granularity of a domain, to group items together that are considered indiscernible for a given purpose or to ‘bin’ ordered domains into range groups. Rough sets involve the following:

- \( R \) is the indiscernibility relation or equivalence relation
- \( A = (U, R) \), an ordered pair, is called an approximation space
- \([x]_R\) denotes the equivalence class of \( R \) containing \( x \), for any element \( x \) of \( U \)
- elementary sets in \( A \) – the equivalence classes of \( R \).

Any finite union of these elementary sets is called a definable set. A rough set \( X \subseteq U \), however, can be defined in terms of the definable sets in terms of its lower \( (\underline{A}X) \) and upper \( (\overline{A}X) \) approximation areas or regions where \([x]_R\) is the equivalence class of \( x \).
\[ AX = \{ x \in U \mid [x]_R \subseteq X \} \]
\[ \overline{AX} = \{ x \in U \mid [x]_R \cap X \neq \emptyset \}. \]

We may refer to \( \overline{AX} \) as the positive region \( U \), \( AX \) as the negative region and \( AX - \overline{AX} \) as the boundary or borderline region of the rough set \( X \). The lower and upper approximation regions, then, allow the distinction between certain and possible inclusion in a rough set. We may obtain possible results, in addition to the obvious, certain results if we also consider the boundary region information.

The rough relational database model is an extension of the standard relational database model of Codd (1970). It captures all the essential features of the theory of rough sets including the notion of indiscernibility of elements through the use of equivalence classes and the idea of denoting an indefinable set by its lower and upper approximation regions. Full details of the rough relational database model are found in (Beaubouef et al., 1995). We review only a few relevant definitions here.

The attribute domains in this model are partitioned by equivalence relations designated by the database designer or user. Within each domain, a group of values that are considered indiscernible form an equivalence class. The query mechanism uses class equivalence rather than value equality in retrievals. A user may not know the particular attribute value, but might be able to think of a value that is equivalent to the value required. For example, if the query requests COLOUR = "BROWN", the result will contain all colours that are defined as equivalent to BROWN, such as TAN, SORREL or CHESTNUT. Therefore, the exact wording of a query is less critical.

The rough relational database retains significant features of the ordinary relational database. Both models represent data as a set of relations containing tuples. The relations themselves are also sets. The tuples of a relation are its elements and like the elements of sets in general, are unordered and non-duplicated. A tuple \( t_i \) takes the form \((d_{i1}, d_{i2}, ..., d_{im})\), where \( d_{ij} \) is a domain value of a particular domain set \( D_j \). In the ordinary relational database, \( d_{ij} \in D_j \). In the rough relational database, however, as in other non-first normal form extensions to the relational model (Makinouchi, 1977; Roth et al., 1987), \( d_{ij} \subseteq D_j \) and although not required that \( d_{ij} \) be a singleton, \( d_{ij} \neq \emptyset \). Let \( P(D_i) \) denote the powerset \( (D_i) - \emptyset \).

**Definition 1:** A rough relation \( R \) is a subset of the set cross product \( P(D_1) \times P(D_2) \times \cdots \times P(D_m) \).

A rough tuple \( t \) is any member of \( R \), which implies that it is also a member of \( P(D_1) \times P(D_2) \times \cdots \times P(D_m) \). If \( t_i \) is some arbitrary tuple, then \( t_i = (d_{i1}, d_{i2}, ..., d_{im}) \) where \( d_{ij} \subseteq D_j \). A tuple in this model differs from that of ordinary databases in that the tuple components may be sets of domain values rather than single values.

**Definition 2:** An interpretation \((x_1, x_2, ..., x_m)\) of a rough tuple \( t_i = (d_{i1}, d_{i2}, ..., d_{im}) \) is any value assignment such that \( x_j \in d_{ij} \) for all \( j \).

In crisp databases clearly there is only one possible interpretation for a tuple, which is not always the case in rough relational databases. This introduces the issue of how to remove duplicate or redundant tuples. This is simple in ordinary databases but not so in the rough model. To deal with this we provide the following definition.
Definition 3: Tuples $t_i = (d_{i1}, d_{i2}, ..., d_{im})$ and $t_k = (d_{k1}, d_{k2}, ..., d_{km})$ are redundant if $[d_{ij}] = [d_{kj}]$ for all $j = 1, ..., m$, where $[d_{ij}]$ is the equivalence class containing $d_{ij}$.

Thus, after an operation on the database resulting in a new relation it is necessary to consider if any tuples must be merged because they are redundant.

At a higher structural level of the database we see that each rough relation $R$ is composed of subrelations corresponding to tuples that are certainly or possibly in $R$, i.e., the positive and boundary regions of $R$. We refer to these subrelations as $RP$ and $RB$ for convenience.

3 Attribute-oriented generalisation

The basis of a generalisation data mining approach rests on three aspects (Han and Kamber, 2000):

1. the set of data relevant to a given data mining task
2. the expected form of knowledge to be discovered
3. the background knowledge, which usually supports the whole process of knowledge acquisition.

Generalisation of data is typically performed with utilisation of concept hierarchies, which in ordinary databases are considered to be a part of background knowledge and are indispensable for the process. We assume that for selected attributes, $A_i$, in the database, one or more concept hierarchies, $H_k$, are available to provide generalisation for the attribute values.

The idea of using concept hierarchies for attribute-oriented induction in data mining has been investigated by several research groups (Han et al., 1992; Han, 1995; Carter and Hamilton, 1998; Hilderman et al., 1999). Generalisation of database objects is performed on an attribute-by-attribute basis, applying a separate concept hierarchy for each of the generalised attributes included in the relation of task-relevant data.

The basic steps/guidelines for attribute-oriented generalisation in an object-oriented database are summarised below (Han et al., 1994):

1. An initial query $Q$ to the database $DB$ provides the starting generalisation relation $R_G(A_1, ..., A_n)$ which contains the set of data that is relevant to the user’s generalisation interest.

   $$Q(DB) \Rightarrow R_G$$

2. Let $D(A_i)$ be the domain set of the attribute $A_i$ in the database $DB$ and $D_G(A_i)$ be the corresponding set of distinct values in $R_G$. Note $|D(A_i)|$ is the cardinality of $D(A_i)$.

   After the query $Q$, we will have $|D_G(A_i)| \leq |D(A_i)|$. Now, if there is a large set of distinct values for an attribute $A_i$ in $R_G$ but there is no higher level concept in any $H_k$ provided for the attribute, it should be removed in the generalisation process. That is if $|D_G(A_i)| \geq N_{Large}(A_i)$
and generalisation is not possible for $A_i$, then

$$R_G'(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n) \leftarrow R_G(A_1, \ldots, A_{i-1}, A_i, A_{i+1}, \ldots, A_n)$$

When all such attributes have been deleted we can show the relation re-indexed for convenience as

$$R_G'(A_1, \ldots, A_m)$$

where $m = n - \#$ of deleted attributes.

For this step we could assume in general that there is a supplied threshold $N_{\text{large}}(A_i)$ of distinct values for each attribute $A_i$. Such a threshold could be expressed as a fuzzy bound to represent the linguistic term ‘Large’.

3 Assume a hierarchy $H_k$ is now available to generalise the values, $D_G(A_i)$, for some attribute $A_i$. If there exists a higher-level concept in the concept tree for an attribute value of a tuple $t$, the substitution of the value by its higher-level concept generalises the tuple. Minimal generalisation should be enforced by ascending the tree one level at a time.

For example assume some subset $S = \{x_1, x_2, x_3\}$, $S \subseteq D_G(A_i)$, of values generalises to the concept $z_p$ in $H_k$ (see Figure 1).

![Figure 1](image)

Portion of a concept hierarchy $H_k$ for attribute $A_i$

Then we have

$$t_r(v_{r1}, \ldots, v_{ri} = z_{p}, v_{ri+1}, \ldots) \leftarrow t_r(v_{r1}, \ldots, v_{ri} = x_2, v_{ri+1}, \ldots)$$

4 As the tuples of the relation $R_G$ are generalised as in Step 3 above some may become similar or redundant (equal) and so should not be maintained as ‘duplicates’ in $R_G$.

Let $S(t_r, t_s)$ be the similarity of the two tuples. If two generalised tuples have become similar enough they are merged and an attribute, $A_{\text{count}}$, is added to keep track of how many tuples have been merged to form the current generalised tuple. So if $S(t_r, t_s) = 1$, then

$$t_r(v_{r1}, \ldots, v_{ri}, \ldots) = t_s(v_{s1}, \ldots, v_{si}, \ldots)$$

and we have the merger of these tuples

$$t'(v_1 = v_{r1} = v_{s1}, \ldots, v_i = v_{ri} = v_{si}, \ldots, v_{\text{count}} = 2)$$
The value of the count of a tuple should be carried to its generalised tuple and the counts should be accumulated when merging identical tuples in generalisation.

For databases that utilise representations of uncertainty such as fuzzy (Petry, 1996) and rough databases (Beaubouef et al., 1995) determination of the similarity of tuples is more complex than with exact matching as above. In such cases, we may have \( S(t_i, t_s) < 1 \). However this does not imply that these tuples should not be merged. For these databases we could have a similarity threshold \( T_S \) and if

\[
S(t_i, t_s) > T_S
\]

then the two tuples should be merged. The form of the resultant merged tuple would depend on the database representation.

5 The generalisation is controlled by providing levels that specify how far the process should proceed. If the number of distinct values of an attribute in the given relation is larger than the generalisation threshold value, further generalisation on this attribute should be performed. If the number of tuples in a generalised relation is larger than their generalisation threshold value, the generalisation should proceed further. We can then extract characteristic rules from generalised data.

Attribute generalisation should not be mistaken for simple record summarisation. Summaries of data usually have a much more simplified character and tend to omit data which do not occur originally in large quantities in order to simplify the final report. Gradual generalisation through concept hierarchies allows in contrast, detailed tracking of all data objects and can lead to the discovery of interesting patterns among data at the lowest possible abstraction level of their occurrence, decreasing at the same time the risk of omitting them due to over-generalisation. The appropriate attribute-oriented generalisation allows extraction of knowledge on a specific abstraction level but without omitting even rare attribute values. It might occur that such atypical values, despite being initially (at low level of the generalisation hierarchy) infrequent, can sum up to quite impressive cardinalities when generalised to an efficiently high abstraction level, which can then sometimes strongly influence the suspected proportions among the original data.

So depending on the approach and the intention of data analysts, generalisation of collected data can be treated either as a final step of data mining (e.g., summary-tables are presented to users, allowing them to interpret overall information) or as an introduction to further knowledge extraction (e.g., extraction of abstract association rules directly from the generalised data).

4 Uncertainty approaches in data summarisation and generalisation

Data summarisation can also be developed without concept hierarchies by using a linguistic fuzzy set theory; an important consideration is the treatment of data from a linguistic viewpoint. This approach uses linguistically quantified propositions to summarise the content of a database, by providing a general characterisation of the analysed data (Yager, 1991; Kacprzyk, 1999). A similar approach to fuzzy summarisation for data warehouses has been developed (Feng and Dillon, 2003).

The use of fuzzy concept hierarchies for AOI has been one approach to introducing uncertainty in data generalisation. Fuzzy hierarchies of concepts seem to better model
real life dependencies, since they are able to reflect the degree with which one concept belongs to its direct abstract and more than one direct abstract of a single concept is also allowed. ISA hierarchies, from area of data modelling, have been used to generalise database records to more abstract concepts (Lee and Kim, 1997; Petry and Zhao, 2009). Fuzzy gradual rules for data summarisation have been investigated (Cubero et al., 1999) and the SaintEtiq system for data summarisation through extended concept hierarchies has been implemented (Raschia and Mouaddib, 2002). Consistent fuzzy concept hierarchies, where each degree of membership is normalised to preserve an exact vote propagation of each tuple when generalised is another area of research (Angryk and Petry, 2003). This approach has also been applied to fuzzy databases that use either similarity or proximity relationships to represent uncertainty (Angryk et al., 2004).

5 Concept hierarchies

We begin by briefly discussing the idea of a concept hierarchy associated with an attribute. Let \( A \) be an attribute variable and let \( X \) be the domain of possible data values of \( A \). A concept hierarchy consists of a number of levels each of which is a partitioning of the space \( X \). Furthermore, this partitioning becomes coarser and coarser as we go up the hierarchy. The lowest possible level of a hierarchy consists of a partitioning by the individual elements of \( X \) and the highest level possible is the whole domain \( X \).

Formally, under our assumptions, each level of a crisp concept hierarchy \( H \) is an equivalence relationship. Thus, at level \( k \) of the concept hierarchy we have a relationship \( R_k : X \times X \rightarrow \{0,1\} \) that is

1. **Reflexive:** \( R_k(x, x) = 1 \)
2. **Symmetric:** \( R_k(x, y) = R_k(y, x) \)
3. **Transitive:** If \( R_k(x, y) = R_k(y, z) = 1 \) then \( R_k(x, z) = 1 \).

The semantics of this relationship is that \( R_k(x, y) = 1 \) indicates these two elements, \( x \) and \( y \), are essentially the same.

As it is well-known, such an equivalence relationship partitions the space \( X \) into \( n_k \) disjoint subsets of \( X \). These subsets denoted \( \mathcal{E}_k^i \), the \( i \)th class for the partition of level \( k \), are such that for \( x, y \in \mathcal{E}_k^i \) we have \( R_k(x, y) = 1 \).

The increased coarseness of partitioning as we ascend the concept hierarchy is reflected in the requirement that if \( k > j \) then for all pairs \( x \) and \( y \) we have \( R_k(x, y) \geq R_j(x, y) \). Essentially, this requires that if \( x \) and \( y \) are in the same class for level \( j \) of the hierarchy, they are in the same class in any higher-level \( k \). This implies that if \( k > j \) then for any equivalence class \( \mathcal{E}_j^i \) at level \( j \) there exists an equivalence class \( \mathcal{E}_k^* \) at level \( k \) such that \( \mathcal{E}_j^i \subseteq \mathcal{E}_k^* \).

Here we have implicitly assumed that the indexing of the levels begins with the lowest level denoted 1. At this lowest level we have \( R_1(x, y) = 0 \) iff \( x \neq y \), which means that any distinct data value is in its own class. If the highest level of the hierarchy, \( m \), consists simply of the whole domain \( X \), then \( R_m \) is such that \( R_m(x, y) = 1 \) for all \( x \) and \( y \).

So at each level \( k \), the concept hierarchy is a partition of the set of possible data values \( X \) into \( n_k \) categories (equivalence classes):
If we have \( m \) levels then the concept hierarchy is a collection of \( m \) partitions of the space \( X \). In particular, the concept hierarchy consists of

**Partition 1:** \( E_{1|i} \) for \( i = 1 \) to \( n_1 \)

**Partition 2:** \( E_{2|i} \) for \( i = 1 \) to \( n_2 \)

...,  

**Partition \( m \):** \( E_{m|i} \) for \( i = 1 \) to \( n_m \)

We should note while formally each category \( E_{k|i} \) corresponds to a subset of the data space \( X \), typically it has an associated name that essentially describes the elements in \( E_{k|i} \). In the following we shall generally make no distinction between these two uses of \( E_{k|i} \), as a subset of \( X \) and as a denotation of the subset.

### 6 Generalisation in terms of partitions

The preceding hierarchy concepts notation is useful for describing attribute generalisation. For simplicity we focus on a simple relation \( R(A_1) \) that has only one attribute \( A_1 \), where \( D(A_1) \) denotes the domain of values for \( A_1 \). In some concept hierarchy at level \( m \), assume we have \( E_{m|1}, E_{m|2}, \ldots \).

In general, the specific database for the relation \( R(A_1) \) is \( D(A_1) \subseteq D(A_1) \) and corresponding to this subset of data we have \( E_{m|i} \subseteq E_{m|i} \). If no specific data values in \( R \) generalise to the concept \( C_{m|i} \), then some of the \( E_{m|i} \) may be empty. \( E_{m|i} \) corresponds to those tuples that are merged in our simplified case of \( D(A_1) \), giving as the merged tuple \( t'(C_{m|i}, E_{m|i}) \), where \( |E_{m|i}| \) is the cardinality or count, of the number of tuples merged into \( t' \). After generalisation to a level \( m \), the number of tuples in the single attribute relation \( R \) is given by

\[
N = \sum_{i=1}^{m} \left\{ 1 \text{ if } E_{m|i} \neq \emptyset; 0 \text{ otherwise} \right\}
\]

For relations having more than one attribute, \( R(A_1, A_2) \), for example, then when \( A_1 \) is again generalised, not all tuples can be merged. We may have two generalised tuples having differing \( A_2 \) values such as

\[
t'(C_{m|i}, a) \neq t'(C_{m|i}, b)
\]

that cannot be merged. If \( R'(A_1, A_2) \) is the relation after possible mergings, then \( N \) is a lower bound on the number of tuples of \( R'(A_1, A_2) \) or \( N \leq |R'(A_1, A_2)| \).

Here we have discussed the generalisation of crisp data using crisp hierarchies. Next, we consider the generalisation of rough data with crisp hierarchies.
7 Knowledge discovery by generalisation of rough data with crisp hierarchies

We learn characteristic rules from generalised data. Other types of rules, such as quantitative and discrimination rules are also possible, but are not discussed in this paper. We automatically examine the data in rough tuples, based on certain things. First, however, we must formalise the generalisation of rough data. Again, we focus first on a simple rough relation having only one attribute $R(A_1)$, where $D(A_1)$ denotes the domain of values for $A_1$. The approach discussed in the previous section applies here since the hierarchy is crisp. However, we must now consider the representation of uncertainty in generalised tuples, allowing for set-valued attributes and accounting for tuples in upper and lower approximation regions.

First, we consider generalisation for rough tuples in which the value of the attribute to be generalised is set of $m$ values

$$v_{ri} = \{x_1, x_2, \ldots, x_m\}, v_{r(i+1)}, \ldots$$

If all of the values, $x_1, x_2, \ldots, x_m$, generalise to the same concept $z_p$ then the generalisation proceeds as with a single valued attribute and the analysis in terms of partitions will apply.

The value of the attribute $v_{ri}$ is the set of values, $x_1, x_2, \ldots, x_m$, because these are indistinguishable based on the semantics given for the attribute of the rough database. However, from the set of hierarchies that might be used in generalisation of this attribute, a specific hierarchy might represent a semantic context in which such data values are not all considered indistinguishable. Assume for such a hierarchy one of the above data values, $x_i$, generalises to the different concept $z_t$. Since we are assuming a crisp hierarchy, the concepts $z_p$ and $z_t$ are distinct. Then the tuple $t_r$ must generalise to the two distinct tuples:

$$t_{r1} \left( v_{r1}, \ldots, v_{ri} = z_p, v_{r(i+1)}, \ldots \right)$$

$$t_{r2} \left( v_{r1}, \ldots, v_{ri} = z_t, v_{r(i+1)}, \ldots \right)$$

The extreme case is that in which all $m$ data values generalise to $m$ different concepts $z_{p1}, z_{p2}, \ldots, z_{pm}$. Then we produce $m$ distinct tuples $t_{i1}, \ldots, t_{im}$. Now we must consider the counting of tuples in the cases of set-valued data for the attribute. In the extreme example discussed above where we obtain $m$ distinct tuples $t_{i1}, \ldots, t_{im}$ then the original tuple, $t_r$, would be contributing a total of $m$ to the counts of the various merged tuples. This would represent far more influence for the original $t_r$ as opposed to any of the original tuples that simply generalised to single tuples. We consider this to be inappropriate and as a solution we use proportional counting. In this extreme case, we then would have for merging the two tuples $t_r \left( v_{r1}, \ldots, v_{ri}, \ldots \right) = t_s \left( v_{s1}, \ldots, v_{si}, \ldots \right)$ where $t_s$ was one of the $m$ tuples above. So the merged tuple would be

$$t' \left( v_1 = v_{r1} = v_{s1}, \ldots, v_i = v_{ri} = v_{si}, \ldots, v_{Coun} = (1 + 1/m) \right)$$

Now, we wish to examine the generalisation of tuples as we have described above more formally. Since the attribute value of a rough tuple may be set valued we first provide a
representation of such an attribute’s domain. Let $X_i$ be the base set of values for the attribute $A_i$. Additionally, $A_i$ may take certain subsets of $X_i$ as values. Specifically let these subsets be $B_1, B_2, \ldots, B_m$. All $x, y \in B_i$ are indiscernible, i.e., are in the same elementary set. We now write the domain of $A_i$ as

$$X^+ = [X_i, B_1, B_2, \ldots, B_m]$$

where by $X_i$ we mean any singleton value of the underlying domain $X_i$. This lets us consider an extended concept hierarchy allowing set valued attributes to be generalised from the bottom level $k=1$ of the hierarchy.

The first case to consider is where all the values in a particular $B_i$ generalise to the same concept $z_p$. For a simple example, let $B_i = \{x_1, x_2\}$ and we have

Figure 2  Partitioning extended concept hierarchy

If all of the data generalisations are of this sort then for the extended domain $X^+$ we obtain a partitioning of the domain.

Next, consider the generalisation case as discussed above where the semantics of the indistinguishability relation and the concept hierarchy differ. So we might have the following concept hierarchy example in Figure 3.

Figure 3  Concept hierarchy

Here, the sets of the equivalence classes in such a concept hierarchy do not constitute a formal set partitioning of a domain $X^+$ as in Figure 2. So rather than partitioning the domain $X^+$, we have a set decomposition. At some level this would be the sets $D_{k|1}$, $D_{k|2}$, where in general

$$D_{k|1} \cap D_{k|2} \neq \emptyset$$
This implies that there may not be a unique concept at level \( k \) to which a value at level \( k - 1 \) generalises. An overlapping value, such as in Figure 3, is an example of this. This situation is similar to that which is found in a fuzzy hierarchy in which values have degrees of association to concepts at the next level of the hierarchy (Raschia and Mouaddib, 2002; Petry and Zhao, 2009).

