A Comprehensive Granular Model for Decision Making with Complex Data

Ying Xie*, Tom Johnsten, Vijay V. Raghavan,
Ryan G. Benton, and William Bush

Abstract. This chapter describes a comprehensive granular model for decision making with complex data. This granular model first uses information decomposition to form a horizontal set of granules for each of the data instances. Each granule is a partial view of the corresponding data instance; and aggregately all the partial views of that data instance provide a complete representation for the instance. Then, the decision making based on the original data can be divided and distributed to decision making on the collection of each partial view. The decisions made on all partial views will then be aggregated to form a final global decision. Moreover, on each partial view, a sequential M+1 way decision making (a simple extension of Yao's 3-way decision making) can be carried out to reach a local decision. This chapter further categorizes stock price predication problem using the proposed decision model and incorporates the MLVS model for biological sequence classification into the proposed decision model. It is suggested that the proposed model provide a general framework to address the complexity and volume challenges in big data analytics.

Ying Xie
Department of Computer Science,
Kennesaw State University, Kennesaw, Georgia 30144, USA
e-mail: yxie2@kennesaw.edu

Tom Johnsten · William Bush
School of Computing, University of South Alabama, USA
e-mail: tjohnsten@southalabama.edu,
wmb1321@jagmail.southalabama.edu

Vajay V. Raghavan
The Center of Advanced Computer Studies, University of Louisiana at Lafayette, USA
e-mail: vijay@cacs.louisiana.edu

Ryan G. Benton
Informatics Research Institute, University of Louisiana at Lafayette, USA
e-mail: rbenton@louisiana.edu

* Corresponding author.

W. Pedrycz and S.-M. Chen (eds.), *Granular Computing and Decision-Making*,
Studies in Big Data 10, DOI: 10.1007/978-3-319-16829-6_2

Keywords: Complex Data, Comprehensive Granular Model, M+1 way decision making, Big Data, Granular Computing.

1 Introduction

Granular Computing [1] focuses on philosophy, methodology and paradigm for problem solving and information processing based on granular structures [3]. By using the concepts like granules, levels, and hierarchies, granular computing promotes structured thinking at the philosophical level and structured problem solving at the practical level [2]. In [4], Y. Y. Yao stated that "the principle of computing, guided by granular structures, is to examine the problem at a finer granulation level with more detailed information when there is a need or benefit for doing so." Furthermore, Y. Y. Yao developed a sequential three-way decision framework based on a hierarchy of multiple levels of information granularity [5]. Decision tree can be viewed as a simple special case of the sequential three-way decision making.

The three-way decision framework assumes a single vertical view of the granular structures, where each layer of information granulation is a complete representation of the original data at a particular coarse level. However, for complex decision making on data that is unstructured or big, it may be challenging or even impossible to form a single hierarchy of information granules for the data in order to carry out the sequential three way decision making. In order to address this issue, this chapter describes a comprehensive granular model for decision making with complex data. This granular model first uses information decomposition to form a horizontal set of granules for each of the data instances. Each granule is a partial view of the corresponding data instance; and aggregately all the partial views of that data instance provide a complete representation for the instance. Then, the decision making based on the original data can be divided and distributed to decision making on the collection of each partial view. The decisions made on all partial views will then be aggregated to form a final global decision. If a partial view is still complex enough, then the decomposition process can be continued on the partial view to form a horizontal set of sub partial views for that partial view. This decomposition process continues until a single hierarchical structure of information granules can be formed for a (sub) partial view. Then, a sequential M+1 way decision making (an extension of 3-way decision making) can be carried out on the collection of each partial view to reach a local decision. Decision forest can be viewed as a simple special case of this comprehensive granular model for decision making.

The proposed comprehensive granular model fits well with parallel/distributed computing frameworks. Each collection of a partial view covers all data instances with respect to that partial view, and is independent from other collections of different partial views. Therefore, local decisions can be made on all collections of partial views in a parallel manner. On each collection of a partial view, the sequential M+1 way decision making process may be implemented as multiple iterations of MapReduce processes. Therefore, the proposed granular model can be viewed as a general solution framework for big unstructured data. This chapter will further categorize stock price predication problem and biological sequence classification problem using the proposed decision model.

