Quick Granular Rule Generation from a Database

Yan-Qing Zhang, G. Scott Owen and Rajshekhar Sunderraman

Department of Computer Science
Georgia State University
Atlanta, GA 30303
U.S.A.

Abstract

In this paper, a quick granular knowledge discovery method is proposed to speed up KDD process. Conventional neural-networks-based KDD methods have a major speed bottleneck problem because a neural network needs a long learning time to discover useful knowledge from huge amounts of data. To solve the speed bottleneck problem, the new KDD method can directly calculate all parameters related to granular knowledge by solving mathematical equations rather than train the parameters by using a slow neural network. In addition, the new granular knowledge discovery method can construct a meaningful granular system with the near-optimal number of granular rules for any given error. Meaningful input and output membership functions can be generated. The 2-dimensional function simulations have indicated that the new KDD method is able to compress a database to a smaller granular rule base efficiently and quickly. In the future, the n-dimensional function simulations will be investigated.

1 Introduction

Granular computing is a a superset of the theory of fuzzy information granulation, rough set theory and interval computations, and is a subset of granular mathematics [2][3][8]. Because Knowledge Discovery and Data Mining (KDDM) is related to data classification, information granulation and knowledge extraction, granular computing techniques have been used in KDDM in recent years [2][3][6][7][9][12][13]. For example, the granular neuro-fuzzy technique was used in fuzzy knowledge compression to reduce the number of fuzzy rules [13]. The compensatory genetic fuzzy neural networks were used to learn fuzzy rules from data sets and find new fuzzy rules [10]. However, they could not always discover meaningful granular knowledge from data because trained fuzzy sets could not have reasonable interpretation of linguistic terms and complete partition of a universe of discourse. Therefore, conventional methods such as fuzzy neural networks and genetic neural networks may generate unreasonable granular knowledge. In addition, various training algorithms such as gradient-descent algorithms, genetic algorithms and neural-nets-based algorithms cannot effectively find an appropriate number of fuzzy rules and optimal or near-optimal fuzzy rules because they may reach local minima because of a huge search space and randomness [11][12][15]. In this paper, we propose an efficient granular knowledge discovery method algorithm to directly model a reasonable 2-input-1-output granular system for both given data and any required accuracy.

Section 2 introduces granular partition and granular knowledge representation. Section 3 presents a granular reasoning method to directly calculate all parameters for constructing a granular system. A typical simulation of KDDM in a relational database is given in section 4. The conclusions are given in section 5.

2 Granular Partition and Granular Knowledge

A simplified N-record relational database has three numerical fields x, y and z. How to find the granular relationship among these fields (i.e. z=f(x, y)) is the major KDDM task in this paper. Fuzzy sets are basic processing elements in various applications [1][4][5][14]. Here, fuzzy sets are used as basic granules in granular partition. Numerical values of x are partitioned into m granules A_i, numerical values of y are partitioned into m granules B_i, and numerical values of z are partitioned into m^2 granules C_i for $i = 1, 2, ..., m$ and $j = 1, 2, ..., m^2$. These granules are defined by fuzzy sets:

$$\mu_{A_1}(x) = \begin{cases} \dfrac{a_2 - x}{a_2 - a_1} & \text{for} \quad a_1 \leq x \leq a_2 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$\mu_{A_j}(x) = \begin{cases} \dfrac{x-a_{j-1}}{a_j-a_{j-1}} & \text{for} \quad a_{j-1} \le x \le a_j \\ \dfrac{a_{j+1}-x}{a_{j+1}-a_j} & \text{for} \quad a_j \le x \le a_{j+1} \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

$$\mu_{A_m}(x) = \begin{cases} \dfrac{x-a_{m-1}}{a_m-a_{m-1}} & \text{for} \quad a_{m-1} \le x \le a_m \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

$$\mu_{B_1}(y) = \begin{cases} \dfrac{ab_2-y}{b_2-b_1} & \text{for} \quad b_1 \le x \le b_2 \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

$$\mu_{B_j}(y) = \begin{cases} \dfrac{x-b_{j-1}}{b_j-b_{j-1}} & \text{for} \quad b_{j-1} \le x \le b_j \\ \dfrac{b_{j+1}-x}{b_{j+1}-b_j} & \text{for} \quad b_j \le y \le b_{j+1} \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

$$\mu_{B_m}(y) = \begin{cases} \dfrac{y-b_{m-1}}{b_m-b_{m-1}} & \text{for} \quad b_{m-1} \le y \le b_m \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

$$\mu_{C_k}(z) = \begin{cases} 1+\dfrac{z-c_k}{\eta_k} & \text{for} \quad (c_k-\eta_k) \le z \le c_k \\ 1-\dfrac{z-c_k}{\eta_k} & \text{for} \quad c_k \le z \le (c_k+\eta_k) \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where $j = 2, 3, ..., m-1$, $k = 1, 2, ..., n$, a_k, b_k and c_k are centers of these triangular membership functions of x, y and z, respectively, η_k are widths of membership functions of z.