The above discussion was at the level of generalisation of the individual tuples. For the rough relational database, we also manage uncertainty by partitioning the tuples in a relation based on the positive and boundary regions. We let \( \text{RP}(A_i) \) and \( \text{RB}(A_i) \) denote the sub-relations of \( R(A_i) \) for which the tuples represent data from the positive and boundary approximations respectively. So now the issue is how to manage generalisation of tuples from these subrelations.

We will consider two approaches:

1. Generalise tuples from \( \text{RP}(A_i) \) and \( \text{RB}(A_i) \) separately for a given hierarchy \( H \). The initial result will then be the two sets of generalised tuples \( \text{RP}'_G(A_i) \) and \( \text{RB}'_G(A_i) \).

   Now we must consider tuple merging for this case.
   
   a. Merge tuples in \( \text{RP}'_G(A_i) \) and \( \text{RB}'_G(A_i) \) independently. Maintain these final tuples separately in \( R'_G(A_i) = \text{RP}'_G(A_i) \cup \text{RB}'_G(A_i) \).

   b. Merge tuples as above. If there are similar tuples \( t \) and \( t' \) in \( \text{RP}'_G(A_i) \) and \( \text{RB}'_G(A_i) \): \( t \in \text{RP}'_G(A_i) \); \( t' \in \text{RB}'_G(A_i) \), then let \( t'' = \text{Merge}(t, t') \). Finally, use some decision function to determine which sub-relation to place the merged resultant tuple, \( t'' \).

2. Generalise \( \text{RP}(A_i) \) and \( \text{RB}(A_i) \) together for a given hierarchy \( H \). The initial result will be only one relation \( R'_G(A_i) \). However, keep track for each tuple as to their origin, from the positive and boundary approximations, respectively.

   a. Merge the tuples in \( R'_G(A_i) \), but keep two attribute counts, \( A_{\text{pct}} \) and \( A_{\text{bct}} \), to indicate the proportions of positive and boundary approximation tuples in the final tuples. Note this result is now a crisp database as there are no positive and boundary sub-relations.

   b. Follow the merging as described above but use a decision function possibly based on the upper and lower counts of the tuples to decide if the final merged tuple will belong to the final positive or boundary sub-relation.

A possible approach to a decision rule for merged tuple placement could be based on the semantics of positive and boundary sub-relations. Unless we have strong indication that most of the counts are from a positive region we should place the tuple in the boundary subrelation as it represents possibility as opposed to the certainty reflected in the positive region. One representation for such a rule might be to use a fuzzy function, ‘Most’ to determine this. Let \( x = \text{pct}/(\text{pct} + \text{bct}) \). We can express this by a fuzzy subset \( F \) on the unit interval such that \( F(x) \) indicates the degree to which the value \( x \) satisfies the user's idea of 'Most'. Thus, if \( F(0.85) = 1 \) this could indicate that the tuple might be placed into the positive subrelation \( \text{RP}_G(A_i) \). We note that formally \( F:[0,1] \rightarrow [0,1] \) which must satisfy the conditions \( F(0) = 0, F(1) = 1 \) and \( F(x) \geq F(y) \) if \( x > y. \)
The extraction of the function $F$ from a user can be assisted by the concept of linguistic quantifiers introduced by Zadeh (1983). He noted that human dialogue makes considerable use of terms such as *most*, *about 50%*, *some*, *all* which he referred to as linguistic quantifiers. These are used to provide a linguistic explanation of some proportion and can be represented by fuzzy subsets over the unit interval such that the membership measures the satisfaction to the concept. In Figure 4, we illustrate a typical graphical representation of $F$.

**Figure 4** Example of $F(x)$ for criterion ‘Most’

![Graphical representation of $F(x)$](image)

A specific example of this sort of function is illustrated by the function $F_1$ below:

$$F_1(x, a, b) = \begin{cases} 
0 & x < a \\
(x - a) / (b - a) & a \leq x \leq b \\
1 & x \geq b 
\end{cases}$$

where the values of $a$ and $b$ might be 0.75 and 0.85, respectively. The form of the function allows one to apply a threshold to decide in which specific subrelation to place the tuple. If desired it would be possible to use $F(x)$ to define a membership value for the tuple and extend the database to a fuzzy rough database model as described in (Beaubouef and Petry, 2000).

8 Database examples for generalisation of rough data with crisp hierarchies

We illustrate various concepts of the previous section with a sample rough relational database. This database is a subset of a herd management database for a large ranch housing various types and breeds of livestock. The schema of some relevant relations include, for example:

- **ANIMAL _ BREEDS**: (Breed, Size, Origin, Predominant use)
- **CATTLE**: (Tag, Breed, Category, Colour, Problem/note)
- **GOATS**: (Tag, Breed, Category, Colour, Problem/note).
Some attribute domains may be partitioned into equivalence classes such as

\[
\text{COLOUR} = \{ \{\text{BROWN, TAN, SORREL, CHESTNUT}\}, \{\text{GRAY, GRAY}\}, \\
\{\text{WHITE, BEIGE}\}, \{\text{BLACK, EBONY}\}, \ldots \} \\
\text{BREED} = \{ \{\text{BEEFMASTER, BM}\}, \{\text{ANGUS, BLACK ANGUS}\}, \ldots \} \\
\text{PROBLEM / NOTE} = \{ \{\text{GETS OUT, ESCAPES}\}, \{\text{DANGER, DANGEROUS}\}, \\
\{\text{HORNED}\}, \{\text{LIMPS, LAME, CRIPPLE}\}, \{\text{STERILE}\}, \ldots \} \\
\text{CATEGORY} = \{ \{\text{COW}\}, \{\text{BULL}\}, \{\text{STEER}\}, \{\text{NANNY}\}, \\
\{\text{BUCK, BILLY}\}, \{\text{KID}\}, \{\text{Calf}\}, \ldots \} \\
\]

where values belonging to the same class are considered indiscernible for the particular application.

Sample tuples from typical rough relations in this database are depicted in Table 1 below. For notational purposes, tuples of the boundary region of a rough relation are shown in italics if there are any. Normally boundary region tuples result from queries, but sometimes they occur in base relations.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Breed</th>
<th>Category</th>
<th>Colour</th>
<th>Problem/note</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>ANGUS</td>
<td>COW</td>
<td>BLACK</td>
<td>NONE</td>
</tr>
<tr>
<td>002</td>
<td>ANGUS</td>
<td>COW</td>
<td>BLACK</td>
<td>NONE</td>
</tr>
<tr>
<td>003</td>
<td>{ANGUS, BM}</td>
<td>CALF</td>
<td>BLACK</td>
<td>GETS_OUT</td>
</tr>
<tr>
<td>004</td>
<td>{ANGUS, BM}</td>
<td>CALF</td>
<td>BROWN</td>
<td>ESCAPES</td>
</tr>
<tr>
<td>005</td>
<td>BEEFMASTER</td>
<td>COW</td>
<td>TAN</td>
<td>HORNE D</td>
</tr>
<tr>
<td>006</td>
<td>ANGUS</td>
<td>BULL</td>
<td>BLACK</td>
<td>DANGER</td>
</tr>
<tr>
<td>007</td>
<td>JERSEY</td>
<td>STEER</td>
<td>BROWN</td>
<td>STERILE</td>
</tr>
<tr>
<td>008</td>
<td>HOLSTEIN</td>
<td>COW</td>
<td>{BLACK, WHITE}</td>
<td>{STERILE, HORNE D}</td>
</tr>
<tr>
<td>009</td>
<td>CHAROLAIS</td>
<td>COW</td>
<td>BEIGE</td>
<td>HORNE D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Breed</th>
<th>Category</th>
<th>Colour</th>
<th>Problem/note</th>
</tr>
</thead>
<tbody>
<tr>
<td>020</td>
<td>BOER</td>
<td>KID</td>
<td>WHITE</td>
<td>{HORNED, GETS_OUT}</td>
</tr>
<tr>
<td>021</td>
<td>BOER</td>
<td>KID</td>
<td>BROWN</td>
<td>ESCAPES</td>
</tr>
<tr>
<td>022</td>
<td>NUBIAN</td>
<td>NANNY</td>
<td>GRAY</td>
<td>{HORNED, STERILE}</td>
</tr>
</tbody>
</table>

**ANIMAL_BREEDS**

<table>
<thead>
<tr>
<th>Breed</th>
<th>Size</th>
<th>Origin</th>
<th>Predominant use</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>LARGE</td>
<td>TEXAS</td>
<td>MEAT</td>
</tr>
<tr>
<td>ANGUS</td>
<td>SMALL</td>
<td>SCOTLAND</td>
<td>MEAT</td>
</tr>
<tr>
<td>BOER</td>
<td>LARGE</td>
<td>AFRICA</td>
<td>MEAT</td>
</tr>
</tbody>
</table>
Examination of only the small subset of data shown in these tables might lead to rules such as ‘all dairy cows are sterile’, since Jersey and Holstein are dairy cattle or ‘all calves escape’ or even the trivial ‘all angus bulls are dangerous’. Generalisation, however, will allow for the discovery of more rules. For example, assume we want to discover rules related to the attribute PROBLEM/NOTE related to all animals. In this case, we would need to gather all the task relevant data into a single rough relation by taking the union of CATTLE and GOATS. Next concept hierarchies must be defined for the PROBLEM/NOTE attribute and then generalisation can be performed. After generalisation, the rules are formed. As discussed in the previous section, there are several issues involved in the process of generalisation in the rough relational database. We continue the herd management example to clarify these issues.

8.1 Task relevant data

The first step in the knowledge discovery process is to gather together in one rough relation that data that is relevant for the task at hand. This data may extend over several rough relations and if so, must be combined by executing an appropriate query to obtain the target class. This eliminates unnecessary data from taking part in the generalisation process and thus, saves time and space.

Data in a database is considered positive examples of truth. If a rough tuple exists in the database, it is certainly true if it belongs in the lower approximation region and possibly true if it belongs to the boundary region of the rough relation.

8.2 Defining concept hierarchies

Concept hierarchies are associated with attributes. A domain expert or knowledge engineer will normally specify the groupings and levels for the concept hierarchy, but sometimes, as with numerical ranges, they can be calculated semi-automatically based on statistics. The specification of the concept hierarchy forms the background knowledge that guides the generalisation process and therefore, different concept hierarchies defined for an attribute may lead to different rules being discovered. It is sometimes desirable to experiment with various concept hierarchies, when applicable, in order to extract different types of knowledge.

An example of a concept hierarchy for the attribute BREED is shown below:

Concept hierarchy Example 1

\{
\{cattle, goat\} \prec livestock
\{beef -cattle, dairy -cattle\} \prec cattle
\{Beefmaster, Angus, Charolaix\} \prec beef -cattle
\{Jersey, Holstein\} \prec dairy -cattle
\{meat -goat, dairy -goat, wool -goat\} \prec goat
\{Boer, African\} \prec meat -goat
\{Nubian, Saanen, Toggenburg\} \prec dairy -goat
\{Angora, Cashmere\} \prec wool -goat
\ldots
\}
A different concept hierarchy based on purpose of the breed might be:

Concept hierarchy Example 2

\[
\begin{align*}
\{\text{meat-animal, dairy-animal}\} & \prec \text{livestock} \\
\{\text{dairy-goat, dairy-cattle}\} & \prec \text{dairy-animal} \\
\{\text{beef-cattle, meat-goat}\} & \prec \text{meat-animal} \\
\{\text{Beefmaster, Angus, Charolais}\} & \prec \text{beef-cattle} \\
\{\text{Jersey, Holstein}\} & \prec \text{dairy-cattle} \\
\{\text{Boer, African}\} & \prec \text{meat-goat} \\
\{\text{Nubian, Saanen, Toggenburg}\} & \prec \text{dairy-goat} \\
\{\text{Angora, Cashmere}\} & \prec \text{wool-goat} \\
\ldots
\end{align*}
\]

A different concept hierarchy based on size of the breed rather than purpose might be:

Concept hierarchy Example 3

\[
\begin{align*}
\{\text{small-body-breed, large-body-breed}\} & \prec \text{livestock} \\
\{\text{Angus, Jersey, Cashmere}\} & \prec \text{small-body-breed} \\
\{\text{Holstein, Charolais, Beefmaster, Nubian, Saanen, Toggenburg, Boer}\} & \prec \text{large-body-breed}
\end{align*}
\]

We could also have a concept hierarchy based on category, such as male adult, female adult, juvenile, etc.

8.3 Generalisation using concept hierarchies

Generalisation is performed for each attribute as discussed previously. Consider the following case: All values in a particular $B_i$ generalise to the same concept $z_p$, as depicted in Figure 2.

Let the concept hierarchy for COLOUR be defined as:

\[
\begin{align*}
\{\text{brown, tan, sorrel, chestnut}\}, \{\text{dun}\} & \prec \text{brownish} \\
\{\text{gray, grey}, \{\text{slate}\}\} & \prec \text{grayish} \\
\{\text{white, beige}, \{\text{blond}, \{\text{pearl}\}\}\} & \prec \text{lights} \\
\{\text{black, ebony}\} & \prec \text{black} \\
\ldots
\end{align*}
\]

\[
\begin{align*}
\{\text{lights, grayish, ...}\} & \prec \text{fair-coloured} \\
\{\text{black, brownish, ...}\} & \prec \text{dark-coloured}
\end{align*}
\]

Recall that the lowest level concepts are the individual equivalence classes. Consider the following sample set of tuples from the CATTLE relation:
Assume we are trying to discover rules related to colour and breed. The task relevant data then becomes:

\[
\begin{align*}
(31, \text{BEEFMASTER, COW, BROWN, STERILE}) \\
(38, \text{BEEFMASTER, COW, \{TAN, SORREL\}, ESCAPES}) \\
(39, \text{BEEFMASTER, COW, TAN, HORNED}) \\
(33, \text{ANGUS, BULL, BLACK, NONE}) \\
(54, \text{ANGUS, CALF, EBONY, ESCAPES}) \\
(55, \text{ANGUS, COW, \{BLACK, EBONY\}, NONE}) \\
(64, \text{BM, BULL, \{BROWN, TAN\}, NONE}) \\
(66, \text{BM, COW, CHESTNUT, NONE})
\end{align*}
\]

Here the first level of generalisation results in

\[
\begin{align*}
(\text{BEEFMASTER, BROWN}) \\
(\text{BEEFMASTER, \{TAN, SORREL\}}) \\
(\text{BEEFMASTER, TAN}) \\
(\text{ANGUS, BLACK}) \\
(\text{ANGUS, EBONY}) \\
(\text{ANGUS, \{BLACK, EBONY\}}) \\
(\text{BM, \{BROWN, TAN\}}) \\
(\text{BM, CHESTNUT})
\end{align*}
\]

because the first three original tuples have colours generalising to the concept ‘brownish’; the last two tuples generalise to ‘brownish’ and BM is equivalent to BEEFMASTER and so these tuples also merge. The next level of generalisation results in

\[
\begin{align*}
(\text{BEEFMASTER, \text{brownish}}, \text{Count} = 5) \\
(\text{ANGUS, \text{black}}, \text{Count} = 3)
\end{align*}
\]

From this example, we could infer that ‘angus cattle are black’ after the first generalisation with a count of three tuples supporting the rule. Additionally, after the second generalisation and a count of five tuples, we would obtain the rule ‘beefmasters are dark-coloured’.

A different situation exists when a concept hierarchy is defined on an existing rough relational database and the concept hierarchy does not generalise directly from the equivalence relations. For example, assume the equivalence classes previously defined, where COLOUR contains the classes
\[\text{COLOUR} = \{[\text{BROWN, TAN, ROAN, SORREL, CHESTNUT}], [\text{GRAY, GREY}], \]
\[\quad [\text{WHITE, BEIGE}], [\text{BLACK, EBONY}]\} \]

Now assume the user’s concept hierarchy is based on different semantics and has been defined as follows:

\[
\begin{align*}
\{[\text{brown, tan}, [\text{dun}]] &\prec \text{brownish} \\
\{[\text{sorrel, chestnut}] &\prec \text{dark-brown} \\
\{[\text{gray, grey}, [\text{slate}]] &\prec \text{grayish} \\
\{[\text{red, roan}]] &\prec \text{reddish} \\
\{[\text{white, beige}, [\text{blond}, [\text{pearl}]] &\prec \text{lights} \\
\{\text{black, ebony} &\prec \text{black} \\
\{\text{lights, grayish, }... &\prec \text{fair-coloured} \\
\{\text{brownish, reddish} &\prec \text{medium-coloured} \\
\{\text{black, dark-brown, }... &\prec \text{dark-coloured}. \\
\end{align*}
\]

Now, it is possible to have tuples that can generalise to more than one concept at the next higher level in the hierarchy such as

\[
\{68, \text{BEEFMASTER, COW, }\{\text{BROWN, SORREL}\}, \text{NONE}\}
\]
\[
\{69, \text{BEEFMASTER, CALF, }\{\text{ROAN, CHESTNUT, BROWN}\}, \text{NONE}\}.
\]

These tuples will then generalise to more than one tuple based on this differing concept hierarchy. The first tuple will generate the two tuples: (BEEFMASTER, \(\text{brownish}\)) and (BEEFMASTER, \(\text{dark-brown}\)). The second tuple generalises to three tuples: (BEEFMASTER, \(\text{reddish}\)), (BEEFMASTER, \(\text{dark-brown}\)) and (BEEFMASTER, \(\text{brownish}\)). If we consider the tuples that were generalised in first example in this section to then the last tuple above will contribute only 1/3 to the count if it were merged with those yielding:

\[
\{\text{BEEFMASTER, brownish}\}, \text{Count} = (5 + 1/3).
\]

We next consider issues related to the positive and boundary regions of the rough relation. Our first approach involves the generalisation and merging of tuples separately for each of these two regions. Let us consider again the concept hierarchy based on colour and recall that tuples in the boundary region are shown in italics. Assume that the task relevant data has been produced as follows, with four tuples in the positive region and two in the boundary region:
If we generalise the two regions separately, we obtain for the positive region the tuples

\[(\text{BEEFMASTER, BROWN}), \text{Count} = 2\]
\[(\text{BEEFMASTER, \{TAN, DUN\}})\]
\[(\text{ANGUS, BLACK})\]
\[(\text{ANGUS, EBONY})\]
\[(\text{ANGUS, \{BLACK, EBONY\}})\]
\[(\text{BEEFMASTER, BROWN})\]
\[(\text{BEEFMASTER, \{TAN, DUN\}})\]

as before. Additionally, for the boundary region, generalisation results in

\[(\text{BEEFMASTER, brownish}), \text{Count} = 2\]

and with the next level of generalisation

\[(\text{BEEFMASTER, dark-coloured}), \text{Count} = 2.\]

Rules can then be generated from generalised data from each of the subrelations resulting from each of the positive and boundary regions, separately, giving both ‘positive’ rules and ‘possible’ rules, respectively. Alternatively in this approach, the generalisation process can occur separately for the two regions, followed by a merging process that allows for the merging of generalised tuples of both regions together, using some decision function to determine which sub-relation to place merged tuples. In the current example, we may decide in merging the (BEEFMASTER, dark-coloured) tuples that certain information is always certain and that even uncertain information supports fully all the data. In this case we may generate a rule ‘beefmasters are dark-coloured’ based on these facts, rather than ‘beefmasters are possibly dark-coloured’, which would result if more emphasis were placed on the uncertainty of information.

The second approach generalises the positive and boundary regions together for a given hierarchy, keeping track of the origin of each tuple. Consider the sample tuples below:

\[(\text{BEEFMASTER, BROWN})\]
\[(\text{BEEFMASTER, \{TAN, DUN\}})\]
\[(\text{BEEFMASTER, \{TAN, BROWN\}})\]
\[(\text{ANGUS, BLACK})\]
\[(\text{ANGUS, EBONY})\]
\[(\text{ANGUS, \{BLACK, EBONY\}})\]
\[(\text{BEEFMASTER, BROWN})\]
\[(\text{BEEFMASTER, \{TAN, DUN\}})\]
The first level of generalisation results in the following generalised tuples:

- (BEEFMASTER, brownish), PositiveCount = 3, BoundaryCount = 2
- (ANGUS, black), PositiveCount = 3.

A second level generalisation results in

- (BEEFMASTER, dark-coloured), PositiveCount = 3, BoundaryCount = 2
- (ANGUS, dark-coloured), PositiveCount = 3.

Again, rules can be generated from these merged tuples based on the proportion of tuples from each region or by using some decision function.

9 Conclusions

In this paper, we investigated the extraction of characteristic rules from a rough relational database based on creating rules from generalised data through the process of generalisation using attribute induction. In this iterative process, data becomes more and more generalised until it reaches a point where some threshold criteria is met for forming rules from the generalised data. We illustrated these concepts using a sample subset of a herd management database.

The rough relational database naturally adapts to this type of knowledge discovery since the indiscernibility relation defined on attribute values already has implicit generalisation. Additionally, having the upper approximation boundary region present in the generalised data can result in rules that are possible, providing greater information to the domain expert or knowledge engineer. This information can be used in refining the attribute induction knowledge discovery process and can lead to heretofore undiscovered dependencies in the underlying data.

Acknowledgements

The authors would like to thank the Naval Research Laboratory’s Base Program, Program Element No. 0602435N for sponsoring this research.