2 A Generalized M+1 Way Decision

In [5], Yao described a three way decision model for 2-state classification problems. Compared with traditional two way (acceptance/rejection) decision model, the three way decision model maps a data instance for decision to one of the three disjoint regions, POS, NEG, and BND. Decision of "acceptance" is made on those objects mapped to the POS region; decision of "rejection" is made on those mapped to the NEG region; and "noncommitment" is assigned to those mapped to BND region. A noncommitment assignment suggests that more information of the corresponding data instance is needed in order to accept or reject that object. The essential ideas of three way decision making has been widely used in real-life decision making in different domains, such as medical decision-making, social judgment theory, and hypothesis testing, and peering review processes [6]. In order to apply this idea to multi-class classification problem, Yao's Three Way Decision Model can be naturally extended to a $M+1$ Way Decision Model. Formally, the $M+1$ Way Decision Model can be described as follows.

Given a set of data instances U with N elements $\{u_1, u_2, \dots, u_N\}$ and a set of class labels C with M elements $\{c_1, c_2, \dots, c_M\}$, the $M+1$ Way decision model divides U into $M+1$ disjoint regions $\{r_{c_1}, r_{c_2}, \dots, r_{c_M}, r_{c_{M+1}}\}$, such that if a data instance $u_i \in r_{c_j}$ (where $1 \le j \le M$), then the class label c_j is assigned to x_i; if $u_i \in r_{c_{M+1}}$, a "noncommitment" is assigned to u_i.

In [6], Yao uses a Two-Poset based evaluation to segment U into the acceptance region and the rejection region. We follow the exact same method to extend the Two-Poset based evaluation to a M-Poset based evaluation for $M+1$ way decision model.

Let $L_C = \left\{ \left(l_{r_{c_i}}, \preccurlyeq_{r_{c_i}} \right) \middle| 1 \le i \le M \right\}$ be M posets, and $V = \{v_{r_{c_i}} : U \to l_{r_{c_i}} | 1 \le i \le M\}$ be M evaluation functions. Given $x \in U \; and \; 1 \le i \le M$, $v_{r_{c_i}}(x)$ returns an acceptance value of x to r_{c_i}. For two objects $x, y \in U \; and \; 1 \le i \le M$, if $v_{r_{c_i}}(x) \preccurlyeq_{r_{c_i}} v_{r_{c_i}}(y)$, then x is less acceptable than y to r_{c_i}. Further let $1 \le i \le M \; and \; l^+_{r_{c_i}} \subseteq l_{r_{c_i}}$ be the set of designated values of acceptance to r_{c_i}. Then, given an object $x \in U$, we have $x \in r_{c_i}$ if and only if $v_{r_{c_i}}(x) \in l^+_{r_{c_i}}$. In other words, we can define r_{c_i} $(1 \le i \le M)$ as follows: $r_{c_i}(\; 1 \le i \le M) = \{x \in U | v_{r_{c_i}}(x) \in l^+_{r_{c_i}}\}$. Then we can further define $r_{c_{M+1}}$ as follows: $r_{c_{M+1}} = \{x \in U | \forall (1 \le i \le M) \to v_{r_{c_i}}(x) \notin l^+_{r_{c_i}}\}$.