After the granular partition, X and Y have m granular values A_i, B_i for $i = 1, 2, ..., m$, respectively, and Z has n granular values C_j for $j = 1, 2, ..., n$ and $n = m^2$. Now it is ready to establish granular rule base using these granules. The granular rule base has n granular IF-THEN rules such that

$$IF\ X\ is\ A_{\lceil \frac{k}{m} \rceil}\ and\ Y\ is\ B_{kmodm}\ THEN\ Z\ is\ C_k, \quad (8)$$

for $k = 1, 2, ..., n$ and $n = m^2$.

3 Granular Reasoning and Granular System

According to the granular reasoning method called normal fuzzy reasoning method [10], a new 2-input-1-output granular system $g(x, y)$ can be made step by step as follows:

Step 1: In general, four granular rules are fired when $x \in [a_k, a_{k+1}]$ and $y \in [b_k, b_{k+1}]$, four relevant fuzzy membership functions are

$$\alpha = \frac{a_{k+1}-x}{a_{k+1}-a_k}, \quad (9)$$

$$\bar{\alpha} = \frac{x-a_k}{a_{k+1}-a_k} = 1-\alpha, \quad (10)$$

$$\beta = \frac{a_{k+1}-x}{a_{k+1}-a_k}, \quad (11)$$

$$\bar{\beta} = \frac{x-a_k}{a_{k+1}-a_k} = 1-\beta. \quad (12)$$

we have four strengths of the four fired rules:

$$\lambda_1 = \frac{\alpha+\beta}{2(\alpha+\bar{\alpha}+\beta+\bar{\beta})} = \frac{\alpha+\beta}{4}, \quad (13)$$

$$\lambda_2 = \frac{\alpha+\bar{\beta}}{2(\alpha+\bar{\alpha}+\beta+\bar{\beta})} = \frac{\alpha+\bar{\beta}}{4}, \quad (14)$$

$$\lambda_3 = \frac{\bar{\alpha}+\beta}{2(\alpha+\bar{\alpha}+\beta+\bar{\beta})} = \frac{\bar{\alpha}+\beta}{4}, \quad (15)$$

$$\lambda_4 = \frac{\bar{\alpha}+\bar{\beta}}{2(\alpha+\bar{\alpha}+\beta+\bar{\beta})} = \frac{\bar{\alpha}+\bar{\beta}}{4}. \quad (16)$$

Step 2: four contributions of the four fired rules are

$$z_1 = c_1 + \frac{w_1\eta_1(1-\alpha)+w_2\eta_1(1-\beta)}{2}, \quad (17)$$

$$z_2 = c_2 + \frac{w_1\eta_2(1-\alpha)+w_2\eta_2\beta}{2}, \quad (18)$$

$$z_3 = c_3 + \frac{w_1\eta_3\alpha+w_2\eta_3(1-\beta)}{2}, \quad (19)$$

$$z_4 = c_4 + \frac{w_1\eta_4\alpha+w_2\eta_4\beta}{2}, \quad (20)$$

where $w_1, w_2 \in \{-1, +1\}$.

Step 3: the granular system $g(x, y)$ is given by

$$g(x,y) = \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3 + \lambda_4 z_4. \quad (21)$$

For clarity, we have

$$g(x,y) = \lambda_1 c_1 + \lambda_2 c_2 + \lambda_3 c_3 + \lambda_4 c_4 + \theta_1\eta_1 + \theta_1\eta_1 + \theta_1\eta_1 + \theta_1\eta_1, \quad (22)$$

where

$$\theta_1 = \frac{\lambda_1[w_1\eta_1(1-\alpha)+w_2\eta_1(1-\beta)]}{2}, \quad (23)$$

$$\theta_2 = \frac{\lambda_2[w_1\eta_1(1-\alpha)+w_2\eta_1\beta]}{2}, \quad (24)$$

$$\theta_3 = \frac{\lambda_3[w_1\eta_1\alpha+w_2\eta_1(1-\beta)]}{2}, \quad (25)$$

$$\theta_4 = \frac{\lambda_4[w_1\eta_1\alpha+w_2\eta_1\beta]}{2}. \quad (26)$$

4 Constructive Granular Computation of Parameters

Instead of training a neural network to optimize parameters (i.e., weights) slowly, we propose a constructive granular computation method which can directly calculate all parameters by solving related mathematical equations. The major advantages are (1) optimizing parameters very quickly and (2) $g(x, y)$ can approximate any continuous function $f(x, y)$.