References


Time series data analysis in multiple granularity levels

Mehmet Sayal*
Hewlett-Packard Labs,
1501 Page Mill Road,
Palo Alto, CA 94304 USA
E-mail: mehmet.sayal@hp.com
*Corresponding author

Ming-Chien Shan
SAP Research,
Palo Alto, CA 94304 USA
E-mail: ming-chien.shan@sap.com

Abstract: A single-pass method for analysing time series data and extracting time-correlations (time-delayed relationships) among multiple time series data streams is described. The proposed method can detect and report time-delayed relationships among multiple time series data streams without having to transform the original data into another domain. Each time-correlation rule explains how the changes in the values of one set of time series data streams influence the values in another set of time series data streams. Those rules can be stored digitally and fed into various data analysis tools for further analysis. Performance experiments showed that the described method is 95% accurate and has a linear running time with respect to the amount of input data for pair-wise time series correlations.

Keywords: time series; time domain analysis; correlation; convolution.


Biographical notes: Mehmet Sayal is a Senior Scientist at Hewlett-Packard Labs. His current interests include business intelligence, data mining, time series data stream analysis and procurement auction analysis and optimisation. He has more than 20 publications in those research areas, three granted US patents and approximately 12 applications in process. He has served as Chair, PC member, Editorial Board Member or reviewer in several prestigious conferences and journals, such as VLDB, IEEE ICDE, ACM CIKM, ACM Electronic Commerce, ACM SAC, Elsevier CAHE and INS, IEEE TKDE, IEEE NCA and WISE, IEEE DEECS, IEEE IWEA (2007, 2008 and 2009), ECBS and IJDMMM.

Copyright © 2009 Inderscience Enterprises Ltd.
1 Introduction

In this paper, a single-pass method for analysing time series data and extracting time-correlations (time-delayed relationships) among multiple time series data streams is described. The method described in this paper can detect and report time-delayed relationships among multiple time series data streams in an online fashion (i.e., as the data flows in continuously from a large number of data streams). The described method can be applied in many different areas in order to analyse and compare the changes in numeric variables. For example, the stock prices, sale quantities of products, health status indicators of patients who are under similar treatments, cost and profit metrics, etc., can be analysed and compared using the proposed time-correlation detection method.

A time series (Fayyad et al., 1996) is formally defined as a sequence of data values that are recorded in one of the following manners:

- at equal time intervals
- or at random points in time by also recording the time of recording.

The method described in this paper uses the second definition in order to provide a generic approach that can be applied to both periodically recorded time series data and time series data that is recorded with random time intervals.

Time-correlations are defined as time-delayed relationships among multiple time series data streams such that a change in the values of one set of time series data streams is related to a change in the values of another set of time series data streams. Since each time series data stream contains the measured values of an individual numeric variable at different points in time, the time-correlations detected through the comparative analysis of multiple time series data streams actually tell us the time-delayed relationships among numeric variables which are represented by those data streams.

When looking for time-correlations, it is necessary to consider a few factors that may determine the way in which one time series may affect another one. Figure 1 shows a few time series examples among which certain relationships exist. The reader is requested to note that this is just an example and the relationships depicted in that figure may not apply to all business cases. As a first example, server performance has a time-delayed impact on database performance that can be easily observed in the figure. This relationship is easy to observe on the figure because the amount of impact on database performance is the same as the amount of change in server performance. As another example, the number of orders has an immediate impact (i.e., no time delay) on average time per activity (average processing time for each activity in the organisation that is
related to handling of received orders). It can also be observed in the figure that a large change in number of orders causes a smaller impact on average time per activity. That means, although average time per activity is affected by number of orders, its sensitivity to changes in number of orders is low. As a final example, let us consider the relationship between the average time per activity and percentage of on-time order delivery. The average time per activity has an immediate affect on percentage of on-time order delivery in the opposite direction. That means, when there is an increase in average time per activity, a decrease occurs in percentage of on-time order delivery and vice versa. In summary, the main factors that we consider while analysing the relationships among time series data are time-delay, sensitivity and direction. We also measure confidence as an indicator of how certain we are about each time-correlation.

Figure 1  Time series examples among which certain relationships exist (see online version for colours)

The main contributions of this paper include the following:

- It is shown that time series data can be analysed efficiently in time domain without having to transform it into another domain. The most popular approach so far was to transform data into frequency domain for fast analysis (Agrawal et al., 1993; Berndt and Clifford, 1996; Faloutsos et al., 1994; Goldin and Kanellakis, 1995; Yi et al., 1998; Zhu and Shasha, 2002), which causes loss of information such as time delays and many characteristics of the original data.
- Detected time-correlation rules are digitally stored for further analysis.
- Textual descriptions of the rules are generated in order to explain detected time-correlations to non-technical users.
- Time series data is aggregated and analysed at multiple time granularities at the same time.

The rest of this paper is organised as follows. Section 2 describes the time-correlation detection method. Performance evaluation is explained in Section 3. Section 4 summarises related work and Section 5 contains the concluding remarks.
2 Time correlation detection

2.1 Summary of the method

The proposed method for detecting time-correlations from time series data streams consists of the following steps, as shown in Figure 2:

- summarising the data at different time granularities (e.g., seconds, minutes, hours, days, weeks, months and years)
- detecting change points (points at which significant changes in data values are recognised)
- generating time-correlation rules
- merging multiple time series data to generate merged time series data
- comparing time series data to generate time correlation rules.

The input of the method is any collection of numeric data streams that are time-stamped (i.e., time series data streams). The input data can be read from one or more database tables, XML documents, flat text files with character delimited data fields or directly from data streams. The output is a set of time-correlation rules that describe time-delayed relationships among time series data streams. Each time-correlation rule includes information about:

- **Direction:** Direction is ‘same’ if the change in the value of one time series data is correlated to a change in the same direction for another time series data; ‘opposite’ if the change direction is opposite.

- **Sensitivity:** The magnitude of change in data values in two correlated time series are recorded in order to indicate how sensitive one time series is to the changes in another time series.
• **Time delay:** The time delay for correlated time series data streams are recorded in order to explain how much time delay exists between two related (matching) changes in the values of two or more correlated time series data streams.

• **Confidence:** Confidence provides an indication of how certain we are about detected time-correlations among time series data streams. Confidence is measured as a value between zero and one. A confidence value that is close to one indicates high certainty. Similarly, a confidence value that is close to zero indicates low certainty.

Figure 3 shows the overall architecture of the prototype implementation, called *intelligent program for event correlation* (IPeC). The prototype receives time series data streams as its input and generates time-correlation rules, textual rule explanations and stores the generated rules digitally to enable their reuse in other data analysis tools.

The described method applies data aggregation and change detection algorithms in order to summarise data at a proper granularity level and reduce the search space. Data mining and statistical techniques are used for statistical correlation calculation, data aggregation and change detection. Sensitivity analysis is embedded inside change detection and correlation rule generation steps. Sensitivity analysis is achieved through recording the change amounts while detecting change points and comparing those change amounts from different time series data streams while generating the correlation rules.

**Figure 3** Overall architecture of IPeC prototype implementation (see online version for colours)

<table>
<thead>
<tr>
<th>Correlation Engine</th>
<th>Rule Explanations</th>
<th>Rule Re-use Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Detection</td>
<td>Data Mining &amp; Statistics</td>
<td>Automatic Aggregation</td>
</tr>
<tr>
<td>Time-series data streams</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rule explanations are textual explanations of detected rules. For example, assume that Table 1 shows a database table containing a simplified digital view of a time-correlation rule and skipping the details about sensitivity information (change magnitude) for simplicity:

**Table 1** Example database table that shows the digital storage of the generated time-correlation rules

<table>
<thead>
<tr>
<th>Col1_name</th>
<th>Col2_name</th>
<th>Time_difference</th>
<th>Time_unit</th>
<th>Confidence</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health of resource X</td>
<td>Response time of activity Y</td>
<td>1</td>
<td>Hour</td>
<td>0.6</td>
<td>Negative</td>
</tr>
<tr>
<td>Response time of activity Y</td>
<td>Violations of SLA S1</td>
<td>0</td>
<td>Hour</td>
<td>0.7</td>
<td>Positive</td>
</tr>
</tbody>
</table>
Textual explanation of this rule can easily be generated in the following form:

If health of resource X decreases more than 5%,
then response time of activity Y increases more than 10% after one hour.

The structure of the time-correlation rule resembles that of an association rule in data mining domain. A typical association rule indicates an association among two or more data objects. A time-correlation rule indicates a time-delayed relationship among the changes in the values of two or more numeric data objects. A time-correlation rule provides much more information than an association rule and it is significantly more difficult to calculate because of the additional complexity introduced by the inclusion of time domain in the rule and the sensitivity and direction analyses. An association rule only tells us that a group of data objects are associated such that the existence of certain values in some data objects of the rule (i.e., antecedent or the ‘if’ part) implies the existence of certain values in the other data objects (i.e., consequent or the ‘then’ part). An association rule does not provide much detail about the relationship among its data objects, except for confidence and support values which only explain the statistical significance of the rule. On the other hand, a time-correlation rule provides additional metrics (i.e., time delay, sensitivity and direction), as well as the confidence and support, that provide actionable information for numeric data objects. It is possible to define a time-correlation rule as a ‘time-lagged association rule with sensitivity and direction information’ in data mining domain. The proposed method and IPEC prototype implementation demonstrate how to extract time-correlation rules efficiently from numeric time series data streams.

The digital storage of the generated time-correlation rules in database tables or XML files makes it possible to feed those rules into various analysis tools, such as simulation, forecasting, etc. As an example, Figure 4 shows how the generated time-correlation rules can be reused for simulation.

Figure 4 indicates that variables X and Z affect the variables Y and W respectively. The reuse of those rules in a simulation is carried out as follows:

1. Time-correlation rules are detected.
2. The rules are written into a rule database.
3. User of the simulation tool makes changes to variables X and Z.
4. The simulation tool retrieves the rules from the rule database in order to assess the expected impact of the changes in variables X and Z.
5. The simulation tool finds out from time-correlation rules that variables Y and W are going to be impacted because of the changes in the values of variables X and Z. The time-correlation rules include information about direction, sensitivity and time-delay of the impact so that the simulation tool can figure out exactly when, in which direction and how much the impact is going to be observed. The confidence values of time-correlation rules tell the simulator the probability of observing an impact.
2.2 Summarising the data at different time granularities

The first step of time-correlation detection method is data aggregation. Time-stamped numeric data values (time series data) need to be summarised for two main reasons:

1. Volume of time series data is usually very large. Therefore, it is more time efficient to summarise the data before analysing it. For example, if there are thousands of data records for each minute, it may be more time efficient to summarise the data at minute level (e.g., by calculating mean, count and standard deviation of recorded values), so that the data can be analyse faster.

2. Time-stamps of different data streams may not match each other at fine granularities. That makes it difficult to compare time-stamped data with each other. For example, Table 2 shows an example database table that contains three time series data streams for the grades of a high school student: Math, Physics and English. The table does not contain comparable time stamps at the ‘day’ granularity level because all exam scores were recorded on different days. Summarising the scores by month is not enough because each month of the year does not contain a recorded value for every time series (i.e., for every course). Consequently, sometimes it may be necessary to summarise data using higher time granularity so that the numeric data from different data streams can be compared with each other.

The proposed method works in existence of missing data, but sometimes it may still be desired to summarise the data at higher granularities so that it is semantically comparable (e.g., at year level to compare students’ relative progress).

Table 2 Example database table containing time series data

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Time-stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>85</td>
<td>January 12, 2002</td>
</tr>
<tr>
<td>Physics</td>
<td>93</td>
<td>January 26, 2002</td>
</tr>
<tr>
<td>English</td>
<td>74</td>
<td>February 20, 2002</td>
</tr>
<tr>
<td>Math</td>
<td>96</td>
<td>March 23, 2002</td>
</tr>
<tr>
<td>Physics</td>
<td>81</td>
<td>April 2, 2002</td>
</tr>
<tr>
<td>English</td>
<td>65</td>
<td>April 5, 2002</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Math</td>
<td>97</td>
<td>January 10, 2003</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Figure 5 shows an example of how data aggregation can be done at a particular time granularity level. The aggregation is performed by calculating the sum, count, mean, minimum, maximum or standard deviation of individual data values within each time unit. The figure shows the mean value calculation, which is equal to the sum of values divided by the count of values in each time unit.

![Figure 5](image)

Moving windows are used for incrementally aggregating the data in increasing granularity levels. Figure 6 shows how the data aggregation is incrementally propagated from lower levels to higher levels so that the raw time series data can be aggregated at multiple granularity levels at the same time. For example, assume that the desired lowest level of aggregation is minute level. The data will be aggregated at the minute level as soon as it arrives. The moving window at the minute level contains the aggregated data by each minute. Assume that the data value at time $t$ is aggregated into the minute bucket $m$ in the minute level moving window. When a new data value that falls into minute bucket $m + 1$ arrives, the aggregate data in bucket $m$ is propagated to the upper aggregation level, say hour level. Bucket $m$ is moved and new coming data points start aggregation into bucket $m + 1$ from this point on. Similarly, the data aggregated at any granularity level is propagated to the upper granularity level as soon as the newest data point falls into a new aggregation bucket at that level.

![Figure 6](image)

Sometimes, the time delay of correlation between data streams may be shifted slightly in the time domain. The experiments showed that the time shift is not very large and 99% of the time, it is observed within one time unit difference. In order to capture those cases, the time-correlation detection method uses a fuzzy moving window of three time units. For example, aggregation of data values in the ‘hour’ granularity level involves the current hour as well as a fraction of the previous and next hours. As a result, slight shifts
in the time domain can be incorporated in aggregated values. Some of the existing methods use complex and costly algorithms for time-shifting or time-warping in order to handle such cases (Berndt and Clifford, 1996). The experiments showed that time-shift is almost never larger than one unit difference; therefore, such complicated algorithms do not provide much advantage for time-correlation detection.

2.3 Detecting change points

The second step of time-correlation detection method is change point detection (identification of points in time where significant changes occur in data values or their trends). While the definition of change point may differ depending on the particular area of application (Staudacher et al., 2005), certain generic methods, such as cumulative sum (CUSUM), are useful in several areas. CUSUM (Page, 1954a, 1954b) is preferred at this step because it can be calculated incrementally and it can detect both sharp and gradually accumulating changes in data values. CUSUM is calculated as follows at each data point:

1. subtract the mean (or median) of the data off of each value in the input data stream
2. add all the mean/median-subtracted points before each data point to obtain the CUSUM value.

It is necessary to modify the basic CUSUM method in order to adapt the CUSUM and mean values to changing data as time series data streams continuously flow in. Two possible approaches are sliding window and aging mechanism. Sliding windows have the limitation that the choice of window size has a significant effect on the accuracy. Therefore, aging mechanism is preferred. Aging is achieved by combining a fraction of the new data value with a fraction of the last calculated value (i.e., last CUSUM or last mean) as follows:

\[ C(i) = p \times V + (1 - p) \times C(i - 1) \]

where \( C(i) \) is the new calculated value, \( C(i-1) \) is the last calculated value, \( V \) is the latest data value from the time series data stream and \( p \) is the parameter that determines the fractions of new data value and last calculated value that will be added to obtain the new calculated value. The value of \( p \) is chosen as a floating point number between zero and one. Large values of parameter \( p \) result in fading away the effect of old data values quickly; whereas, small values of parameter \( p \) result in slower fade away. This aging mechanism is sometimes called ‘exponential decay’ in literature.

Once CUSUM value for every data point is calculated, the calculated CUSUM values are compared with upper and lower control thresholds to determine which data points can be marked as change points. The data points for which the CUSUM value is above the upper threshold or below the lower threshold are marked as change points. The best way to set thresholds is using a fraction or factor of standard deviation. It is easy to calculate a moving mean and standard deviation using a moving window or aging. Therefore, the calculation of standard deviation does not introduce a large overhead in the proposed method. The thresholds are generally set as to three times the standard deviation in order to detect extreme changes. We use only one standard deviation to set the thresholds, because we would like to detect all noticeable change points, not only the extreme ones.
Detected change points are marked with one of the labels, indicating the direction of change that is detected: Down (trend of data values change from up or straight to down) or Up (trend of data values change from down or straight to up). In addition to direction, the amount of change is recorded for each change point. The amount of change is used in sensitivity analysis.

The start and end points of a change always carry some uncertainty as it is indicated in Saidane and Lavergne (2008). The proposed method reports the exact start and end boundaries for each change event, but also enables fuzzy processing of those boundary points in order to make sure that a slight inaccuracy in determination of those points does not affect the outcome of the following steps significantly.

2.4 Generating time-correlation rules

Time-correlation detection method first reduces many-to-one and many-to-many time series comparisons into pair-wise (one-to-one) time series comparisons. Then, the problem of comparing multiple time series data streams can be tackled efficiently and easily. In order to explain the reduction and comparison steps of the method, it is first necessary to explain what is meant by one-to-one, many-to-one and many-to-many time series comparisons:

- **One-to-one**: Comparison of two time series data streams with each other. For example, if A and B identify two time series data streams, one-to-one comparison tries to find out if changes in data values of A have any time delayed relationship with changes in data values of B.

- **Many-to-one**: Comparison of multiple time series data streams with a single time series data stream. For example, if A, B and C identify three time series data streams, many-to-one comparison tries to find out if changes in data values of A and B collectively have a time delayed relationship with changes in data values of C.

- **Many-to-many**: Comparison of multiple time series data streams with multiple time series data streams. For example, if A, B, C and D identify four time series data streams, many-to-many comparison tries to find out if changes in data values of A and B collectively have a time delayed relationship with changes in data values of C and D.

Many-to-many comparisons do not have any practical use because their results can easily be derived from results of many-to-one comparisons. Time-correlation detection method first describes how to reduce many-to-one and many-to-many time series comparisons into one-to-one time series comparisons. Then, it explains how to perform one-to-one time series comparison in order to extract time correlation rules.

2.5 Merging multiple time series data streams into one

The purpose of merging multiple time series data streams into one is to be able to compare multiple time series data streams with each other in one pass. This yields performance improvement because the merged time series data streams can be reused, similar to the way query results can be reused in database management systems in order to provide performance improvement. For example, after merging two time series data
streams A and B, the merged time series data stream can be stored in order to generate higher order merged time series data streams, such as A*B*C, where the symbol '*' is used for indicating the merge operation.

Time-correlation detection method uses convolution for merging multiple time series data streams into one. Convolution is a well-known computational method in which an integral expresses the amount of overlap of one function $g(x)$ as it is shifted over another function $f(x)$. Convolution may therefore blend one function with another. Convolution of two functions $f(x)$ and $g(x)$ over a finite range is given by the equation:

$$f * g = \int_0^\infty f(\tau)g(t-\tau)d\tau$$

where $f * g$ denotes the convolution of $f$ and $g$.

Since the merge operation is applied on discrete time series data streams (after using CUSUM to detect change points), the operation can be performed much faster than merging the time series data streams in their original numeric values.

2.6 Comparing two time series

Time-correlation detection method uses statistical correlation to calculate the time correlation between two time series data streams. The time series data streams that are compared at this step may correspond to either merged time series or regular (original) time series. The statistical correlation between two time series is calculated as:

$$\text{cor}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma(X)\sigma(Y)}$$

where $X$ and $Y$ identify two time series and covariance is given by

$$\text{cov}(X,Y) = E[\{x_i - E(X)\}[y_i - E(Y)]] \quad \forall i, j \in t, i = j$$

where $\sigma(X)$ corresponds to the standard deviation of values in time series $X$, $\sigma(Y)$ corresponds to the standard deviation of values in time series $Y$, $E(X)$ and $E(Y)$ correspond to the mean values of time series data values from X and Y, $t$ corresponds to aggregated time span of the time series data (e.g., minutes, hours, days, etc.), and $x_i$ and $y_j$ represent the individual data values from two time series X and Y.

We modify the statistical correlation and covariance formulas to calculate time-correlation which incorporates the time distance:

$$\text{Time correlation} = \max \{\text{cor}(X,Y,d)\}, d \in [0,1,2,...]$$

$$\text{cor}(X,Y,d) = \frac{\text{cov}(X,Y,d)}{\sigma(X)\sigma(Y)}$$

$$\text{cov}(X,Y,d) = E[\{x_i - E(X)\}[y_i - E(Y)]] \quad \forall i, j \in t; |i-j| = d$$

where $d$ is a positive integer indicating the time distance that yields the largest correlation for two time series $X$ and $Y$.

Sensitivity is calculated using the following formula:

$$\text{change}(x_i)/\text{change}(y_j) \text{ where } i, j \in t; |i-j| = d$$
by setting the distance \((d)\) between \(i\) and \(j\) to that of the maximum statistical correlation found and comparing the magnitude of change in correlated time series data streams. In other words, the statistical correlation between aggregated data points with varying time distances are calculated and the maximum calculated correlation and the corresponding time distance \((d)\) give us the time correlation between the compared time series data streams. The sensitivity is calculated by comparing the change amounts in the correlated time series data streams using time distance \((d)\) of the calculated maximum statistical correlation. The direction of correlation is obtained from the calculated statistical correlation. A positive correlation indicates the direction is ‘same’ whereas a negative correlation indicates the direction is ‘opposite’. The values \(\text{change}(x_i)\) and \(\text{change}(y_j)\) represent the change amounts detected at data points \(x_i\) and \(y_j\) in data streams \(X\) and \(Y\) respectively. Those change amounts were calculated in an earlier step of the proposed method (i.e., in change detection step). Thus, we are comparing only the detected change points from two time series data streams with each other, as opposed to comparing all data values from two streams and this yields significant performance improvement both in terms of running time and the storage space requirement of the method. Figure 7 summarises an important part of the comparison step in the proposed method, where the change points of two time series are compared with each other to determine the distance \((d)\) that yields the highest correlation value.