3 A Generalized Sequential M+1 Way Decision Algorithm

We further generalize Yao's Sequential Three way decision algorithm [5] to Sequential $M+1$ Way Decision Algorithm. Assume for each data instance $x \in U$, there exists $n+1$ levels of granular description of x. By following the same notation used in [5], the $n+1$ level of granular description of x can be represented as follows:

$$Dec_0(x) \preccurlyeq Dec_1(x) \preccurlyeq \cdots \preccurlyeq Dec_n(x)$$

where the relation $\preccurlyeq$ denotes a "finer than" relationship. In order to make a decision on x, the Sequential $M+1$ Way Decision Algorithm first uses $Dec_n(x)$ as the representation for x. A decision is made on x by assigning the class label c_i to x, if $\exists(1 \le i \le M) \rightarrow v_{r_{c_i}}\big(Dec_n(x)\big) \in l^+_{r_{c_i}}$; otherwise, we continue to use $Dec_{n-1}(x)$ as the representation for x. If we are still unable to make decision on x even reaching the level 0 of granular description of x, we have two options. Option 1: assign "noncommitment" as the label to x as the final decision on x. This option indicates that no decision can be made at required confidence level on x based upon all information available on x. Option 2: assign the class label that corresponds to the largest $v_{r_{c_i}}\big(Dec_n(x)\big)$ $(1 \le i \le M)$. Option 2 can be applied to the situations where a data instance has to be categorized to one of the M categories.

4 Decision Making on Complex Data

The sequential M+1 way decision algorithm assumes a single vertical view of the granular structures, where each layer of information granulation is a complete representation of the original data at a particular coarse level. However, for complex decision making on data that is unstructured or big, it may be challenging or even impossible to form a single hierarchy of information granules for the data in order to carry out the sequential $M+1$ way decision algorithm. In order to address this issue, we propose a comprehensive granular model for decision making with complex data. This granular model first uses information decomposition to form a horizontal set of granules for each of the data instances. Each granule is a partial view of the corresponding data instance; and aggregately all the partial views of that data instance provide a complete representation for the instance. Then, the decision making based on the original data can be divided and distributed to decision making on the collection of each partial view. The decisions made on all partial views will then be aggregated to form a final global decision. More formally, assume that for each data instance $x \in U$, we decompose x into a set of $l \ge 1$ partial views $PV_x = \{x_1, x_2, \ldots, x_l\}$, where $PV_x[i] = x_i$. This decomposition could be lossless or lossy. For lossless decomposition, PV_x contains the same amount of information as what x contains; i.e., there exists an operator op, such that $x = agg_{op}\{x_i | x_i \in PV_x\}$. For lossy decomposition, PV_x contains less information than x. Furthermore, for each of the partial view of x, we assume there exists $n+1$ levels of granular description of the partial view; i.e., for the ith partial view of x, $PV_x[i]$, we have

$$Dec_0(PV_x[i]) \preccurlyeq Dec_1(PV_x[i]) \preccurlyeq \cdots \preccurlyeq Dec_n(PV_x[i])$$

Overall, the representation of the set of complex data instances can be illustrated in figure 1.

If a partial view is still complex enough, then the decomposition process can be continued on the partial view to form a horizontal set of sub partial views for that

partial view. This decomposition process continues until a single hierarchical structure of information granules can be formed for a (sub) partial view. For simplicity of description, we will not further illustrate this situation.

Based on this decomposition on each data instance in U, the set of all data instances U is decomposed into l sets U_1, $U_2, \dots, U_i, \dots, U_l$, where $U_i = \{PV_{u_j}[i] | u_j \in U\}$, i.e., the set of the ith partial views from all data instances in U. The decomposition on U can be illustrated in figure 2.

Then the overall decision process on U can be decomposed to a decision process on each U_i. In other words, given a data instance $u_j \in U$, we will have l completely independent decision process that can be carried out by l distributed computing processes. The final decision on u_j will be an ensemble of the results of the l independent decision processes. One possible ensemble approach can be described as follows.