The N-record relational database has N data sets $(x_{\lceil frackm \rceil},\ y_{kmodm},\ z_k)$ for $k = 1, 2, ..., N$. Now an optimization function is given below,

$$Q = \sum_{i=1}^{m}\sum_{j=1}^{m}[z_{m\lceil frackm \rceil + kmodm} - g(x_i, y_j)]^2. \qquad (27)$$

To minimize Q, we have

$$\begin{cases} \frac{\partial Q}{\partial c_1} = 0, \\ \frac{\partial Q}{\partial c_2} = 0, \\ \frac{\partial Q}{\partial c_3} = 0, \\ \frac{\partial Q}{\partial c_4} = 0, \\ \frac{\partial Q}{\partial \eta_1} = 0, \\ \frac{\partial Q}{\partial \eta_2} = 0, \\ \frac{\partial Q}{\partial \eta_3} = 0, \\ \frac{\partial Q}{\partial \eta_4} = 0. \end{cases} \qquad (28)$$

Then we can solve the above 8-parameter linear equation to get optimal values of c_1, c_2, c_3, c_4, η_1, η_2, η_3, η_4 by initializing all possible values of w_1 and w_2. According to divide-and-conquer technique, we divide a whole 2-dimensional space of (x, y) into m^2 subspaces. Each subspace is related to the 8-parameter linear equation. After solving m^2 8-parameter linear equations, we can find optimal c_1, c_2, c_3, c_4, η_1, η_2, η_3, η_4, and then build the granular system because all parameters for granular rules have been found.

5 Granular KDDM Simulations

To do granular KDDM simulations, we create a 6561-record relational database with a tornado-type mapping relationship between two input fields and one output field. The tornado-type function is defined below:

$$f(x, y) = 2 - \frac{2sin(\sqrt[4]{(3x^2 + 4y^2)^3})}{\sqrt[4]{(3x^2 + 4y^2)^3}}. \qquad (29)$$

The known relational database shown in Table 1 has three fields (i.e., X, Y and Z) and 6561 records, where $Z = f(X, Y)$ shown in Fig. 1, $X = -8.0 + i/5$ and $Y = -8.0 + j/5$ for $i = 0, 1, 2, ..., 80$ and $j = 0, 2, ..., 80$. The objective of KDDM is to find a samll number of granular rules which can represent 6561 relations.

Table 1: The known 3-field relational database

X	Y	Z
-8.0	-8.0	1.99976
-8.0	-7.8	1.98168
-8.0	-7.6	2.01767
...	...	...
8.0	7.6	2.01767
8.0	7.8	1.98168
8.0	8.0	1.99976

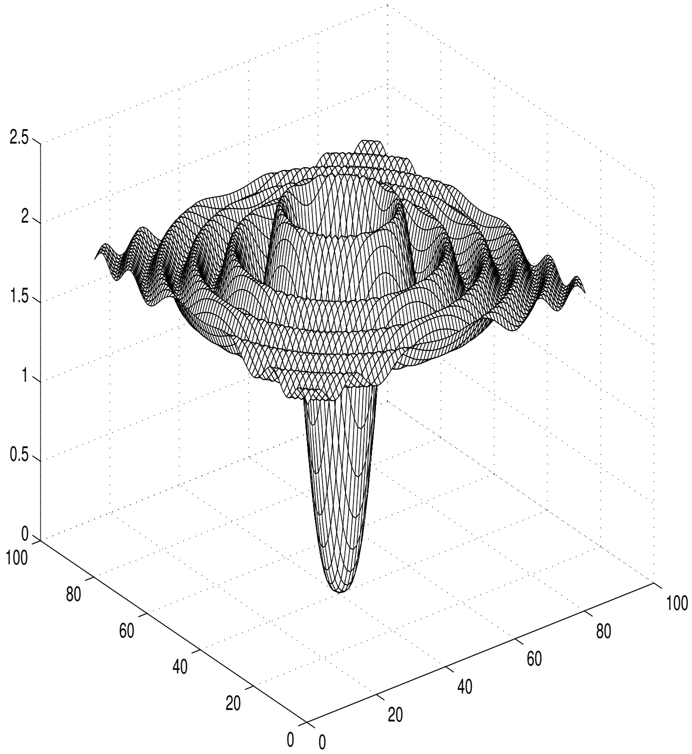

Figure 1: The tornado function $f(x_i, y_j)$.