Figure 7  Partial pseudo-code of the comparison step where the distance \((d)\) that yields the highest correlation is determined

```plaintext
bestCorrelation = 0;
bestDistance = 0;
input maximumDistance from user;
for d=0 to maximumDistance do
    correlation = cor(X,Y,d);
    if (correlation > bestCorrelation)
        bestCorrelation = correlation;
        bestDistance = d;
    end if;
end for;
```

The most challenging task in this comparison step is to determine the time distance \((d)\) for which the calculated correlation is the highest. An exhaustive search in which all possible time distances are tried and maximum correlation is determined is prohibitive due to performance reasons unless there exists an upper limit for the distance \((d)\) that is set by the user (see Figure 7). Otherwise it is necessary to find the distance \((d)\) in a faster way. The described method uses sampling in order to first determine which time distances are likely to return a high correlation between the time series data streams that are being compared. Then, the actual correlation is calculated for a few of those candidate distances and the one with the highest correlation is selected. Although the number of candidate distances considered has a significant effect on the accuracy of the result, the experiments showed that after sampling the data, it is enough to consider at most three or four candidate distances to find the highest correlation distance for 95% of the time.
Once the time-correlation is determined, the confidence can be calculated easily by comparing the percentage of times the calculated statistical correlation with the time delay (d) of the maximum correlation is higher than a predefined threshold.

Comparison of all possible combinations of time series data streams is computationally prohibitive. If a pair (or group) of time series data streams are time-correlated, they should have similar number of change points detected in the previous steps. This is an important feature of time-correlations, which can also be used for speeding up the proposed method significantly. The proposed method compares only those time series data streams that have similar number of change points detected. As a result, an important performance gain is achieved without causing false dismissals (i.e., without discarding the comparison of data streams that may actually be time-correlated).

3 Performance evaluation

Performance evaluation is done using IPEC prototype implementation and four data sets:

- Continuous data streams of stock quotes for 2000 NYSE stocks (i.e., 2000 data streams, each of which contains one stock quote per second).
- Real-time data from 2000 sensors that measure the temperature at different points of a server cluster system at Hewlett-Packard (recorded every second).
- Workflow execution logs for financial operations. This data set included workflow execution logs that record the start and end times of individual steps in business process executions and various parameters related with the business processes. The data rate for this data set fluctuates depending on the time of the day and there exist certain peak load periods. The processing delay of the proposed method was within the range that was measured for other three data sets, even during the peak load periods.
- Synthetic data generated using random walk model that contains 20,000 data streams, each producing one numeric data sample per second. The number of data streams in this set is much larger than those of the other three sets. The results obtained using this data set were compared with the results from the other data sets for up to 2,000 data streams. After that, this data set was used for modelling further experiments in which a larger number of data streams (i.e., 20,000) is used.

The experiments with all four data sets were repeated several times and each run of the experiments involved at least 30,000 data records from each data stream. The experiments showed that the proposed method extracts time-delayed correlations (or temporal associations) within a small time delay while the data flows in continuously from a large number of data streams.

The accuracy of the algorithm was compared against that of exhaustive search algorithm. It has been shown that 95% of time-correlations that are detected by the exhaustive search are also detected by the proposed method using reasonable parameters.
Figure 8 Logarithmic value of processing delay compared with number of data streams (as number of data streams increase by an order of magnitude each time) (see online version for colours)

Figure 8 shows the logarithmic value of processing delay (the time between receipt of each data record and maximum delay for completing comparative analysis for the data record, plus the total filtering and reporting time for correlation rules) compared with number of data streams, which is increased ten times at each run of the experiments. Processing delay is used as the performance measure instead of total running time of the method, because data continuously flows in and it is not practical to measure the total running time for such data stream analysis. Previous research on data stream analysis also used processing delay as the performance measure (Zhu and Shasha, 2002). The figure shows a linear increase in logarithmic values of processing delay compared with increasing order of magnitude in number of data streams, which suggests at a first glance that the comparison of actual measurements would not yield a linear increase in processing delay. However, the x-axis values increase ten times at each data point in the figure. It can be seen that the processing delay would actually increase linearly if the number of data streams increased linearly (instead of ten times). This linear increase is demonstrated more clearly in Figure 9. The small jump between 200-2,000 data stream tests in Figure 8 is due to the fact that the processing delay up to 200 data streams was hidden behind the arrival rate of the data. That means, the processing delay of each incoming data value was much shorter than the time between data value arrivals for the whole set of data streams. The proposed method can process incoming data values from up to 200 data streams in less than a second and the time measured as the delay in Figure 8 is only due to filtering of final correlation rules based on support and confidence thresholds.

Experiments showed that the proposed method has linear running time for pair-wise correlations (i.e., total execution time, not only the processing delay) with respect to the total number of data points that exist in all data streams. The running time of higher order correlations is not linear, but the method still requires only a single pass over the input data. The results of further experiments are not included here due to space limitation.
4 Related work

To the best of the authors’ knowledge, the time-correlation detection method described in this paper is the first research effort that can be used for detecting time-delayed relationships among time series data streams (numeric data streams with timestamps) in time domain (without transforming data into another domain, such as the frequency domain). There exists no other method that can report direction, sensitivity and time-delay of correlations among time series data streams.

Previous research on similarity based pattern querying can be considered somewhat related to the method described in this paper. However, those existing methods can only tell us whether given time series query is similar enough to existing time series subsequences stored in a database and cannot report any information about the direction, sensitivity or time-delay of the relationship among time series data streams. Most of the existing research in this field used discrete fourier transform (DFT) in order to transform the time series from time domain into frequency domain (Agrawal et al., 1993; Berndt and Clifford, 1996; Faloutsos et al., 1994; Goldin and Kanellakis, 1995; Yi et al., 1998; Zhu and Shasha, 2002). Those approaches have a significant limitation that similarity model is different from the data representation (Perng et al., 2000). This limitation is the main reason why those approaches cannot be applied for time-correlation detection (detection of time-delayed relationships).

In order to consider the fact that sometimes the time series data streams may have different time domain scales or data value magnitude ranges, etc., the similarity models have been extended to consider various transformations on the data. A few examples are warping (Berndt and Clifford, 1996; Rafiei and Mendelzon, 1997; Shatkay and Zdonik, 1996; Yi et al., 1998), amplitude shifting (Chu and Wong, 1999; Goldin and Kanellakis, 1995; Shatkay and Zdonik, 1996) and allowing time series segments of different amplitude scales to be considered similar (Agrawal et al., 1995; Chu and Wong, 1999; Das et al., 1997; Goldin and Kanellakis, 1995). Such complicated transformations are useful in similarity search, but they are unnecessary in time-correlation detection because...
time delay and magnitude change need to be detected and recorded rather than being ignored.

Based on the observation that humans can ignore small fluctuations and apply smoothing when comparing time series data visually, Rafiei et al proposed a method based on smoothing and noise reduction using moving averages (Rafiei and Mendelzon, 1997). The aggregation algorithm described in this paper has an advantage over the previous algorithms because it allows aggregation of data in multiple granularity levels.

Perng et al. (2000) suggested a new method that first tries to identify landmarks, i.e., important points in time when certain events happen and then using six different transformations to confirm similarity of time series data streams. The paper by Perng et al did not explain any details about how to implement those transformations, but it presented an interesting idea that detection of important points can be useful for similarity search. The time-correlation detection method described in our paper builds on top of the idea of identifying important points and uses those important points to detect time-correlations among time series data streams efficiently.

5 Conclusions

A single-pass method for detecting time-correlations in time series data streams is described. The method consists of a few main steps in which the original time series data is summarised using aggregation; converted into discrete data using a well-known change detection technique; sampled for fast identification of candidate time distances and scanned in linear time for confirming the candidate correlation rules. Time-correlation rules generated by the described method can easily be reused for analysing transactional or operational data in various business environments. Moreover, the detected rules can be converted into textual representations in order to explain those rules to non-technical users easily.

References


Drawing reasonable conclusions from information under similarity modelled contexts

Ronald R. Yager
Machine Intelligence Institute,
Iona College,
New Rochelle, NY 10801, USA
E-mail: yager@panix.com

Abstract: We are interested in the process of making reasonable conclusions about the value of a variable. We indicate that reasonableness generally depends on the information we have about the variable as well as the context in which we shall use the assumed value. In order to include a wide range of imprecise and uncertain information, we use granular computing technologies such as fuzzy sets, Dempster-Shafer belief structures and probability theory to represent our knowledge and conclusions. While context is a very diverse idea, in order to provide some structure, we restrict ourselves to the special case where context can be modelled using a similarity relationship. Within this framework, we suggest a measure of the reasonableness of drawing conclusions from information in the context of a similarity relationship. We look at the properties of this measure and investigate its performance in a number of special cases.

Keywords: granular computing; context; similarity relationship; reasonable assumptions; drawing conclusions; conjecturing; uncertainty reduction.


Biographical notes: Ronald R. Yager is the Director of the Machine Intelligence Institute and a Professor of Information Systems at Iona College. He has worked in the area of machine intelligence for over 25 years and published over 500 papers. He is among the world’s top 1% most highly cited researchers. He is a recipient of the IEEE Computational Intelligence Society Pioneer award in Fuzzy Systems and a fellow of the IEEE, the New York Academy of Sciences and the Fuzzy Systems Association. He served at the National Science Foundation as a Program Director. He is the Editor-in-chief of the International Journal of Intelligent Systems and serves on the editorial board of numerous technology journals.

1 Introduction

A pervasive process in our modern information centred world is the gathering of information about a variable of interest from various sources and using this information for our benefit. This information can either be precise or the type of soft imprecise granular information characteristic of linguistic information. The information can be
certain or have an uncertainty associated with it. Fuzzy sets, Dempster-Shafer (D-S) and other related granular computing technologies (Lin et al., 2002; Bargiela and Pedrycz, 2003; Pedrycz et al., 2008) provide a compatible suite of technologies in which we can represent information that can have uncertain granular aspects. In many situations in which we have this information, we may need to alter the information to make it more useful for our intended use. These alterations may be motivated by various purposes. One example is in the presentation of the information to an end user who may benefit by our expressing the information using terms that are compatible with their vocabulary. For example the information that John’s age is between 14 and 18 may be better expressed by saying he is a young man.

Other examples of the use of alteration are the formulation of an opinion, making an assumption, generating a conjecture or hypothesis, making an inference or drawing a conclusion on the basis of the available information. These are particularly important processes in human reasoning and decision making. In the following we shall generically refer to these processes as drawing a conclusion.

Generally, alterations that go from more precise to less precise as well as those that do not increase our certainty in an uncertain situation pose no problems. However, the real benefit of altering our information comes when we go in the opposite direction, from less precise to more precise, from less certain to more certain. Almost every decision we make involves reducing uncertainty and resolving imprecision. While these types of information enhancing alterations cannot be done without some risk, they are required and pervasive in the real world and must be done in a reasonable way.

Let us look at this process of drawing a conclusion in a little more detail. Let V be a variable taking its value in the space X. Consider that the process of assuming that V is some value G. This assumption must be reasonable with respect to our available information. For example saying John is a teenager when we know his age is over 50 is clearly unreasonable. Another notion that plays a role in determining the reasonableness of assuming that \( V = G \) is the context in which this assumption will be used. While the concept of context is very complicated in some cases, the relevant aspects of context can be captured with the use of a similarity relationship. For example, if we are telling a person’s information about the temperature in the context of their deciding which jacket to wear, we can assume the existence of a similarity relationship such that all temperatures in the 80’s are equivalent.

Our objective here is to investigate this issue of determining the reasonableness of concluding a value for a variable in the situation in which we have some information about the variable as well as some information about the context in which the conclusion will be used expressed in terms of a similarity relationship.

2 Soundness of altering information

In this section, we review the idea of the soundness of altering the value of a variable introduced in Yager (2007). We shall see that this can be used as the basis for a measure of the reasonableness of making a conclusion about the value of a variable.

Let V be a variable taking its value in the space X. Assume our knowledge of \( V = A \) that V is A, where A is a fuzzy subset of X. It is well known that if B is such that \( A \subseteq B \), \( A(x) \leq B(x) \) for all x then from \( V = A \) we can infer that \( V = B \). Zadeh (1979a) referred to this as the entailment principle. Thus in this case, altering \( V = A \) to \( V = B \) can be seen as...
a sound alteration. For example, the knowledge that John is between 18 and 25 years old allows us to infer that he is more than 15 years old. However, in general, we cannot infer that he is 20 years old.

Yager (2007) investigated the issue of the soundness of inferences in the face of similarity relations. We first recall a similarity relation (Zadeh, 1971; Finnie and Sun, 2002) on the space $X$ in a mapping $S: X \times X \rightarrow [0, 1]$ that has the following properties:

1. reflexivity: $S(x, x) = 1$
2. symmetry: $S(x, y) = S(y, x)$.

The larger $S(x, y)$, the more similar the two elements.

We now recall the approach used by Yager (2007) to determine the soundness of altering $V$ is $A$ to $V$ is $B$ in the face of a similarity relationship. What will become apparent is that this measure of soundness is effectively a measure of the reasonableness of assuming $V$ is $B$ given the knowledge that $V$ is $A$.

We first introduce the concept of concordance of $A(x)$ with $B(y)$ in the context of a similarity relationship $S$. We denote this as $\text{Con}_S(A(x), B(y))$, it is defined as:

$$\text{Con}_S(A(x), B(y)) = (1 - \text{Max}(0, A(x) - B(y))) \land S(x, y)$$

We now define the concordance of $A(x)$ with $B$ which we express as:

$$\text{Con}_S(A(x), B) = \text{Max}_y [\text{Con}_S(A(x), B(y))]$$

Finally, we use this to determine the soundness of altering $V$ is $A$ to $V$ is $B$ in context of the similarity relationship $S$:

$$\text{Sound}_S(B / A) = \text{Min}_x [(1 - A(x)) \lor \text{Con}_S(A(x), B)]$$

It is shown that $\text{Sound}_S(B / A) \in [0, 1]$ and as indicated in Yager (2007), the larger its value the more appropriate the replacement of $V$ is $A$ by $V$ is $B$.

It can be easily shown that for any $S$ if $B_1 \subseteq B_2$, $B_1(x) \leq B_2(x)$, for all $x$ then:

$$\text{Sound}_S(B_1 / A) \leq \text{Sound}_S(B_2 / A)$$

Thus, as we increase $B$ we can increase its soundness.

The following important result was shown in Yager (2007), if $A \subseteq B$ then $\text{Sound}_S(B / A) = 1$ for any $S$. This is consistent with the concept of entailment or inference (Zadeh, 1979a; Anderson and Belnap, 1975; Yager and Kreinovich, 2007). It is always sound go from more precision to less precision.

As we just noted given $V$ is $A$ it is logically correct to infer $V$ is $B$ in the case $A \subseteq B$. As we shall consequently make clear our measure $\text{Sound}_S(B / A)$ can be seen as a softening and stretching of the idea of the concept of containment of $A$ in $B$. Here the condition $A(x) \leq B(x)$ is essentially softened by using the formation $(1 - \text{Max}(0, A(x) - B(x)))$ to measure the degree of inclusion. Here we see that if we have the condition $A(x) \leq B(x)$, $A(x) - B(x) \leq 0$, our formulation results in a value of one. Thus, this measures captures the case in which $A(x) \leq B(x)$. On the other hand, if $A(x) > B(x)$ instead of reporting a value of truth of zero we obtain a softer value,
$1 - (A(x) - B(x))$. Here, the more $A(x)$ is bigger than $B(x)$, the smaller the reported degree of inclusion.

The second modification to the crisp requirement that $A(x) \leq B(x)$ obtained by using $\text{Con}_S(A(x), B(y))$ is that for every $x$ we consider $A(x) \leq B(y)$ for values of $y$ ‘similar’ to $x$. So we are stretching our idea by allowing ‘near by’ values to be considered as covering $x$. We see then that $\text{Sound}_S(B/A)$ can be viewed as the reasonableness of concluding $V$ is $B$ given $V$ is $A$ in the context of $S$.

There are two notable similarity relations $S_*$ and $S^*$. These are defined respectful as $S_*(x, y) = 0$ for $x \neq y$ and $S_*(x, x) = 1$ and $S^*(x, y) = 1$ for all $x$ and $y$. It is shown that there are the extreme cases of similarity relations, for all $x$ and $y$, $S_*(x, y) \leq S(x, y) \leq S^*(x, y)$. We see that $S_*$ is actually the case where no similarity is assumed, all the elements are completely distinct. The other extreme $S^*$ is when there is complete similarity between all elements, everything is the same.

Consider first the case of $S^*$. Let us look at:

$\text{Sound}_{S^*}(B/A) = \min_x [(1 - A(x)) \vee \text{Con}_{S^*}(A(x), B)]$.

We first recall that:

$\text{Con}_{S^*}(A(x), B(y)) = (1 - \max(0, A(x) - B(y)))$

Let $z$ be such that $B(z) = \max_y [B(y)]$, it is the element in $X$ with the largest membership grade in $B$. What is clear is that:

$\text{Con}_{S_*}(A(x), B(z)) = \max_y [\text{Con}_{S_*}(A(x), B(y))] = \text{Con}_{S_*}(A(x), B(z))$

If $B(z) = 1$, $B$ is normal, then $\text{Con}_{S_*}(A(x), B(z)) = 1$ and we get $\text{Sound}_{S_*}(B/A) = 1$. Thus, every alteration is reasonable. Consider the case with $B(z) \neq 1$. Let $a$ be the maximal membership in $A$. Then we can show that $\text{Sound}_{S_*}(B/A) = (1 - a) \vee (1 - (0 \vee (a - B(z))))$. Thus, if $B(z) \geq a$ then the soundness is still one.

Consider now the case where $S = S_*$, here:

$\text{Con}_S(A(x), B(x)) = (1 - \max(0, A(x) - B(x)))$

While for $x \neq y$:

$\text{Con}_S(A(x), B(y)) = 0$.

In this case,

$\text{Sound}_S(B/A) = \min_x [\overline{A(x)} \vee (1 - \max(0, A(x) - B(x)))]

Let $X_m$ be the subset of $X$ for which $B(x) \leq A(x)$ then:

$\min_{x \in X_m} [\overline{A(x)} \vee (1 - A(x) + B(x))] = \min_{x \in X_m} [1 - ((A(x) - B(x))] = 1 - \max_{x \in X_m} [A(x) - B(x)]$

It is simply the negation of the biggest value by which $A(x)$ exceeds $B(x)$. Since for those $x$ for which $B(x) > A(x)$ we have $\overline{A(x)} \vee (1 - \max(0, A(x) - B(x))) = 1$ then we get:

$\text{Sound}_{S^*}(B/A) = 1 - \max_{x \in X} [(A(x) - B(x)) \vee 0]$
A special case of similarity relation is an equivalence relation. We recall an equivalence relation is a binary similarity relation $S(x, y) \in \{0, 1\}$ that is transitive, if $S(x, y) = S(y, z) = 1$ then $S(x, z) = 1$. An equivalence relation partitions $X$ into disjoint crisp subsets $E_1$, $E_2$, ..., $E_r$, called equivalence classes such that $E_i \cap E_j = \emptyset$ and $\cup E_j = X$. The equivalence classes are defined such that $x$ and $y$ are in the same class if $S(x, y) = 1$.

In Yager (2007) it was shown in the context of an equivalence relationship $S$: $$\text{Sound}_{\mathcal{S}}[B / A] = \min_{j=1}^{r} \left[ 1 - \max(0, A(z_j) - B(y_j)) \right]$$

where $A(z_j) = \max_{z \in E_j} [A(z)]$ and $B(y_j) = \max_{y \in E_j} [B(y)]$. Here then $z_j$ is the element in $E_j$ with the largest membership grade in $A$ and $y_j$ is the element in $E_j$ with the largest membership grade in $B$.

What is significant about the case of equivalence relationship is that we can decompose the problem and look at each equivalence class separately. In Yager (2007) it was shown that if equivalence class $j^*$ is such that $A(z_{j^*}) - B(y_{j^*}) = \max_{j} [A(z_j) - B(y_j)]$ then:

$$\text{Sound}_{\mathcal{S}}[B / A] = 1 - \max[0, A(z_{j^*}) - B(y_{j^*})]$$

In order to further our intuition we now consider the case where the sets $A$ and $B$ are crisp, here $A(z)$ and $B(z) \in \{0, 1\}$ for all $z \in X$. Recalling that:

$$\text{Sound}_{\mathcal{S}}[B / A] = \min_{x \in A} [1 - A(x)] \lor \text{Con}_{\mathcal{S}}(A(x), B)]$$

we see that for $x \in A$, $1 - A(x) = 0$ and for $x \not\in A$, $1 - A(x) = 1$, hence, we have:

$$\text{Sound}_{\mathcal{S}}[B / A] = \min_{x \in A} [\text{Con}_{\mathcal{S}}(A(x), B)]$$

Furthermore, $$\text{Con}_{\mathcal{S}}(A(x), B) = \max_{y \in X} [\text{Con}_{\mathcal{S}}(A(x), B(y))]$$

where:

$$\text{Con}_{\mathcal{S}}(A(x), B(y)) = (1 - \max(0, A(x) - B(y))) \land S(x, y).$$

Our interest is only in $x \in A$, $A(x) = 1$. First, consider the case where $x \in B$. Here $B(x) = 1$ and since $S(x, x) = 1$ then $\text{Con}_{\mathcal{S}}(A(x), B(x)) = 1$. Hence, $\text{Con}_{\mathcal{S}}(A(x), B(x)) = 1$ for $x \in A \cap B$. Now consider the case where $x \not\in B$, so $x \in A$ and $x \not\in B$. In this case:

$$\text{Con}_{\mathcal{S}}(A(x), B(x)) = (1 - \max(0, A(x) - B(x))) \land S(x, x) = (1 - (\max(0, 1 - 0))) = 0$$

More generally if $z \in X$ is any element so that $z \not\in B$ then:

$$\text{Con}_{\mathcal{S}}(A(x), B(z)) = (1 - \max(0, 1 - 0)) \land S(x, z) = 0$$

Finally, if $y$ is any element contained in $B$, $B(y) = 1$ then:

$$\text{Con}_{\mathcal{S}}(A(x), B(y)) = (1 - \max(0, A(x) - B(y))) \land S(x, y)$$

however, with $A(x) = B(y) = 1$ then $\text{Con}_{\mathcal{S}}(A(x), B(y)) = S(x, y)$. Thus, we see that for $x \in A$: 

$$\text{Sound}_{\mathcal{S}}[B / A] = 1 - \max[0, A(z_{j^*}) - B(y_{j^*})]$$
Con_S(A(x), B(y)) = \max_{y \in X} \{Con_S(A(x) - B(y))\} = \max_{y \in B} \{S(x, y)\}

We now summarise our results in the following observations.