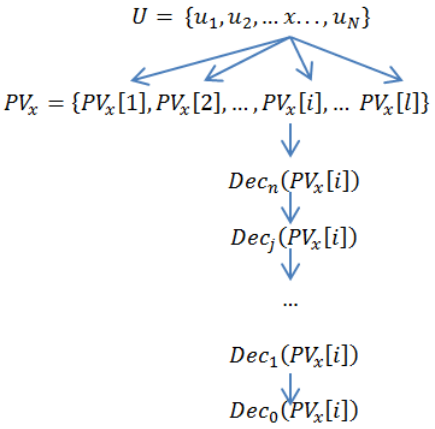

Fig. 1 An Illustration of the Proposed Model for Representing a Complex Data Set

$$U \Longrightarrow \begin{cases} U_1 = \{PV_{u_1}[1], PV_{u_2}[1], \dots, PV_{u_N}[1]\} \\ U_2 = \{PV_{u_1}[2], PV_{u_2}[2], \dots, PV_{u_N}[2] \\ \dots \\ U_i = \{PV_{u_1}[i], PV_{u_2}[i], \dots, PV_{u_N}[i] \\ \dots \\ U_l = \{PV_{u_1}[l], PV_{u_2}[l], \dots, PV_{u_N}[l] \end{cases}$$

Fig. 2 An Illustration of Decomposing a Data Set U into l sets of Partial Views

Given a data instance $u_j \in U$, assume a class label $c_k\ (1 \le k \le M)$ is assigned to u_j on $Dec_p\left(PV_{u_j}[s]\right)$ in U_s, where $\left(PV_{u_j}[s]\right)$ is the $s^{th}(1 \le s \le l)$

partial view of u_j and $Dec_p\left(PV_{u_j}[s]\right)$ is the granular representation of this partial view $PV_{u_j}[s]$ at p $(0 \le p \le n)$ level. Then the weight of this decision can be represented as $\theta_{j,s,p,k}(0 < \theta_{j,s,p,k} \le 1)$, where $\theta_{j,s,p,k} = f(j,s,p,k)$ can be evaluated by a function that reflects the confidence of the decision that assigns c_k to u_j based on U_s at the granularity level p. For instance, a possible expression of this weight function can be $\theta_{j,s,p,k} = \omega_s * (p+1)/(n+1)$, where $\omega_s(0 < \omega_s \le 1)$ reflects the relative significance of using U_s in the overall decision making; $\frac{p+1}{n+1}$ (where p is the granularity level of data representation at which the decision is made and $0, 1, \ldots, n$ are the available granularity levels from finest to coarsest) suggests that reaching decision at a courser level justifies a stronger differential power of this partial view for u_j. Therefore, the overall weight for assigning c_k to u_j can be calculated as $\sum_{1 \le s \le l} \theta_{j,s,p(s),k}$, where $p(s)$ represents the coarsest granularity level at which c_k is assigned to u_j using U_s. The class label with the highest overall weight will be the final class label that is assigned to u_j.

5 Categorizing Stock Price Prediction Using the Proposed Decision Model

As is well known, stock price prediction is a broad yet very challenging research area. In this section, we will categorize one particular stock price prediction task by using the proposed decision model. This prediction task can be described as follows. Given the historical price movement of a stock or index, such as SPY, in a particular time frame (which can be weekly, daily, hourly, 5 minutes, and so on), predict the price movement for the next time unit (next week, next day, next hour, or next 5 minutes). If we visualize the price movement at a historical time unit using a bar as shown in figure 3, the task is to predict what will be the upcoming bar.

We now use the proposed decision model to categorize this prediction task. We first annotate each historical bar by using the bar that immediately follows it. For instance, bar1 in figure 3 can be annotated as Lower Open Lower Close (LOLC), given that its following bar has a lower open and lower close compared to its own close price; bar2 can be annotated as Higher Open Higher Close (HOHC), given that its following bar has a higher open and higher close (HOHC) compared to its own close price; bar3 can be annotated as Lower Open Higher Close (LOHC), given that its following bar has a lower open and higher close compared to its own close price; and bar4 can be annotated as higher open and lower close (HOLC), given that its following bar has a higher open and lower close compared to its own close price. Therefore, we identify a set of 4 class labels {LOLC, HOHC, LOHC, HOLC}. Now, the task is to predict the class label for the current bar which is the last bar shown in figure 3. The class label assigned to the current bar is the predication of the price movement for the upcoming time unit.