Speed Comparison between the granular system using the granular constructive computation method and a neural network using the BP learning algorithm is the main issue in simulations. The simulation results in Table 2 can indicate that the execution time T_new of the new piecewise granular constructive approach is much smaller than the execution time T_old of a conventional BP-based neural network. The main reason is that the new granular constructive method can directly calculate parameters in a very short time, but the BP-based neural network takes a lot of time to adjust parameters gradually and slowly.

Table 2: Speed Comparison of the granular system and a neural network

Average Absolute Error	T_new(Sec.)	T_old (hours)
0.065	2	>3
0.055	2	>5
0.045	3	>8
0.035	4	>12

6 Conclusions

The new constructive granular knowledge discovery method can build a piecewise granular system quickly by directly calculating parameters of granular rules with any required accuracy. In addition, the new approach is able to generate meaningful input and output membership functions, i.e., discover understandable granular knowledge. In summary, both quick granular knowledge discovery and meaningful interpretation of granular rules are two important advantages of the new granular knowledge discovery method. In the future, an n-dimensional granular knowledge discovery method will be investigated.

References

[1] G.J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications,* Prentice Hall P T R, 1995.

[2] T.Y. Lin, "Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems," *Rough Sets in Knowledge Discovery*, pp. 107-121, A. Skoworn and L. Polkowski (eds), Springer-Verlag, 1998.

[3] T.Y. Lin, "Granular Computing on Binary Relations II: Rough Set Representations and Belief Functions," *Rough Sets in Knowledge Discovery*, pp. 121-140, A. Skoworn and L. Polkowski (eds), Springer-Verlag, 1998.

[4] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. on SMC*, Vol. SMC-15, No. 1, pp. 116-132, 1985.

[5] L.-X. Wang, *Adaptive Fuzzy Systems and Control design and stability analysis,* PTR Prentice Hall, 1994.

[6] Y.Y. Yao, "Granular computing using neighborhood systems," *Advances in Soft Computing: Engineering Design and Manufacturing*, Roy, R., Furuhashi, T., and Chawdhry, P.K. (Ed.), pp. 539-553, Springer-Verlag, London, 1999.

[7] Y.Y. Yao and N. Zhong, "Potential applications of granular computing in knowledge discovery and data mining," *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics, Volume 5, Computer Science and Engineering,* Torres, M., Sanchez, B. and Aguilar, J. (Ed.), pp. 573-580, Orlando, 1999.

[8] L. Zadeh, "http://www.mathcs.sjsu.edu/GrC /GrC_information/zadeh.txt," 1999.

[9] L.A. Zadeh, "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, Vol. 19, pp. 111-127, 1997.

[10] Y.-Q. Zhang and A. Kandel, *Compensatory Genetic Fuzzy Neural Networks And Their Applications,* World Scientific, 1998.

[11] Y.-Q. Zhang and A. Kandel, "Compensatory Neurofuzzy Systems with Fast Learning Algorithms," *IEEE Trans. on Neural Networks*, Vol. 9, No. 1, pp.83-105, 1998.

[12] Y.-Q. Zhang, M. Ma and A. Kandel, "Piecewise constructive approach to constructing fuzzy systems," Proc. of SPIE'99, Vol. 3722, pp.48-56, April 5-9, 1999.

[13] Y.-Q. Zhang, M. D. Fraser, R. A. Gagliano and A. Kandel, "Granular Neuro-fuzzy Knowledge Compression and Expansion," SCI/ISAS'99, Vol.5, pp.581-586, July 31-Aug. 4, 1999.

[14] H.-J. Zimmermann, *Fuzzy Sets Theory and Its Applications,* Kluwer Academic Publishers, 1996.

[15] Y. Zhou, Y. Lu and C. Shi, "Using Neural Network to Extract Knowledge from Database," Proc. of PKDD'97, pp. 376-383, 1997.