**Observation:** Assume A and B are crisp sets

\[\text{Sound}_S(B / A) = \min_{x \in A, y \in B} \{\max[S(x, y)]\}\]

Thus, here in this case of crisp sets for each \(x \in A\) we find the most similar element in \(B\) and then the soundness or reasonableness of inferring \(V\) is \(B\) in the least of these maximal similarities.

### 3 Possibility and certainty measures under similarity measure

Within Zadeh’s (1979b, 2005, 2006) fuzzy set-based theory of approximate reasoning the concepts of possibility and certainty (necessity) provide respectively generalisations of the degree of intersection and containment. Assume \(V\) is a variable taking its value in \(X\). Let \(A\) and \(B\) be fuzzy subsets of \(X\). The possibility of \(V\) is \(B\) given \(V\) is \(A\) is defined as:

\[\text{Poss}[V \text{ is } B / V \text{ is } A] = \max_{x \in X} [A(x) \wedge B(x)]\]

We observe in the case where \(A\) and \(B\) are crisp sets that \(\text{Poss}[V \text{ is } B / V \text{ is } A] = 1\) if \(A \cap B \neq \emptyset\) and is zero if \(A \cap B = \emptyset\). The concept of possibility generalises the binary idea of intersection by providing a measure of the degree of intersection. In the following, whenever it causes no confusion, we shall use the simpler notation \(\text{Poss}(B / A)\) instead of \(\text{Poss}[V \text{ is } B / V \text{ is } A]\).

The certainty of \(V\) is \(B\) given \(V\) is \(A\) is defined as:

\[\text{Cert}[V \text{ is } B / V \text{ is } A] = 1 - \text{Poss}[B / A]\]

We observe that:

\[\text{Cert}[B / A] = 1 - \max_{x} [\bar{B}(x) \wedge A(x)]\]

which can also be expressed as:

\[\text{Cert}[B / A] = \min_{x} [\bar{A}(x) \vee B(x)]\]

Let us look at this concept of certainty. We first consider the case where \(A\) and \(B\) are crisp. We see that if \(A \subseteq B\), then \(A \not\subset \bar{B}, A \cap \bar{B} = \emptyset\) and hence \(\text{Poss}[B / A] = 0\) and \(\text{Cert}[B / A] = 1\). On the other hand, if \(A \not\subset B\) then \(A \cap \bar{B} \neq \emptyset\) and the Poss[\(B / A\)] = 1 and Cert[\(B / A\)] = 0. Thus, we see this captures the idea of \(A\) being contained in \(B\).

We should point out a fundamental difference between possibility and certainty. While Poss[\(B / A\)] = Poss[\(A / B\)] we have Cert[\(B / A\)] \neq Cert[\(A / B\)].

**Note:** Dubois and Prade (1987) refer to certainty as necessity.
We now consider a generalisation of the concepts of possibility and certainty to the case
where we have a similarity relationship $S$ on the space $X$. In the case of possibility we can observe:

\[
\text{Poss}[B / A] = \max_{x,y} [A(x) \land B(y) \land S(x, y)]
\]

Thus, here the measure of possibility is softened by including some similarity.

We see in the case where $S = S*$ that:

\[
\text{Poss}_{S*}[B / A] = \text{Poss}[B / A] = \max_{x} [A(x) \land B(x)].
\]

In the case where $S = S^*$ then we see that:

\[
\text{Poss}_{S^*}[B / A] = \max_{x} [A(x)] \land \max_{x} B(x)
\]

We can express:

\[
\text{Poss}[V B / V A] = \max_{x} [A(x) \land (\max_{y} (B(y) \land S(x, y)))]
\]

Here we can view $B_S(x) = \max_{y} (B(y) \land S(x, y))$ as a kind of distorted view of $B$ in the
context of $S$. For each $x \in X$ $B_S(x)$ can be viewed as the similarity of the element $x$ with
the fuzzy set $B$. Using this, we see $\text{Poss}_{S}[B / A] = \max_{x} [A(x) \land B_S(x)]$.

One can naturally extend this consideration of a similarity relationship to the case of
the measure of certainty:

\[
\text{Cert}_{S}[B / A] = 1 - \text{Poss}_{S}[\bar{B} / A] = 1 - \max_{x,y} [\bar{B}(y) \land A(x) \land S(x, y)]
\]

\[
\text{Cert}_S[B / A] = \max_{x,y} [B(y) \lor \bar{A}(x) \lor \bar{S}(x, y)]
\]

So we see that we can naturally extend the ideas of possibility and certainty to
environments in which we have an underlining similarity relation.

In the preceding definition of possibility we have used the t-norm $\min (\land)$ to
implement the intersection operation in the definition of possibility. A more general
formulation for the definition of possibility is to use an arbitrary t-norm $T$ in place of $\land$
(Klement et al., 2000; Klir, 2006). In this case, we get:

\[
\text{Poss}[V is B / V is A] = \max_{x} [T(A(x), B(x))]
\]

Using this more general definition for possibility in the formulation for the concept of
certainty, we get:

\[
\text{Cert}(B / A) = 1 - \text{Poss}_T[\bar{B} / A] = 1 - \max_{x} [T(\bar{B}(x), A(x))]
\]

These more general definitions for possibility and certainty can easily be enhanced to
include the idea similarity over the underlying space. In this case:

\[
\text{Poss}_{S}[V is B / V is A] = \max_{x,y} [T(A(x), B(x)) \land S(x, y)]
\]

and

\[
\text{Cert}_{S}[V is B / V is A] = 1 - \text{Poss}_{S}[\bar{B} / A] = 1 - \max_{x,y} [T(\bar{A}(x), B(x)) \land S(x, y)]
\]
A very important example of t-norm is the Lukasiewicz t-norm, $T_L$. For this t-norm:

$$T_L(a, b) = \text{Max}[0, a + b - 1] = 0 \lor (a + b - 1)$$

Let us look at the formulation $\text{Cert}_S[B/A]$ in this case:

$$\text{Cert}_S[B/A] = 1 - \text{Max}_{x,y}[T_L(\overline{A(x)}, B(x)) \land S(x, y)]$$

$$\text{Cert}_S[B/A] = 1 - \text{Max}_{x,y}[(0 \lor (1 - A(x) + B(x) - 1)) \land S(x, y)]$$

Therefore, we see using $T_L$ we get:

$$\text{Cert}_S[B/A] = 1 - \text{Max}_{x,y}[0, ((A(x) \lor B(y)) \land S(x, y))]$$

This is our measure of soundness. Thus, our measure of soundness of alternative $V$ is $B$ to $V$ is $A$ is closely related to the measure certainty of $V$ is $B$ given $V$ is $A$.

### 4 Dempster-Shafer belief structures

In the preceding, we considered the issue of determining the reasonableness of concluding $V$ is $B$ given $V$ is $A$ in situations in which we have some underlying similarity relationship over the space $X$. In particular, we used the measure of soundness of altering $A$ to $B$, $\text{Sound}_{S}(B/A)$. We now look to extend our capability for making reasonable conclusions in the context described by the similarity relationship $S$ to situations in which in addition to the fact that $V$ is $A$ and we also have probabilistic information about the value of the variable $V$. In anticipation of addressing this issue, we review some ideas from the D-S theory of evidence (Dempster, 1967; Shafer, 1976; Yager and Liu, 2008) which will provide a framework for expressing probability information compatible with our objective.

Assume $V$ is a variable taking its value in the space $X$. A standard D-S structure associated with this variable consists of a collection of focal elements, non-null subsets of $X$, $D_i$, $i = 1$ to $q$ and an associated non-negative function $m(D_i) = d_i$ for which $\sum_{i=1}^{q} d_i = 1$.

Various semantics have been associated with the D-S structure, for our purposes we associate with this D-S structure the semantics that it is providing that information that $V$ is $D_i$ with probability $d_i$.

A number of special examples of D-S belief structure are worth noting. The case where $D_i = \{x_i\}$ and $m(D_i) = d_i$ can be seen as a classical probability distribution in which $d_i$ is the probability of $x_i$. This sometimes is referred to as a Bayesian distribution.

Another special case is the situation where $m(D_i) = 1$. Here we are saying that we are sure that the value of $V$ lies in $D_i$. This can be seen as the same as the unqualified statement $V$ is $D_i$. A particular case of this is where we have $D_i = X$, here we have no information about the value of $V$. A related but different situation is the case where $D_i = \{x_i\}$ and $m(D_i) = 1/n$. Here we are saying that all the outcomes have the same probability. We should note that this is different from the preceding case $m(X) = 1$ in which we are saying we know nothing, not even the fact that they all the $x_i$ have the same probability.
Another notable special case is where $D_1 = A$ and $D_2 = X$ and $m(D_1) = \alpha$ and $m(D_2) = 1 - \alpha$. Here we are saying with at least $\alpha$ probability the value of $V$ lies in $A$.

Two set measures have been associated within the D-S framework; these are the measures of plausibility and belief. If $B$ is an arbitrary subset of $X$ and $m$ is a belief structure on $V$ the plausibility of $V$ is $B$ denoted $Pl(B)$ is defined as:

$$Pl(B) = \sum_{D_1 \cap B \neq \emptyset} m(D_1)$$

The second measure of the belief that $V$ is $B$ denoted $Bel(B)$ is defined as:

$$Bel(B) = \sum_{D_1 \subseteq B} m(D_1)$$

It can be easily shown that $Pl(B) \geq Bel(B)$.

While in general, $Pl(B) \neq Bel(B)$ in the special case where $m$ is a Bayesian belief structure then we get $Pl(B) = Bel(B) = \sum_{x \in B} m(\{x\})$. In this case, this is essentially the probability of $B$ given the probability distribution $p_i = m(\{x_i\})$.

In the case where $D_1 = X$ and $m(D_1) = 1$ we get for all $B \neq \emptyset$, $Pl(B) = 1$ and $Bel(B) = 0$.

Consider the case where we have $m(D) = 1$, we are certain that $V$ lies in $D$. Consider now the plausibility and belief of some subsets $B$.

1. $B \cap D = \emptyset$ \implies $Pl(B) = Bel(B) = 0$
2. $B \cap D \neq \emptyset$ and $D \subseteq B$ \implies $Pl(B) = Bel(B) = 1$
3. $B \cap D \neq \emptyset$ and $D \not\subseteq B$ \implies $Pl(B) = 1$ and $Bel(B) = 0$

Consider the definition:

$$Pl(B) = \sum_{D_1 \cap B \neq \emptyset} m(D_1),$$

it is the sum of the weights of the focal elements that intersect $B$. Consider now:

$$Poss[B / D_i] = \max_x [B \cap D_i].$$

We see that this equals one where $B \cap D_i \neq \emptyset$ and this equals zero when $B \cap D_i = \emptyset$. Using this we can rewrite the definition of plausibility as:

$$Pl(B) = \sum_{i=1}^{q} Poss(B / D_i)m(D_i).$$

In this case, $Pl(B)$ looks like the expected possibility associated with the focal elements. Consider now the definition:

$$Bel(B) = \sum_{D_i \subseteq B} m(D_i),$$
it is the sum of the weights of the focal elements contained in D. Consider now:

\[
\text{Cert}(B / D_i) = 1 - \text{Poss}(\overline{B} / D_i)
\]

We see that this equals one when \(D_i \subseteq B\) and equals zero when \(D_i \not\subseteq B\). Using this, we can rewrite the definition of belief as:

\[
\text{Bel}(B) = \sum_{i=1}^{q} \text{Cert}(B / D_i) m(D_i).
\]

Dempster’s (1967) rule provides a mechanism for combining multiple independent D-S belief structures. Assume \(m_1\) has focal elements \(E_i\) for \(i = 1\) to \(q_1\) and \(m_2\) has focal elements \(F_j\) for \(j = 1\) to \(q_2\). Denoting \(m = m_1 \oplus m_2\) as the belief structure obtained by combining these two belief structures. We obtain \(m\) as follows. The focal elements of \(m\) are \(D_k = E_i \cap F_j \neq \emptyset\), the non-null intersection of the focal elements of \(m_1\) and \(m_2\) and

\[
m(D_k) = \frac{m(E_i) \cdot m(F_j)}{1 - \text{Con}}
\]

where \(\text{Con} = \sum_{E_i \cap F_j = \emptyset} m(E_i) \cdot m(F_j)\). We note that

\[
1 - \text{Con} = 1 - \sum_{E_i \cap F_j = \emptyset} m(E_i) \cdot m(F_j) = \sum_{E_i \cap F_j \neq \emptyset} m(E_i) \cdot m(F_j). \text{ Hence:}
\]

\[
m(D_k) = \sum_{E_i \cap F_j \neq \emptyset} \frac{m(E_i) \cdot m(F_j)}{m(E_i) \cdot m(F_j)}.
\]

**Note:** Dempster’s rule requires that \(\text{Con} \neq 0\). This is assured if there exists at least one pair \(E_i\) and \(F_j\) such that \(E_i \cap F_j \neq \emptyset\). We shall refer to two belief structures satisfying this as non-conflicting.

We now consider the combination of some notable belief structures. Assume \(m_1\) is such that \(m_1(X) = 1\). It has one focal element with weight equal to 1. This can be seen as corresponding to a belief structure with no information. Let \(m_2\) be any arbitrary belief structure with focal elements \(D_i\), \(i = 1\) to \(q\) and \(m(D_i) = \alpha_i\). We can show that \(m = m_1 \oplus m_2 = m_2\). Thus the combination of \(m_1\) with any belief structure \(m_2\) always yields \(m_2\). This is reasonable since \(m_1\) corresponds to no information.

Another important case is the following. Let \(m_1\) be such that \(m_1(D) = 1\). This corresponds to the statement \(V\) is \(D\). Let \(m_2\) be a Bayesian belief structure, it has focal element \(E_i = \{x_i\}\) with \(m_2(E_i) = p_i\). Consider now \(m = m_1 \oplus m_2\). In this case:

\[
\text{Con} = \sum_{E_i \cap D \neq \emptyset} p_i = 1 - \sum_{E_i \cap D = \emptyset} p_i = 1 - \sum_{x_i \in D} p_i
\]

The focal elements of \(m\) are \(F_k = \{x_k\}\) such that \(x_k \in D\) and \(m(F_k) = \frac{p_k}{1 - \text{Con}} = \frac{p_k}{\sum_{x_i \in D} p_i}\).
Thus in this case, \( m \) is a Bayesian D-S structure with:

\[
m(x_k) = \sum_{x_k \in D} \frac{p_k}{p_k} \quad \text{for } x_k \in D
\]

\[
m(x_k) = 0 \quad \text{for } x_k \not\in D
\]

In this case, for any subset \( B \):

\[
Bel(B) = Pl(B) = \frac{\sum_{x_k \in D \cap B} p_k}{\sum_{x_k \in D} p_k}
\]

Actually, we see this is the same as the conditional probability of \( B \) given \( V \) is in \( D \):

\[
Prob(B/D) = \frac{Prob(B \cap D)}{Prob(D)}
\]

An interesting related case is where \( m_1 \) is such that \( m_1(D) = \alpha \) and \( m_1(X) = 1 - \alpha \) and \( m_2 \) is a Bayesian structure with \( m_2({x}_i) = p_i \). In this case, the combination \( m \) is such that the focal elements are \( F_k = \{x_k\} \) for all \( x_k \) and:

\[
m(F_k) = \frac{p_k}{1 - \alpha \sum_{x_i \in D} p_i} \quad \text{for } x_k \in D
\]

\[
m(F_k) = \frac{(1 - \alpha)p_k}{1 - \alpha \sum_{x_i \in D} p_i} \quad \text{for } x_k \not\in D
\]

An important general observation can be made about the combination of belief structures. Assume \( m_1 \) is any arbitrary belief structure and \( m_2 \) is a Bayesian belief structure. If \( m = m_1 \oplus m_2 \) then \( m \) is always a Bayesian belief structure. Let \( m_1 \) have \( q \) focal elements \( D_i \) with \( m_1(D_i) = \alpha_i \). Let \( m_2 \) have focal elements \( E_k = \{x_k\} \), with \( m(F_k) \). The focal elements of \( m \) are going to be all sets corresponding \( D_i \cap E_k \neq \emptyset \). However, since \( E_k = \{x_k\} \) then \( D_i \cap E_k \) is going to be either \( \{x_k\} \) or \( \emptyset \). From this, it follows that the focal elements of \( m \) are singletons and hence \( m \) is Bayesian. Thus, the combination of a Bayesian D-S structure with any other non-conflicting D-S structure is always Bayesian.

We now consider the concept of \( Pl(V \text{ is } m_2/V \text{ is } m_1) \). For simplicity, assume \( m_1 \) has focal elements \( E_i \) with \( m_1(E_i) = \alpha_i \) and \( m_2 \) has focal elements \( F_j \) where \( m_2(F_j) = \beta_j \). Here we can obtain:

\[
Pl(V \text{ is } F_j/V \text{ is } m_1) = \sum_i \text{Poss}(F_j/E_i) \alpha_i
\]

We now define:

\[
Pl(V \text{ is } m_2/V \text{ is } m_1) = \sum_j Pl(V \text{ is } F_j/V \text{ is } m_1) \beta_j
\]

\[
Pl(V \text{ is } m_2/V \text{ is } m_1) = \sum_j \left( \sum_i \text{Poss}(F_j/E_i) \alpha_i \beta_j \right)
\]
We further recall that $\text{Poss}(F / E_i) = 1$ if $E_i \cap F_j \neq \emptyset$ and $\text{Poss}(F / E_i) = 0$ if $E_i \cap F_j = \emptyset$.

From this see that:

$$\text{Pl}(V \text{ is } m_2 / V \text{ is } m_1) = \sum_{E_i \cap F_j \neq \emptyset} \alpha_i \beta_j$$

$$\text{Pl}(V \text{ is } m_2 / V \text{ is } m_1) = 1 - \text{Con}(m_2 / m_1)$$

It is the negation of the conflict between $m_1$ and $m_2$. It also should be clear that:

$$\text{Pl}(V \text{ is } m_2 / V \text{ is } m_1) = \text{Pl}(V \text{ is } m_1 / V \text{ is } m_2)$$

We now can also provide a definition for $\text{Bel}(V \text{ is } m_2 \cap V \text{ is } m_1)$. Let us first obtain:

$$\text{Bel}(V \text{ is } F_j \cap V \text{ is } m_1) = \sum_{E_i \subseteq F_j} \alpha_i = \sum_{E_i \subseteq F_j} \text{Cert}(F_j / E_i) \alpha_i$$

Using this we define:

$$\text{Bel}(V \text{ is } m_2 / V \text{ is } m_1) = \sum_j \text{Bel}(V \text{ is } F_j \cap V \text{ is } m_1) \beta_j$$

$$\text{Bel}(V \text{ is } m_2 / V \text{ is } m_1) = \sum_j \left( \sum_i \text{Cert}(F_j / E_i) \alpha_i \beta_j \right)$$

$$\text{Bel}(V \text{ is } m_2 / V \text{ is } m_1) = \sum_{E_i \subseteq F_j} \alpha_i \beta_j$$

Let us look at some examples. First consider two Bayesian belief structures with $m_1(E_i) = \alpha_i$ and $m_2(F_j) = \beta_j$, where $E_i = \{x_i\}$ and $F_j = \{x_j\}$. In this case we get:

$$\text{Pl}(V \text{ is } m_2 / V \text{ is } m_1) = \text{Bel}(V \text{ is } m_2 / V \text{ is } m_1) = \sum_i \alpha_i \beta_j$$

Consider now the case where $m_1$ is Bayesian, $E_i = \{x_i\}$ with $m_1(E_i) = \alpha_i$ and $m_2$ has focal elements $F_j$ with $m_2(F_j) = \beta_j$. We see that:

$$\text{Pl}(m_2 / m_1) = \sum_{E_i \cap F_j \neq \emptyset} \alpha_i \beta_j = \sum_i \alpha_i \left( \sum_{j \text{ such that } x_i \subseteq F_j} \beta_j \right)$$

Now let us determine the belief:

$$\text{Bel}(m_2 / m_1) = \sum_{E_i \cap F_j \neq \emptyset} \alpha_i \beta_j = \sum_i \alpha_i \left( \sum_{j \text{ such that } x_i \subseteq F_j} \beta_j \right)$$

It is the same.

An extension of the D-S structure to the case where the focal elements are fuzzy subsets was introduced in Yen (1990). Here we let the focal elements $D_i, i = 1 \text{ to } q$, be a collection of normal fuzzy subsets of $X$. We recall a fuzzy subset $D_i$ is normal if $\text{Max}_x[D_i(x)] = 1$. Associated with each of these focal elements, as is in the crisp case, is a value $m(D_i) \in [0, 1]$ such that $\sum_{i=1}^{q} m(D_i) = 1$. 


The concepts of plausibility and belief can be extended to this fuzzy framework. Let $B$ be any normal fuzzy subset of $X$. Then:

$$\text{Pl}(B) = \text{Pl}(V \text{ is } B / V \text{ is } m) = \sum_i \text{Poss}(B / D_i)m(D_i)$$

where we recall $\text{Poss}[B/D_i] = \text{Max}[B \cap D_i]$. Thus $\text{Pl}(B)$ is the expected possibility. We see that if the sets $B$ and $D_i$ are crisp then $\text{Poss}[B/D_i] = 1$ if $B \cap D_i \neq \emptyset$ otherwise it is zero. From this, it follows that in the crisp case this definition collapses to the usual definition, $\text{Pl}(B) = \sum_{B \subseteq D_i \neq \emptyset} m(D_i)$.

In an analogous manner we define:

$$\text{Bel}(B) = \sum_i \text{Poss}(B / D_i)m(D_i) = \sum_i (1 - \text{Poss}(\overline{B} / D_i))m(D_i)$$

This again can be shown to collapse to the usual definition where all sets are crisp.

The issue of extending Dempster’s rule to the case of fuzzy sets is slightly more complicated because of the possibility of subnormal fuzzy subsets, those with maximal membership grade of less than one. Here we follow the approach suggested by Yen (1990).

Assume $m_1$ and $m_2$ are two belief structures with focal elements $E_i$, $i = 1$ to $n_1$ and $F_j$, $j = 1$ to $n_2$ respectively. Furthermore let us denote $m_1(E_i) = \alpha_i$ and $m_2(F_j) = \beta_j$. For each pair $E_i$ and $F_j$, let $G_{ij} = E_i \cap F_j$ and $k_{ij} = \text{Max}_x(G_{ij}(x))$. With each $G_{ij}$ we associate a fuzzy subsets $H_{ij}$ of $X$ such that $H_{ij}(x) = G_{ij}(x)/k_{ij}$. Here we note that each $H_{ij}$ is a normal fuzzy set, if $G_{ij}$ is normal, $k_{ij} = 1$, then $H_{ij} = G_{ij}$. We have simply normalised $H_{ij}$. Finally, we let $K = \sum_{i,j} k_{ij}\alpha_i\beta_j$. Using the preceding, we now define $m = m_1 \oplus m_2$, the combination of the belief structures $m_1 \oplus m_2$, as a belief structure on $X$ whose focal elements are the $H_{ij}$ and

$$m(H_{ij}) = \frac{k_{ij}\alpha_i\beta_j}{K}.$$ It can be easily shown that if the sets are crisp this becomes the usual Dempster’s rule.

We now look at a special case. Here we assume $m_1$ has just one focal element, fuzzy subset $A$ where of course $m_1(A) = 1$. Let $m_2$ be a Bayesian belief structure, its focal elements are $E_j = \{x_j\}$ with $m(E_j) = p_j$. In this case, $G_{ij} = A \cap E_j = \left\{ \frac{A(x_j)}{x_j} \right\}$ and $k_j = A(x_j)$. Here defining $H_j$ such that $H_j(x) = G_{ij}(x) / k_j$ we get $H_j = \{x_j\} = E_j$. In addition with $K = \sum_j A(x_j)p_j$ we get that:

$$m(H_j) = \frac{A(x_j)p_j}{\sum_i A(x_i)p_i}$$
We see that m is itself a Bayesian structure.

Let B be any fuzzy subset of X then $P_l(B) = \sum_j \text{Poss}[B/H_j]m(H_j)$. Since with $H_j = \{x_j\}$ we have $\text{Poss}[B/H_j] = B(x_j)$ and hence:

$$P_l(B) = \frac{\sum_j B(x_j)A(x_j)p_j}{\sum_i A(x_i)p_i}$$

Since m is Bayesian then $P_l(B) = \text{Cert}(B) = \text{Prob}(B)$.

The above result can be seen to be equivalent to Zadeh’s (1968) definition of the conditional probability of the fuzzy set B given the fuzzy set A, $P(B / A) = \frac{\text{Prob}(A \cap B)}{\text{Prob}(A)}$.

Here we define $A \cap B$ using the product t-norm and use Zadeh’s definition of the probability of a fuzzy set F as $\text{Prob}(F) = \sum_j F(x_j)p_j$ (Zadeh, 1968).

### 5 On the reasonableness of conclusions

Earlier we suggested using the measure $\text{Sound}(B/A)$ to indicate the degree of reasonableness of concluding $V$ is B given the knowledge $V$ is A in the context of a similarity relation $S$. Here we shall extend our capability by obtaining a degree of reasonableness of concluding $V$ is B given the knowledge $V$ is m, in the context of a similarity relation $S$. We shall denote this as $\text{Reas}_S(V$ is B / $V$ is m). Here our knowledge is expressed as a D-S belief structure.

Let $V$ is m be a D-S belief structure having q focal elements, $F_j$, $j = 1$ to q with $m(F_j) = \alpha_j$. Using this we define:

$$\text{Reas}_S(V$ B / $V$ m) $\sum_j \text{Sound}_S(B / F_j)m(F_j).$$

Thus we see it as the expected soundness.

Before looking at some special cases we recall:

$$\text{Sound}_S(B / F_j) = \min_{x \in X} \text{Con}_S(F_j(x), B)$$

where $\text{Con}_S(F_j(x), B) = \max_{y \in X}[(1 - (0 \lor (F_j(x) - B(y)))) \land S(x, y)]$.

Consider now the case where $m$ is such that $F_1 = A$ and $m(F_1) = 1$, here we essentially have $V$ is A. In this case we get $\text{Reas}_S(B / A) = \text{Sound}_S(B / A)$ which is our original definition.

Consider now the case where $S$ is $S^*$, all elements in X are equivalent. In this case, $\text{Reas}_S(B / m) = \sum_j \text{Sound}_S(B / F_j)m(F_j)$. Since we have already shown $\text{Sound}_S(B / A) = 1$ for any A then we get $\text{Reas}_S(B / m) = \sum_j m(F_j) = 1$. Thus, here no matter what the form of $m$, every conclusion is reasonable.
Consider now the case where $S = S^\ast$. Here:

$$\text{Con}_{S^\ast}[F_j(x), B] = 1 - \text{Max}(0, F_j(x) - B(x)) = \text{Min}[1, \overline{F_j(x)} + B(x)]$$

$$\text{Sound}_{S^\ast}(B / F_j) = \text{Min}[^{\overline{F_j(x)}} \cup (1 \land (\overline{F_j(x)} + B(x)))]$$

$$\text{Sound}_{S^\ast}(B / F_j) = \text{Min}[\overline{F_j(x)} \lor 1 \land (\overline{F_j(x)} + B(x)) \lor \overline{F_j(x)}]$$

$$\text{Sound}_{S^\ast}(B / F_j) = \text{Min}[1 \land (\overline{F_j(x)} + B(x))] = 1 \land \text{Min}[(\overline{F_j(x)} + B(x))]$$

From this we get:

$$\text{Reas}_{S^\ast}(B / m) = \sum_j (1 \land \text{Min}[(\overline{F_j(x)} + B(x))])m(F_j)$$

An interesting special case is the situation in which our conclusion $B$, is some unique value $z \in X$. So here our conclusion is $V$ is $\{z\}$.

Let us look at the reasonableness of this in the light of the information $V$ is $m$ and the context of a similarity relationship $S$. First we calculate $\text{Sound}_S(\{z\} / A)$ for some arbitrary set $A$. We see that $\text{Con}_S[A(x), B] = \text{Max}(1 - \text{Max}(0, A(x) - B(y))) \land S(x, y)$ for $B = \{z\}$ becomes:

$$\text{Con}_S[A(x), B] = S(x, z) \lor \text{Max}[(1 - \text{Max}(0, A(x) - B(y))) \land S(x, y)]$$

Since $B(y) = 0$ for $y \neq z$ then:

$$\text{Con}_S[A(x), B] = S(x, z) \lor \text{Max}[(1 - A(x)) \land S(x, y)]$$

but $S(x, y) = 1$ for the case $y = x$ hence:

$$\text{Con}_S[A(x), \{z\}] = S(x, z) \lor 1 - A(x)$$

and therefore:

$$\text{Sound}_S(\{z\} / A) = \text{Min}_{x \in X}[\overline{A(x)} \lor S(x, z)].$$

In the special case where $A$ is a crisp set then $\overline{A(x)} = 1$ if $x \not\in A$ and $\overline{A(x)} = 0$ if $x \in A$ and hence $\text{Sound}_S(\{z\} / A) = \text{Min}_{x \in A}S(x, z)$. Thus in this crisp case it the minimal similarity of $z$ to any element in $A$.

We can directly extend this to the case where our knowledge is $V$ is $m$. Letting $F_j$ be the focal elements of $m$. Now we are interested in determining $\text{Reas}_S(\{z\} / m)$ which we calculate as:

$$\text{Reas}_S(\{z\} / m) = \sum_{j=1}^q \text{Sound}_S(\{z\} / F_j)m(F_j) = \sum_{j=1}^q \text{Min}_{x \in X}[\overline{F_j(x)} \lor S(x, z)]m(F_j)$$
In the special case where the \( F_j \) are crisp sets:

\[
\text{Reas}_S(B \mid m) = \sum_{j=1}^{q} \min_{x \in F_j} \min[S(x, z)]m(F_j)
\]

We now consider the case where \( m \) is a Bayesian belief structure. In this case we have as our focal elements \( F_j = \{x_j\} \) and \( m(F_j) = p_j \). We let our conjectured proposition be \( \forall is B \) and we assume our context is described by a similarity relationship \( S \). In this case:

\[
\text{Reas}_S(B \mid m) = \sum_j \text{Sound}_S(B \mid F_j)p_j
\]

Let us first calculate \( \text{Sound}_S(B \mid F_j) = \min_{x \in X}[(1 - F_j(x)) \lor \text{Con}(S, F_j(x), B)] \) when \( F_j = \{x_j\} \).

Since \( F_j(x) = 0 \) for \( x \neq x_j \) and \( F_j(x_j) = 1 \) then we simply get:

\[
\text{Sound}_S(B \mid F_j) = \text{Con}(S, F_j(x_j), B) = \max_{y \in X}[(1 - \max_{x \in X}(0, F_j(x_j) - B(y))) \land S(x_j, y)]
\]

Since \( F_j(x_j) = 1 \) then we get \( \text{Sound}_S(B \mid F_j) = \max_{y \in X}(B(y) \land S(x_j, y)) \). From this we get:

\[
\text{Reason}_S(B \mid m) = \sum_j \max_{y \in X}(B(y) \land S(x_j, y))p_j
\]

Some special cases of this are worth noting. If \( B \) is a crisp subset then:

\[
\max_{y \in S}(y) = \max_{y \in B}[S(x_j, y)]
\]

\[
\text{Reason}_S(B \mid m) = \sum_j \max_{y \in S}(y)p_j
\]

Furthermore, if \( B \) is a crisp subset just consisting of one element \( z \), then \( \max_{y \in S}(S(x_j, y)) = S(x_j, z) \). In this case we get \( \text{Reason}_S(\{z\} \mid p_j) = \sum_j S(x_j, z)p_j \). In this case, the reasonableness of conjecturing \( z \) is the expected similarity of \( z \) to the elements \( x_j \).

In Figure 1 we relate the two noted situations.

**Figure 1** Two situations of interest
In I, we are going from granular knowledge about V to a precise value for V, V is z. In II, we are going from precise probabilities to granules. In the case of crisp sets for Fj in I and B in II we get:

\[ I : \text{Reason}_S(z | m) = \sum_j (\text{Min}_{\text{x} \in F_j} [S(x, z)]m(F_j)) \]

\[ II : \text{Reason}_S(B | p_j) = \sum_j \text{Max}_{y \in B} [S(x_j, y)]p_j \]

Thus, in the first case, get the expected value of the z’s least similarity to an element in Fj while in the second case, we the expected value of the maximal similarity of each xj with an element in B.

We now consider another important case. Assume we have the knowledge that V is A and also have the knowledge that there exists a probability distribution P on the X with respect to the value of V such that P{V = x_i} = p_i. An example this occurs if we have some imprecise observation of V, V is A as well as some historical information about value of V expressed in terms of P. We can represent both of these pieces of information as D-S belief structures. The first piece of information is a simple belief structure V is m_1 where we have one focal element, A and m_1(A) = 1. The second piece of information can be represented as a Bayesian belief structure m_2 in which we have n focal elements E_j = {x_j} and m_2(E_j) = p_j.

The combination of these two pieces of knowledge is m = m_1 \oplus m_2. In this case as we have already shown m is a Bayesian belief structure with focal elements E_j = {x_j} with \[ \hat{m}(E_j) = \frac{A(x_j)p_j}{\sum_j A(x_j) \cdot p_j} \].

Having this information, consider the reasonableness of concluding V is B in the face of a context expressed by a similarity relation S. In this case, we are interested in \[ \text{Reas}_S(V is B | V is \hat{m}) \]. Here \[ \hat{m} \] is a Bayesian belief structure where the probability of x_j is \[ p(x_j) = \frac{A(x_j)p_j}{\sum_j A(x_j) \cdot p_j} \] instead of simply p_j. Using our previous results we get:

\[ \text{Reas}_S(V is B | \hat{m}) = \frac{\sum_j \hat{B}(x_j)A(z_j)p_j}{\sum_j A(x_j)p_j} \]

In the special case where B = {z}, it is a single value, then \[ \hat{B}(x_j) = S(x_j, z) \] and we get:

\[ \text{Reas}_S(V is z | \hat{m}) = \frac{\sum_j S(x_j, z)A(x_j)p_j}{\sum_j A(x_j)p_j} \]
As noted above, this captures an important class of problems. Here we have some imprecise observation of the value of the variable as well as some historical information about the variable and we are interested in conjecturing some value for the variable in the face of some context described by a similarity relation $S$.

Some important special cases are worth noting. In the case where our observation has no information, $A = X$, then we get as is anticipated

$$\text{Reas}_S[V \text{ is } B \mid V \text{ is } \tilde{m}] = \left[ \sum_j \hat{B}(x_j)p_j \right].$$

Consider now the case where $A$ is a precise value $x^*$. Here $A(x) = 1$ for $x = x^*$ and $A(x) = 0$ for $x \neq x^*$. From this we get

$$\text{Reas}_S[V \text{ is } B \mid \tilde{m}] = \hat{B}(x^*) = \text{Max}_y[B(y) \wedge S(y, x^*)].$$

If $B$ is a crisp set then $\text{Reas}_S[V \text{ is } B \mid \tilde{m}]$ is simply the maximum similarity of $x^*$ with an element in $B$.

Another special case is where $p_i = 1/n$. In this case, where we assume that all the $x_j$ have the same probability, we get:

$$\text{Reas}_S(V \text{ is } B \mid \tilde{m}) = \frac{\sum_j \hat{B}(x_j)A(x_j)}{\sum_j A(x_j)} = \sum_j \hat{B}(x_j)a_j$$

where $a_j = \frac{a(x_j)}{\sum_j A(x_j)}$.

Let us now compare this with another closely related special case, one in which we have no information about the probability distribution. In this case $m = m_1$, $V \text{ is } A$,

$$\text{Reas}_S[V \text{ is } B \mid \tilde{m}] = \text{Min}_{x \in X}[\tilde{A}(x) \vee \text{Con}_S(A(x), B)]$$

where $\text{Con}_S(A(x), B) = \text{Max}_y[(1 - \text{Max}(0, A(x) - B(y))) \wedge S(x, y)]$.

Let us now consider the special case where $B(y) \geq A(y)$, here we feel that it should be completely reasonable to conclude $B$. In this case:

$$\text{Reas}_S(V \text{ is } B \mid V \text{ is } \tilde{m}) = \frac{\sum_j \hat{B}(x_j)A(x_j)p_j}{\sum_j A(x_j)p_j}$$

With $\hat{B}(x_j) = \text{Max}_y[B(y) \wedge S(x_j, y)]$ and $B(x) \geq A(y)$ we get $\hat{B}(x_j) \geq B(x_j) \geq A(x_j)$. Here then:

$$\text{Reas}_S[V \text{ is } B \mid V \text{ is } \tilde{m}] \geq \frac{\sum_j \hat{A}(x_j)^2p_j}{\sum_j A(x_j)p_j}$$

In the binary case where $A(x_j) \in \{0, 1\}$ this becomes one.

Consider now the case where $B = \tilde{A}$, $B(x) = 1 - A(x)$. In general, the reasonableness of concluding $B$ will depend on the underlying similarity relation. For example if $S = S^*$
Drawing reasonable conclusions from information

then \( \hat{B}(x_j) = 1 \) and hence \( \text{Reas}_s[V \ is \ B \mid V \ is \ \hat{m}] = \frac{\sum_{j=1}^{n} A(x_j)p_j}{\sum_{j=1}^{n} A(x_j)p_j} = 1 \). On the other hand,

if \( S = S^* \) then \( \hat{B}(x_j) = B(x_j) \). In this case, \( \text{Reas}_s[V \ is \ B \mid V \ is \ \hat{m}] = \frac{\sum_{j=1}^{n} \overline{A}(x_j)A(x_j)p_j}{\sum_{j=1}^{n} A(x_j)p_j} \). If \( A \) is crisp then this becomes zero.

Let us now consider a special case of similarity. Here we assume \( A \) has no similarity with elements in not \( A \). For simplicity we shall consider the crisp environment. Here we mean to indicate \( S(x, y) \wedge A(x) \wedge \overline{A}(y) = 0 \).

\[
\text{Reas}_s(V \ is \ B \mid V \ is \ \hat{m}) = \frac{\sum_j \hat{B}(x_j)A(x_j)p_j}{\sum_j A(x_j)p_j}
\]

with \( \hat{B}(x) = \max_y [B(y) \wedge S(x, y)] \). With \( B = \overline{A} \) then \( \max_y [\overline{A}(y) \wedge S(x, y)] \). Since

\[
\hat{B}(x_j)A(x_j) \leq \hat{B}(x_j) \wedge A(x_j) \quad \text{then} \quad \text{Reas}_s(V \ is \ B \mid V \ is \ \hat{m}) \leq \frac{\sum_j (\hat{B}(x_j) \wedge A(x_j))p_j}{\sum_j A(x_j)p_j}.
\]

Consider the term:

\[
\hat{B}(x_j) \wedge A(x_j) = \max_y [B(y) \wedge S(x_j, y) \wedge A(x_j)] = \max_y [\overline{A}(y) \wedge S(x_j, y) \wedge A(x_j)]
\]

\[
\hat{B}(x_j) \wedge A(x_j) = (\overline{A}(x_j) \wedge S(x_j, y) \wedge A(x_j)) \vee \max_y [\overline{A}(y) \wedge S(x_j, y) \wedge A(x_j)]
\]

\[
\hat{B}(x_j) \wedge A(x_j) = \max_{y \neq x_j} [\overline{A}(y) \wedge S(x_j, y) \wedge A(x_j)] = 0.
\]

Thus in this case, \( \text{Reas}_s[V \ is \ B \mid V \ is \ \hat{m}] = 0 \).

In the fuzzy case, we replace \( S(x, y) \wedge A(x) \wedge \overline{A}(y) = 0 \) by \( S(x, y) \wedge A(x) \wedge \overline{A}(y) \leq \beta \).

In this case:

\[
\hat{B}(x_j) \wedge A(x_j) = (\overline{A}(x_j) \wedge S(x_j, y) \wedge A(x_j)) \vee \max_y [\overline{A}(y) \wedge S(x_j, y) \wedge A(x_j)]
\]

\[
\hat{B}(x_j) \wedge A(x_j) = A(x_j) \wedge \overline{A}(x_j) \vee \beta
\]

In this case \( \text{Reas}_s[B \mid \hat{m}] \leq \frac{\sum_j ((A(x_j) \wedge \overline{A}(x_j)) \vee \beta)p_j}{\sum_j A(x_j)p_j} \).
In the preceding, we showed for the case where we have the information \( V \text{ is } A \) and a probability distribution \( P \) over \( X \) that the reasonableness of a conjecture \( V \text{ is } B \) is:

\[
\text{Reas}_S[V \text{ is } B \mid A, P] = \frac{\sum_j \hat{B}(x_j)A(x_j)p_j}{\sum_j A(x_j)p_j}
\]

where \( \hat{B}(x_j) = \text{Max}_y [\hat{B}(y) \land S(x_j, y)] \). Here we now consider a variation of this problem.

We assume that rather than being completely confident about the observation \( V \text{ is } A \) we have a degree \( \alpha \) of confidence associated with this observation. In this case, the belief structure associated with this qualified statement becomes \( V \text{ is } m_3 \) where \( m_3 \) has focal elements \( A \) and \( X \) and \( m_3(A) = \alpha \) and \( m_3(X) = 1 - \alpha \). In this case, with the probability distribution represented as \( m_2 \), \( m_2(E_j) = p_j \), we get as our combined information \( V \text{ is } \hat{m}_2 \) where \( \hat{m}_2 = m_3 \oplus m_2 \). In this case, since \( m_2 \) is Bayesian, \( \hat{m}_2 \) is still Bayesian, it has focal elements \( E_j = \{x_j\} \) however, in this case, \( \hat{m}_2(E_j) = \frac{\alpha A(x_j)p_j + \overline{\alpha} p_j}{K} \) where

\[
K = \sum_j (\alpha A(x_j)p_j + \overline{\alpha} p_j) = \alpha \sum_j (A(x_j)p_j) + \overline{\alpha}
\]

thus \( \hat{m}_2(E_j) = \frac{\alpha A(x_j)p_j + \overline{\alpha} p_j}{\alpha \sum_j (A(x_j)p_j) + \overline{\alpha}} \). In this case, the reasonableness of the conclusion \( V \text{ is } B \) becomes:

\[
\text{Reas}_S[V \text{ is } B \mid V \text{ is } \hat{m}_2] = \sum_j \text{Sound}_S(B \mid E_j)\hat{m}_2(E_j)
\]

As we have previously shown with \( E_j = \{x_j\} \):

\[
\text{Sound}(B \mid E_j) = \text{Max}_{y \in X} [B(y) \land S(x_j, y)] = \hat{B}(x_j)
\]

and hence

\[
\text{Reas}_S[V \text{ is } B \mid V \text{ is } \hat{m}_2] = \frac{\sum_j B(x_j)(\alpha A(x_j)p_j + \overline{\alpha} p_j)}{\alpha \sum_j (A(x_j)p_j) + \overline{\alpha}}
\]

We can express this as:

\[
\text{Reas}_S[V \text{ is } B \mid V \text{ is } \hat{m}_2] = \frac{\sum_j B(x_j)p_j(1 - \alpha \overline{A}(x_j))}{\alpha \sum_j (A(x_j)p_j) + \overline{\alpha}}
\]

Since \( \hat{B}(x_j) = \text{Max}_y [B(y) \land S(x, y)] \) we see the larger the set \( B \) the larger its reasonableness. We observe that when \( \alpha = 1 \), we have complete confidence, then:

\[
\text{Reas}_S[V \text{ is } B \mid V \text{ is } \hat{m}_2] = \sum_{j=1}^n \hat{B}(x_j) p_j.
\]
We now consider another variation of the basic problem of having \( V \) is \( A \) and a probability distribution \( P \) on \( X \). Implicit in this basic problem is the understanding that we have a current observation of the value of the variable \( V \) manifested by the data \( V \) is \( A \). In addition, the probability distribution provides information about the typical value of \( V \). In this variation we shall consider the situation in which there is some concern about the ‘typicality’ of the current situation. This will manifest itself with some qualification about using the probabilistic information. Here we shall let \( \lambda \in [0, 1] \) indicate the degree of typicality. Here then we have two pieces of information. The first is \( V \) is \( A \) represented by \( m_1 \). The second is a variation on the probability distribution reflecting our questioning the typicality of the current subset. Here then we have \( m_4 \) which has \( n + 1 \) focal elements \( E_j = \{ x_j \} \) for \( j = 1 \) to \( n \) and \( X \). In this case \( m_4(X) = (1 - \lambda) \) and \( m_4(E_j) = \lambda p_j \) for \( j = 1 \) to \( n \). A notable observation here is that \( m_4 \) is not a Bayesian belief structure. In this case we let \( K = (1 - \lambda) + \lambda \sum_j p_j A(x_j) \) we get \( \tilde{m}(A) = \frac{(1 - \lambda)}{K} \) and \( \tilde{m}(E_j) = \frac{\lambda p_j A(x_j)}{K} \). In this case, the reasonableness of conjuncture \( V \) is \( B \) becomes:

\[
\text{Reas}_S [B \mid \tilde{m}] = \frac{1}{K} \left[ \text{Sound}_S [B \mid A](1 - \lambda) + \lambda \sum_{j=1}^n \text{Sound}_S [B \mid E_j] \lambda p_j A(x_j) \right]
\]

The association of non-typicality with a situation implies a weakening of the constraint implied by the probability distribution. A consideration of non-typicality should be based on an indication of something unusual about the current situation. One source of this indication can be something external to our structure. Some additional information informing us that this situation is different than the usual.

Another indication of possible non-typicality is when \( V \) is \( A \) is an unusual observation, it is not very compatible with what we would expect given the probability distribution \( P \). Here we shall suggest one method for obtaining a degree of typicality in this case based on the concept of surprise introduced by Prade and Yager (1994). Here we first normalise the probability distribution \( P \) to get a possibility distribution \( \Pi_P \) so that

\[
\Pi_P(x_j) = \frac{p_j}{\max_i [p_i]}.
\]

Here we note if \( p_j = 1/n \) for all \( x_j \) then \( \Pi_P(x_j) = 1 \) for \( x_j \) and if \( p_k = 1 \) then \( \Pi_P(x_k) = 1 \) and \( \Pi_P(x_j) = 0 \) for all \( j \neq k \). Here then using possibility the distribution \( \Pi_P \) we obtain:

\[
\lambda = \text{Poss}[A \mid \Pi_P] = \max_{x_i \in A} [A(x_i) \land \Pi_P(x_i)]
\]

We note in the case where \( A \) is a crisp set we get \( \lambda = \max_{x_j \in X} \left[ \frac{p_j}{\Pi_P} \right] \) where \( p^* \) is \( \max_{x_j \in X} [p_j] \).

In the above, we are basing our measure of typicality of the situation on whether one element in \( A \) is compatible with the usual case as expressed by \( \Pi_P \). This is equivalent to the condition:
not (all the elements in A are not compatible with $\Pi_p$).

A slightly stronger condition can be had if we require:

not (most of the element in A are not compatible with $\Pi_p$).

Using this condition we get:

$$\lambda = 1 - \text{Most} \left( \frac{\sum_i (A(x_i) \land (1 - \Pi_p(x_i)))}{\sum_i A(x_i)} \right)$$

where Most is a fuzzy subset of the unit interval representing the concept most.

### 6 $\alpha$-Reasonableness

Assume we have information about a variable V described by a D-S belief structure $m$ which has focal elements $F_i$, $i = 1$ to $q$, with weights $m(F_i)$. In the context of a similarity relation $S$, we suggested the reasonableness of a conjecture $V$ is $B$ is obtained as:

$$\text{Reas}_S[V \ is \ B \ | \ V \ is \ m] = \sum_{i=1}^{q} \text{Sound}_S(B/F_i) m(F_i) .$$

Here we introduce a more general formation for the determination of the reasonableness. We shall refer to this as $\alpha$-reasonableness. Here $\alpha \in [0, 1]$ is a parameter that controls the strictness of judging the reasonableness of $V$ is $B$ in the face of the knowledge $V$ is $m$. In particular, the larger $\alpha$ the more generous in determining reasonableness. As we shall see this formulation is based on the weighted OWA aggregation (Yager, 1988) of the soundness associated with the different focal elements.

As a first step we order the $\text{Sound}_S(B/F_i)$ and let $\text{ind}$ be a function so that $\text{ind}(j)$ is the index of the $j$th largest value for $\text{Sound}_S(B/F_i)$. We use the notation $\text{Sound}_S(B/F_{\text{ind}(j)})$ to indicate the $j$th largest value. In addition, we use $m(F_{\text{ind}(j)})$ to indicate the weight associated with the focal element with the $j$th largest soundness value. We now associate with the parameter $\alpha$ a function $Q_\alpha(z) = z^{\frac{1}{1-\alpha}}$ defined on the unit interval. We note $Q_\alpha(0) = 0$, $Q_\alpha(1) = 1$ and $Q_\alpha$ is monotonic, $Q_\alpha(z_1) \geq Q_\alpha(z_2)$ if $z_1 > z_2$. Using the function $Q_\alpha$, we determine a collection of $q$ weights:

$$w_j = Q_\alpha(T_j) - Q_\alpha(T_{j-1}) \quad j = 1 \ to \ q .$$

In the above, $T_0 = 0$ and $T_j = \sum_{k=1}^{j} m(F_{\text{ind}(k)})$ for $j = 1$ to $q$.

Using these weights we calculate:

$$\alpha - \text{Reason}_S[V \ is \ B \ | \ V \ is \ m] = \sum_{j=1}^{q} w_j \text{Sound}_S[B/F_{\text{ind}(j)}]$$
Use see in the special case when $\alpha = 0.5$ that $Q_\alpha(z) = z$. In this case $w_j = m(F_{ind_j})$ and hence:

$$\alpha - \text{Reason}_S[V \text{ is } B \mid V \text{ is } m] = \sum_{j=1}^{\alpha} \text{Sound}_S[B/F_j] \ln(F_j),$$

our original definition. Thus our original definition is a special case when $\alpha = 0.5$. It can be shown that if $\alpha_1 > \alpha_2$ then:

$$\alpha_1 - \text{Reason}_S[V \text{ is } B \mid V \text{ is } B] \geq \alpha_2 - \text{Reason}_S[V \text{ is } B \mid V \text{ is } B]$$

Thus increasing $\alpha$ is easing the requirement for accepting $V$ is $B$ as reasonable. Furthermore, it can be shown (Yager, 1993) that if $\alpha \to 0$ then:

$$\alpha - \text{Reason}_S[V \text{ is } B \mid V \text{ is } B] \to \text{Min}_j[\text{Sound}_S[B/F_j]].$$

On the other hand, if $\alpha \to 1$ then:

$$\alpha - \text{Reason}_S[V \text{ is } B \mid V \text{ is } B] \to \text{Max}_j[\text{Sound}_S[B/F_j]].$$

Thus here we have a parameterised family of measures of reasonableness. Here the value of $\alpha$ is related to the conservativeness of the user, the smaller $\alpha$ the more conservative. This variety of measures can be particularly useful in modelling systems of interacting multi-agent systems. Here different agents can be imbued with different $\alpha$ to reflect their different natures. In some sense, the value $\alpha$ can be seen as reflection of the optimism, the larger $\alpha$ the more optimistic.

7 Conclusions

We were interested in the process of drawing reasonable conclusions about the value of a variable. We indicated that reasonableness generally depends on the information we have about the variable as well as the context in which we shall use the assumed value. In order to include a wide range of imprecise and uncertain information, we used granular computing technologies such as fuzzy sets, D-S belief structures and probability theory to represent our knowledge and assumptions. In order to provide some structure, we restricted ourselves to the special case where context is modelled using a similarity relationship. Within this framework we suggested a measure of the reasonableness and looked at the properties of the measure and investigate its performance in a number of special cases. A particularly important special case we investigated was one in which our information about a variable consisted of an imprecise observation of its value represented using fuzzy sets and historical data about the variable expressed using a probability distribution.
References


A new fuzzy \( k \)-modes clustering algorithm for categorical data

Michael K. Ng* and Liping Jing

Department of Mathematics,
Centre for Mathematical Imaging and Vision,
Hong Kong Baptist University,
Kowloon Tong, Hong Kong
E-mail: mng@math.hkbu.edu.hk
E-mail: lpjinghk@gmail.com
*Corresponding author

Abstract: This correspondence describes extensions to the fuzzy \( k \)-modes algorithm for clustering categorical data. We modify a simple matching dissimilarity measure for categorical objects, which allows the use of the fuzzy \( k \)-modes paradigm to obtain a cluster with strong intra-similarity, and to efficiently cluster large categorical data sets. We derive rigorously the updating formula of the fuzzy \( k \)-modes clustering algorithm with the new dissimilarity measure, and the convergence of the algorithm under the optimisation framework. Experimental results are presented to illustrate that the effectiveness of the new fuzzy \( k \)-modes algorithm is better than those of the other existing \( k \)-modes algorithms.

Keywords: categorical data; clustering; data mining; fuzzy \( k \)-modes algorithm.


Biographical notes: Michael K. Ng is a Professor in the Department of Mathematics at the Hong Kong Baptist University. He obtained his BSc degree in 1990 and MPhil degree in 1992 at the University of Hong Kong and a PhD degree in 1995 at Chinese University of Hong Kong. As an applied mathematician, his main research areas include bioinformatics, data mining, operations research and scientific computing. He has published and edited 5 books and published more than 160 journal papers. Currently, he serves on several editorial boards of international journals.

Liping Jing received his BS and MS degrees in Computer Science from Beijing Jiaotong University, China in 2000 and 2003, respectively, and her PhD in Applied Mathematics from the University of Hong Kong in 2007. From 2007 to 2008, she worked as a Postdoctoral Research Associate in the Department of Mathematics at Hong Kong Baptist University and the Laboratory for Bioinformatics and Medical Informatics in the Department of Computer Science at the University of Texas at Dallas. Currently, she is an Assistant Professor in the Department of Computer Science, Beijing Jiaotong University. Her research interests include data mining, text mining, bioinformatics and business intelligence. She has published more than 20 peer-reviewed research papers in various journals and conferences.
1 Introduction

Since first published in 1997, the $k$-modes algorithm (Huang, 1997; Huang, 1998) has become a popular technique in solving categorical data clustering problems in different application domains (e.g. Andreopoulos, An and Wang, 2005; Manganaro et al., 2005). The $k$-modes algorithm extends the $k$-means algorithm (Jain and Dubes, 1988) by using a simple matching dissimilarity measure for categorical objects, modes instead of means for clusters, and a frequency-based method to update modes in the clustering process to minimise the clustering cost function. These extensions have removed the numeric-only limitation of the $k$-means algorithm and enable the $k$-means clustering process to be used to efficiently cluster large categorical data sets from real world databases. An equivalent non-parametric approach to deriving clusters from categorical data is presented in Chaturved, Green and Carroll (2001). A note in Huang and Ng (2003) discusses the equivalence of the two independently developed $k$-modes approaches. Furthermore, Huang and Ng introduced the fuzzy $k$-modes algorithm (Huang and Ng, 1999). The fuzzy $k$-modes algorithm generates the fuzzy partition matrix from categorical data with the framework of the fuzzy $k$-means type algorithm (Bezdek et al., 2005), and improves on the $k$-modes algorithm by assigning confidence degrees to data in different clusters.

The distance between two objects computed with the simple matching similarity measure is either 0 or 1. This often results in clusters with weak intra-similarity. Recently, we studied a dissimilarity measure to the $k$-modes clustering process to improve the accuracies of the clustering results (Ng et al., 2007). The main idea is to use the relative attribute frequencies of the cluster modes in the similarity measure in the $k$-modes objective function. This modification allows the algorithm to recognise a cluster with weak intra-similarity, and therefore assign less similar objects to such cluster, so that the generated clusters have strong intra-similarities. Experimental results in Ng et al. (2007) have shown that the modified $k$-modes algorithm is very effective.

The aim of this article is to study a new fuzzy $k$-modes clustering algorithm. The proposed algorithm modifies the dissimilarity measure (Ng et al., 2007) in the fuzzy $k$-modes (Huang and Ng, 1999). We rigorously derive the updating formula for the object cluster membership assignment and cluster prototype, and prove the convergence of the algorithm under the new dissimilarity measure indeed minimise the objective function. Several experiments are conducted to show the clustering performance of the proposed algorithm is better than those of the other existing $k$-modes algorithms.

The outline of this article is as follows. In Section 2, we review the fuzzy $k$-modes algorithm. In Section 3, we study and analyse the new fuzzy $k$-modes algorithm. In Section 4, examples are given to illustrate the effectiveness of the proposed fuzzy $k$-modes algorithm. Finally, concluding remarks are given in Section 5.

2 Fuzzy $k$-modes algorithm

We assume the set of objects to be clustered is stored in a database table $T$ defined by a set of attributes, $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$. Each attribute $\mathcal{A}_i$ describes a domain of values, denoted by $\text{DOM}(\mathcal{A}_i)$, associated with a defined semantic and a data type. In this article, we only consider two general data types, numeric (ordinal) and categorical (nominal) and assume
A new fuzzy k-modes clustering algorithm for categorical data

107

other types used in database systems to be mapped to one of these two types. The domains of attributes associated with these two types are called numeric and categorical, respectively. A numeric domain consists of real numbers. A domain DOM($\theta$) is defined as categorical if it is finite and unordered, for example for any $a, b \in$ DOM($\theta$), either $a = b$ or $a \neq b$, see for instance (Gowda and Diday, 1991).

An object $X$ in $T$ can be logically represented as a conjunction of attribute-value pairs $[\theta^1 = x_1] \land [\theta^2 = x_2] \land \ldots \land [\theta^m = x_m]$ where $x_j \in$ DOM($\theta$) for $1 \leq j \leq m$. Without ambiguity, we represent $X$ as a vector $[x_1, x_2, \ldots, x_n]$. $X$ is called a categorical object if it has only categorical values. We consider every object has exactly $m$ attribute-values. If the value of an attribute $\theta^j$ is missing, then we denote the attribute-value of $\theta^j$ by $\epsilon$.

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of $n$ objects. Object $X_i$ is represented as $[x_{i1}, x_{i2}, \ldots, x_{im}]$. We write $X_i = X_k$ if $x_{ij} = x_{kj}$ for $1 \leq j \leq m$. The relation $X_i = X_k$ does not mean that $X_i$ and $X_k$ are the same object in the real world database, but rather that the two objects have equal values in attributes $\theta^1, \theta^2, \ldots, \theta^m$.

The fuzzy k-modes algorithm is to cluster the data $X$ into $k$ clusters by minimising the function (Huang and Ng, 1999)

$$F(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{il} d(Z_l, X_i),$$

subject to

$$0 \leq w_{il} \leq 1; \quad 1 \leq l \leq k, 1 \leq i \leq n,$$

$$\sum_{l=1}^{k} w_{il} = 1, \quad 1 \leq i \leq n,$$

and

$$0 < \sum_{i=1}^{n} w_{il} < n, \quad 1 \leq l \leq k.$$

while $\bullet_{il}$ is the membership degree of data $X_i$ to the $l$th cluster, and is an element of the $(k \times n)$ partition matrix $W = [\bullet_{il}]$. $C = [C_1, C_2, \ldots, C_k, \ldots, C_k]$, and $C_l$ is the $l$th cluster centre with the categorical attributes $\theta^1, \theta^2, \ldots, \theta^m$. The parameter $\alpha$ controls the fuzziness of the membership of each object.

To cluster categorical data, the fuzzy k-modes algorithm extends the hard k-modes algorithm based on the fuzzy c-means type procedure. It updates the cluster centres at each iteration by measuring the distance between each cluster centroid and each object. In Huang and Ng (1999), Huang and Ng used a simple matching dissimilarity measure for categorical objects as follows. Here, let $X$ and $Y$ be two categorical objects represented by $[x_1, x_2, \ldots, x_n]$ and $[y_1, y_2, \ldots, y_n]$, respectively,

$$d(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

where
\[
\delta(x_j, y_j) = \begin{cases} 
0, & x_j = y_j \\
1, & x_j \neq y_j 
\end{cases}
\]

It is easy to verify that the function \(d\) defines a metric space on the set of categorical objects. Traditionally, the simple matching approach is often used in binary variables which are converted from categorical variables (Kaufman and Rousseeuw, 1990, pp.28–29). We note that \(d\) is also a kind of generalised Hamming distance.

Minimisation of \(F(W, C)\) in Equation (1) with the constraints forms a class of constrained non-linear optimisation problems whose solutions are unknown. The usual method towards optimisation of \(F\) is to use partial optimisation for \(A\) and \(W\). When the cluster centres \(C_l = [\mathcal{H}_{1l}, \mathcal{H}_{2l}, \ldots, \mathcal{H}_{pl}, \ldots, \mathcal{H}_{ql}]\) (for all \(l, 1 \leq l \leq k\)) are given, the cluster membership of each object is updated by

\[
w_{il} = \begin{cases} 
1, & \text{if } X_i = Z_l \\
0, & \text{if } X_i = Z_h, h \neq l \\
\frac{1}{\sum_{h=1}^{k} \frac{d(Z_l, X_i)}{d(Z_h, X_i)}} & \text{if } X_i \neq Z_l \text{ and } X_i \neq Z_h, 1 \leq h \leq k.
\end{cases}
\]

When the cluster membership \(W\) are fixed, the cluster centres can be updated by

\[
Z_l = \hat{d}^{(t)} \in \text{DOM}(A_j)
\]

where

\[
\sum_{i, k_j = a_j} w_{il}^{(t)} \geq \sum_{i, k_j = a_j} w_{il}^{(t)}, 1 \leq t \leq n_j.
\]

The whole process is formalised in the fuzzy \(k\)-modes algorithm as follows:

**Algorithm.** The Fuzzy \(K\)-modes Algorithm

1. Randomly assign a cluster label to each object, that is initialise the cluster membership \(W^{(1)} \in \mathbb{R}^{kn}\). Determine \(C^{(1)}\) such that \(F(W^{(1)}, C^{(1)})\) is minimised. Set \(t = 1\).

2. Determine \(W^{(t+1)}\) such that \(F(W^{(t+1)}, C^{(0)})\) is minimised. If \(F(W^{(t+1)}, C^{(0)}) = F(W^{(t)}, C^{(0)})\), then stop; otherwise set \(t = t + 1\) and goto Step 3.

3. Determine \(C^{(t+1)}\) such that \(F(W^{(t+1)}, C^{(t+1)})\) is minimised. If \(F(W^{(t+1)}, C^{(t+1)}) = F(W^{(t)}), C^{(t+1)}\) = \(F(W^{(t+1)}, C^{(t)}), \text{then stop; otherwise goto Step 2.}\)
3 The new algorithm

In Ng et al. (2007), we presented a distance measure to evaluate the dissimilarity between the cluster centre and the object in clustering categorical data. Here, we extend this measure to fuzzy \(k\)-modes clustering in Equation (1). More precisely, the new dissimilarity measure \(d_n(A_l, X_i)\) is defined as follows:

\[
d_n(Z_1, X_i) = \sum_{j=1}^{m} \phi(Z_{i,j}, x_{i,j})
\]

where

\[
\phi(Z_{i,j}, x_{i,j}) = \begin{cases} 
1, & \text{if } Z_{i,j} \neq x_{i,j}, \\
1 - \left| c_{i,j,r} \right| / \left| c_i \right|, & \text{otherwise},
\end{cases}
\]

where \(|c_i|\) is the number of objects (in fuzzy sense) in the \(l\)th cluster, given by

\[
|c_i| = \sum_{i} w^a_{li}
\]

and \(|c_{i,j,r}|\) is the number of objects with category \(d_j^{(r)}\) of the \(j\)th attribute in the \(l\)th cluster (in fuzzy sense), given by

\[
|c_{i,j,r}| = \sum_{i, x_{i,j}=d_j^{(r)}} w^a_{li}
\]

According to the definition of \(\phi(\cdot)\), the dominant level of the mode category is considered in the calculation of the dissimilarity measure. When the mode category is 100% dominant, we have \(|c_j| = |c_{i,j,r}|\) and therefore the corresponding function value is the same as in Equation (5). In Equation (8), \(X_i\) is an object and \(C_l\) is the cluster mode. We remark that the dissimilarity measure in Equation (8) is defined only between an object and a cluster mode. When both \(X_i\) and \(C_l\) are objects (not cluster modes), we can consider one of them form a cluster of only one object and therefore \(|c_{i,j,r}| = |c_j|\) and \(\phi(a_{l,j}, x_{i,j})\) is equal to 0 or 1. It is clear that when both \(X_i\) and \(C_l\) are cluster modes, the dissimilarity measure between them is not defined.

To minimise the objective function with the above new dissimilarity measure,

\[
F_n(W,Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w^a_{li} d_n(Z_j, X_i)
\]

subject to constraints (2), (3) and (4). \(d_n(C_l, X_i)\) is given in Equation (8). We can still use the fuzzy \(k\)-modes algorithm in Section 2 to minimise \(F_n(W, C)\). However, the key issue is to derive rigorously the updating formula of the fuzzy \(k\)-modes clustering algorithm with the new dissimilarity measure.

We remark that the matrix \(W\) can be calculated according to Equation (6). Theorem 1 below shows rigorously the updating formula of \(C\) in the fuzzy \(k\)-modes clustering algorithm with the new dissimilarity measure.
Theorem 1. Let $X$ be a set of categorical objects described by categorical attributes $\theta_1, \theta_2, \ldots, \theta_m$ and $\text{DOM}(A_j) = \{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(m)}\}$, where $n_j$ is the number of categories of attribute $\theta_j$ for $1 \leq j \leq m$. Let the cluster centres $C_l$ be represented by $[z_{l,1}, z_{l,2}, \ldots, z_{l,m}]$ for $1 \leq l \leq k$. Then the quantity $\sum_{l=1}^{k} \sum_{i=1}^{n} w_{l,i}^n d_n(z_{l,i}, x_i)$ is minimised iff

$$z_{l,j} = a_j^{(r)} \in \text{DOM}(A_j)$$

(10)

where

$$\sum_{l=1, x_{l,j} = a_j^{(r)}}^{n} w_{l,i}^n \geq \sum_{l=1, x_{l,j} = a_j^{(r)}}^{n} w_{l,i}^n, \quad 1 \leq t \leq n_j, 1 \leq j \leq m.$$

Proof. For a given $W$, all the inner sums of the quantity

$$\sum_{l=1}^{k} \sum_{i=1}^{n} w_{l,i}^n d_n(z_{l,i}, x_i) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{l,i}^n \phi(z_{l,j}, x_{i,j})$$

are non-negative and independent. Minimising the quantity is equivalent to minimising each inner sum. We write the $(l,j)$th inner sum $(1 \leq l \leq k$ and $1 \leq j \leq m)$ as

$$D_{l,j} = \sum_{i=1}^{n} w_{l,i}^n \phi(z_{l,j}, x_{i,j}).$$

When $z_{l,j} = a_j^{(r)}$, we have

$$D_{l,j} = \sum_{i=1, x_{l,j} = a_j^{(r)}}^{n} w_{l,i}^n \left(1 - \frac{c_{l,j,t}}{c_l}\right) + \sum_{i=1, x_{l,j} \neq a_j^{(r)}}^{n} w_{l,i}^n$$

$$= \sum_{i=1}^{n} w_{l,i}^n - \sum_{i=1, x_{l,j} \neq a_j^{(r)}}^{n} w_{l,i}^n \left(\frac{c_{l,j,t}}{c_l}\right)$$

$$= c_l - \frac{c_l^j c_{l,j,t}}{c_l^j}$$

It is clear that $\mathcal{G}_{l,j}$ is minimised iff $c_{l,j,t}$ is maximal for $1 \leq t \leq n_j$. Thus,

$$\sum_{i, x_{l,j} = a_j^{(r)}} w_{l,i}^n \geq \sum_{i, x_{l,j} \neq a_j^{(r)}} w_{l,i}^n, \quad 1 \leq t \leq n_j.$$ 

The result follows.

According to Equation (10), the category of attribute $\theta_j$ of the cluster mode $C_l$ is determined by the mode of categories of attribute $\theta_j$ in the set of objects belonging to cluster $l$. It is also clear that the formulae in the results of Theorem 1 is the same as in Equation (7).

By comparing the formula in Equations (10) and (7), the cluster centres $C$ are updated in the same manner even we use different distance functions in Equations (5) and
A new fuzzy k-modes clustering algorithm for categorical data

(8), respectively. It implies that the same fuzzy k-mode algorithm can be used. The only difference is that we need to count and store \(|e_{l,j}|\) and \(|e|\) in each iteration for the distance function evaluation.

Combining Theorems 1 and (6) with the algorithm forms the fuzzy k-modes algorithm with the new dissimilarity measure, in which the modes of clusters in each iteration are updated according to Equation (10) and the partition matrix is computed according to Equation (6). We remark that the updating formulae of \(W\) and \(C\) in Equations (6) and (10), respectively, are determined by solving two minimisation subproblems from Equation (9):

\[
\min_{W} F_n(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{il} d_n(Z_{l,i}, X_i) \quad \text{for a given } W
\]

and

\[
\min_{Z} F_n(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{il} d_n(Z_{l,i}, X_i) \quad \text{for a given } Z
\]

The convergence of the fuzzy k-modes algorithm with the new dissimilarity measure can be obtained as in Theorem 2 below.

**Theorem 2.** Let \(\alpha \geq 1\), the fuzzy k-modes algorithm with the new dissimilarity measure converges in a finite number of iterations.

**Proof.** We first note that there are only a finite number \(\left(N = \prod_{j=1}^{m} n_j\right)\) of possible cluster centres (modes). We then show that each possible centre appears at most once by the fuzzy k-modes algorithm. Assume that \(C^{(1)} = C^{(2)}\) where \(t_1 \neq t_2\). According to the fuzzy k-modes algorithm, we can compute the minimisers \(W^{(1)}\) and \(W^{(2)}\) for \(C = C^{(1)}\) and \(C = C^{(2)}\), respectively. Therefore, we have

\[
F_n(W^{(t_1)}, Z^{(t_1)}) = F_n(W^{(t_2)}, Z^{(t_1)}) \geq F_n(W^{(t_2)}, Z^{(t_2)})
\]

and

\[
F_n(W^{(t_2)}, Z^{(t_2)}) = F_n(W^{(t_2)}, Z^{(t_1)}) \geq F_n(W^{(t_1)}, Z^{(t_1)})
\]

It implies that \(F_n(W^{(t_1)}, C^{(1)}) = F_n(W^{(t_2)}, C^{(2)})\). However, the sequence \(F_n(W^{(t_i)}, Z)\) generated by the fuzzy k-modes algorithm with the new dissimilarity measure is strictly decreasing. Hence, the result follows.

The result of Theorem 2 guarantees the decrease of the objective function values with respect the iterations of the fuzzy k-modes algorithm with the new dissimilarity measure.

### 4 Experimental results

In this section, we evaluate the performance and efficiency of the proposed fuzzy k-modes algorithm (FuzzyNewKM). We compare the proposed algorithm with the fuzzy k-modes (FuzzyKM) in Huang and Ng (1999), non-fuzzy k-modes (HardKM) in Huang (1998) and non-fuzzy k-modes (HardNewKM) in (Ng et al., 2007). We carried out
several tests of these algorithms on both real and artificial data sets. The experimental results are summarised and discussed in the following subsection.

4.1 Clustering performance

Four real data sets as shown in Table 1 were used to evaluate the clustering performance. These four data sets are from UCI machine learning repository (UCI, http://mlearn.ics.uci.edu/MLSummary.html). For soybean disease data set, we only selected 21 attributes from total 35 attributes because the other 14 have only one category. Each record is labelled as one of the four diseases – Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. Except for Phytophthora Rot which has 17 records, all other diseases have ten records each. The zoo data set contains 101 records, where each record represents an animal with 15 categorical attributes. Each animal data point is classified into seven classes according to its type (e.g. mammal or bird). The third data set, credit, contains 653 applicants data for credit-approval. Each application is described by nine attributes and classified as approved or rejected (\(k = 2\)). The forth data set, the adult set, contains 30,162 records, where each record represent a personal income with eight categorical attributes (\(k = 2\)).

A clustering result was measured by the clustering accuracy \(r\) defined as:

\[
    r = \frac{\sum_{l=1}^{k} a_l}{n}
\]

where \(a_l\) was the number of objects occurring in both cluster \(l\) and its corresponding class and \(n\) was the number of objects in the data set. In the fuzzy-type algorithms (FuzzyNewKM and FuzzyKM), we associate the object to the cluster based on \(\Phi\). In particular, we assign the \(i\)th object to the \(l\)th cluster if

\[ w_{li} \geq w_{ji} \quad \forall 1 \leq j \leq k. \]

Each algorithm was run 100 times on each data set. For FuzzyNewKM and FuzzyKM, the fuzzy parameter is set to be 1.1 that is the same as the tested value in Huang and Ng (1999). Table 2 shows the clustering results evaluated by accuracy \(r\). The four numbers in each cell represent the average, standard deviation, minimum and maximum of the clustering accuracy values in the 100 runs, respectively. From the table, we can see that the proposed fuzzy \(k\)-modes with new dissimilarity measure performs much better than other three clustering algorithms. In the table, the bold-faced numbers refer to the best clustering result obtained by one of the testing algorithms. We see from the table that the performance of the proposed algorithm is better than the other methods.

<table>
<thead>
<tr>
<th>Name</th>
<th>(n)</th>
<th>(m)</th>
<th>(k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybem</td>
<td>47</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>Zoo</td>
<td>101</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>Credit</td>
<td>653</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Adult</td>
<td>3,0162</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 2 The summary result for 100 runs of four algorithms on the real data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Evaluation</th>
<th>FuzzyNewKM</th>
<th>FuzzyKM</th>
<th>HardNewKM</th>
<th>HardKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybean</td>
<td>Average</td>
<td>1.00000</td>
<td>0.87287</td>
<td>0.93245</td>
<td>0.85532</td>
</tr>
<tr>
<td></td>
<td>Standard derivation</td>
<td>0.00000</td>
<td>0.14359</td>
<td>0.12433</td>
<td>0.14616</td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>1.00000</td>
<td>0.61702</td>
<td>0.61702</td>
<td>0.55319</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.82178</td>
<td>0.74070</td>
<td>0.71518</td>
<td>0.70198</td>
</tr>
<tr>
<td>Zoo</td>
<td>Standard derivation</td>
<td>0.02451</td>
<td>0.03028</td>
<td>0.10371</td>
<td>0.10296</td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>0.74356</td>
<td>0.69406</td>
<td>0.49505</td>
<td>0.45545</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>0.84248</td>
<td>0.80297</td>
<td>0.83069</td>
<td>0.78079</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.80293</td>
<td>0.76106</td>
<td>0.65971</td>
<td>0.65942</td>
</tr>
<tr>
<td>Credit</td>
<td>Standard derivation</td>
<td>0.11270</td>
<td>0.11888</td>
<td>0.12837</td>
<td>0.13705</td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>0.60365</td>
<td>0.54671</td>
<td>0.52680</td>
<td>0.50230</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>0.87371</td>
<td>0.86371</td>
<td>0.79939</td>
<td>0.82389</td>
</tr>
<tr>
<td>Adult</td>
<td>Average</td>
<td>0.74858</td>
<td>0.68223</td>
<td>0.62794</td>
<td>0.57169</td>
</tr>
<tr>
<td></td>
<td>Standard derivation</td>
<td>0.01069</td>
<td>0.00351</td>
<td>0.08982</td>
<td>0.06412</td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>0.73072</td>
<td>0.67287</td>
<td>0.51144</td>
<td>0.50527</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>0.76037</td>
<td>0.69117</td>
<td>0.74816</td>
<td>0.69989</td>
</tr>
</tbody>
</table>

Table 3 shows the precision and recall for each pair of cluster and the corresponding class. There are two numbers in each entry of the table, the first number refers to the precision and the second number refers to the recall. Their definitions are given as follows:

\[
\text{Precision} = \frac{|\text{Cluster}_i \cap \text{Class}_j|}{|\text{Cluster}_i|} \quad \text{and} \quad \text{Recall} = \frac{|\text{Cluster}_i \cap \text{Class}_j|}{|\text{Class}_j|}
\]

where \(|\text{Cluster}_i|\) is the number of objects in the \(i\)th cluster and \(|\text{Class}_j|\) is the number of objects in the \(j\)th class, and \(|\text{Cluster}_i \cap \text{Class}_j|\) is the number of objects in the both \(i\)th cluster and \(j\)th class. It is clear from the table that the results by the proposed algorithm is quite good. We can match the determined clusters with the corresponding classes quite well. The precision and the recall in such matching are quite close to 1 for some entries in the table. In particular, the precision and the recall are exactly equal to 1 in the clustering results of soybean data set.

Figures 1–4 show the distributions of the number of runs with respect to the number of records correctly classified by each algorithm for soybean, zoo, credit and adult data sets, respectively. In these figures, it is clear that the overall clustering performance of the proposed fuzzy \(k\)-modes with new dissimilarity measure is better than those of the other three algorithms. For instance, the numbers of runs with correct classifications of more than 40 objects (i.e. \(r > 0.85106\)) in soybean data set for FuzzyNewKM and HardNewKM are larger than those for FuzzyKM and HardKM. Meanwhile, the number of runs with all correct classifications (i.e. the number of records correctly clustered is 47) from FuzzyNewKM is much larger than those from other three algorithms. The FuzzyNewKM algorithm also gives many runs with correct classification in between 80
and 85 objects in zoo data set, but the other algorithms give only a few runs for such correct classification. Similarly, the FuzzyNewKM algorithm gives more than runs with correct classification above 550 (2.2 × 10⁵) objects in credit data set (adult data set) than the other algorithms.

Table 3 The precision and recall

<table>
<thead>
<tr>
<th>Soybean data set</th>
<th>Cluster1</th>
<th>Cluster2</th>
<th>Cluster3</th>
<th>Cluster4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class1</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class2</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class3</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class4</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zoo data set</th>
<th>Cluster1</th>
<th>Cluster2</th>
<th>Cluster3</th>
<th>Cluster4</th>
<th>Cluster5</th>
<th>Cluster6</th>
<th>Cluster7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class1</td>
<td>1.0000</td>
<td>0.0500</td>
<td>0.2000</td>
<td>0.2222</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0769</td>
</tr>
<tr>
<td>Class2</td>
<td>0.0000</td>
<td>0.8000</td>
<td>0.0000</td>
<td>0.1111</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1000</td>
</tr>
<tr>
<td>Class3</td>
<td>0.0000</td>
<td>0.0500</td>
<td>0.8000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class4</td>
<td>0.0000</td>
<td>0.2000</td>
<td>0.8000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class5</td>
<td>0.0000</td>
<td>0.1000</td>
<td>0.0000</td>
<td>0.5556</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0769</td>
</tr>
<tr>
<td>Class6</td>
<td>0.0000</td>
<td>0.1538</td>
<td>0.0000</td>
<td>0.7692</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0769</td>
</tr>
<tr>
<td>Class7</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.1250</td>
</tr>
<tr>
<td>Class8</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Class9</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.8000</td>
</tr>
</tbody>
</table>
Table 3  The precision and recall (continued)

<table>
<thead>
<tr>
<th></th>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Credit data set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class1</td>
<td>0.7969</td>
<td>0.1933</td>
</tr>
<tr>
<td></td>
<td>0.8572</td>
<td>0.1429</td>
</tr>
<tr>
<td>Class2</td>
<td>0.2031</td>
<td>0.8104</td>
</tr>
<tr>
<td></td>
<td>0.2635</td>
<td>0.07365</td>
</tr>
<tr>
<td><strong>Adult data set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class1</td>
<td>0.9002</td>
<td>0.5033</td>
</tr>
<tr>
<td></td>
<td>0.7482</td>
<td>0.2518</td>
</tr>
<tr>
<td>Class2</td>
<td>0.0998</td>
<td>0.4967</td>
</tr>
<tr>
<td></td>
<td>0.2503</td>
<td>0.7497</td>
</tr>
</tbody>
</table>

Figure 1  Distributions of the number of runs with respect to the number of the correctly clustered objects in each run for soybean data set using (a) FuzzyNewKM; (b) FuzzyKM; (c) HardNewKM; (d) HardKM (see online version for colours)
Figure 2  Distributions of the number of runs with respect to the number of the correctly clustered objects in each run for zoo data set using (a) FuzzyNewKM; (b) FuzzyKM; (c) HardNewKM; (d) HardKM (see online version for colours)

Figure 3  Distributions of the number of runs with respect to the number of the correctly clustered objects in each run for credit data set using (a) FuzzyNewKM; (b) FuzzyKM; (c) HardNewKM; (d) HardKM (see online version for colours)
4.2 Efficiency

The purpose of the second experiment is to test the efficiency of the proposed fuzzy k-modes algorithm when clustering large categorical data sets. Synthetic categorical data sets are generated by the method in Huang and Ng (1999). The number of clusters, attributes and categories of synthetic data ranges between 3 and 24. The number of objects ranges between 10,000 and 80,000. The computational results are performed by using a machine with an IBM laptop with 1.5G CPU and 1G RAM.

The computational times of FuzzyNewKM and HardNewKM are plotted with respect to the number of clusters, categories, attributes and objects, while the other corresponding parameters are fixed. All the experiments are repeated ten times and the average computational times are depicted.

Figure 5(a) shows the computational times against the number of clusters, while the numbers of categories and attributes are 12 and the number of objects is 80,000. Figure 5(b) shows the computational times against the number of categories, while the numbers of clusters is 3, the number of attributes is 12 and the number of objects is 80,000. Figure 5(c) shows the computational times against the number of attributes, while the numbers of clusters is 3, the number of categories is 12 and the number of objects is 80,000. Figure 5(d) shows the computational times against the number of objects, while the numbers of categories and attributes are 12 and the number of clusters is 3.
Figure 5  Computational times with respect to (a) numbers of clusters; (b) number of categories; (c) number of attributes; (d) number of objects (see online version for colours)

From these four subfigures, we can see that the proposed algorithm is scalable, that is the computational times increase linearly with respect to either the number of clusters, categories, attributes or objects. Meanwhile, we find that FuzzyNewKM requires less computational times than HardNewKM. We note that in both algorithms, the new dissimilarity measures are used accordingly. Also, the cost of each iteration of FuzzyNewKM per iteration is larger than that of each iteration of HardNewKM. As the number of iterations required by FuzzyNewKM is much less than that required by HardNewKM, the total computational time of using FuzzyNewKM is less than that of using HardNewKM. Similar results are observed in the experiment of real data sets.

5 Conclusions

In this article, we proposed a fuzzy $k$-modes clustering algorithm with new dissimilarity measure. Furthermore, we rigorously derive the updating formula of the proposed algorithm and the convergence of the algorithm under the optimisation framework. Experimental results show that the proposed fuzzy $k$-mode algorithm with the new dissimilarity measure is efficient and effective in clustering categorical data sets.
References
UCI Available at: http://mlearn.ics.uci.edu/MLSummary.html.
CALL FOR PAPERS

International Journal of High Performance Systems Architecture (IJHPSA)

Website: www.inderscience.com
ISSN (Online): 1751-6536
ISSN (Print): 1751-6528

IJHPSA proposes and fosters discussion on all aspects of the design and implementation of high-performance architectures, which are centred around the concept of parallel processing. The journal will cover all types of advanced architectures ranging from pipelined structures, array processors and multiprocessor systems. Dedicated high-performance architectures and systems, as well as hardware and software design methods and tools, will also fall within the scope of IJHPSA.

Objectives

IJHPSA provides a lively forum for the communications of original research. It acts as a scientific and professional journal to timely disseminate original research work on advanced systems architecture, which is at the heart of high-performance and cost-effective systems.

Readership

The audience for IJHPSA consists of researchers, academics, graduate students, professionals and system engineers working and interested in high-performance systems and their underlying advanced architectures.

Contents

IJHPSA publishes original peer-reviewed papers, which include research papers, review papers, case studies, book reviews. Special Issues devoted to important topics on advanced systems architecture will occasionally be published.

Subject Coverage

Suitable topics include, but are not limited to:

- **Pipelining**
  - Pipelined architectures
  - Vectorisation methods
  - Vector processing
  - Dynamic pipelines
  - Pipeline chaining

- **Multiprocessor Systems**
  - Multiprocessor architectures
  - Tightly and loosely-coupled multiprocessors
  - Interconnection networks
  - Parallel memory organisation and multicache
  - Multiprocessing control

- **Array Processors**
  - Interconnections network topologies, such as systolic, mesh, cube, etc.
  - Parallel algorithms for array processors
  - Associative array processors
  - Masking and routing mechanisms

- **RISC Architectures**
  - Instruction set
  - Pipelined RISC architectures

- **Data Flow Systems**
  - Data-driven computing
  - Data flow architectures

- **VLSI Architectures**
  - VLSI computing structures
  - Dedicated VLSI architectures

Performance evaluation and new trends

Specific Notes for Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere.

All papers are refereed through a double blind process. A guide for authors, sample copies and other relevant information for submitting papers are available at www.inderscience.com/papers

All papers must be submitted online.

To submit a paper, please go to Online Submission of Papers.

Editor in Chief: Dr. Nadia Nedjah