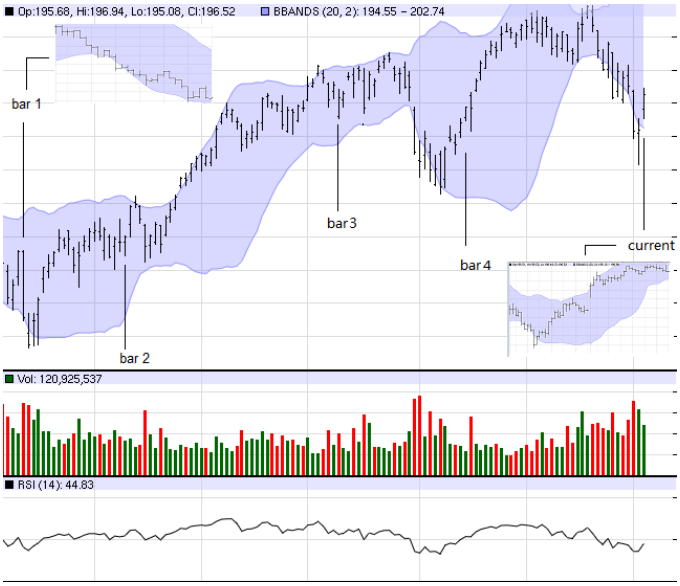

Fig. 3 A Sample Chart of Stock Price Movement (copy from www.barchart.com)

In order to predict the class label for the current bar, we need to identify features that can be used to describe each bar. This is where complexity comes into the picture, given that each historical bar could be associated with numerous factors that may indicate or correlate with the stock price movement at next time unit. Moreover, those factors may be heterogeneous in nature, formats, and scales; and some features are complex data by themselves. Therefore, it is very challenging for any typical machine learning algorithm to utilize all these features together. However, by using our proposed decision model, each of those features or a combination of a group of features can serve as a partial view for a given bar, and all partial views deliver comprehensive information for that bar. A partial view can be certain technique analysis (TA) features, sentimental features, or fundamental analysis (FA) features. Therefore, the proposed decision model provides a framework for a comprehensive analysis of stock price movement.

Without loss of generality, we only consider some technique analysis features for an illustration. These features include price movement within a time unit, price movement over a period of time, volume, RSI, and moving average. Based on these features, we form two partial views for each bar. The first partial view, denoted as $PV[1]$, is the combination of price movement within a time unit, volume, RSI, and moving average; the second partial view, denoted as $PV[2]$, is the combination of price movement over a period of time, volume, and RSI. As can be seen, $PV[1]$ is for predicting next price movement based on price movement at the current time unit; whereas $PV[2]$ is for predicting next price movement based on price movement cross multiple time units. Furthermore, for each partial view, we form multi-level descriptions with different granularities.

For instance, the multi-level granular descriptions of $PV[1]$ on bar 1 are shown below:

$Dec_2(PV_{bar1}[1])$: Price movement within a time unit – {Open High, Close Low},

Volume – {High}

RSI – {Median}

Moving average – {above moving average}

$Dec_1(PV_{bar1}[1])$: Price movement within a time unit – {High-Low, Low-Low, Open-Low, Close-Low},

Volume – {High}

RSI – {Median}

Moving average – {above moving average}

$Dec_0(PV_{bar1}[1])$: Price movement within a time unit – {time series of price movement within this time unit*},

Volume – {High}

RSI – {Median}

Moving average – {above moving average}

**The time series of price movement within the time unit for bar 1 and the current bare are illustrated in figure 3.*

Based on the multi-level granular descriptions of $PV[1]$ for each bar, a 4+1 Way Sequential Decision Algorithm can be applied in order to make a decision on the current bar. The algorithm first compare $Dec_2(PV_{current}[1])$ with each $Dec_2(PV_{bar\_x}[1])$, where bar_x is a historical bar. This comparison finds all historical bars that match the current bar on Volume, RSI, moving average, and price movement values. If at least $\alpha\%$ of all matched bars share the same class label, then decision can be reached at this level on this partial view; otherwise; the 4+1 Way Sequential Decision Algorithm goes down to the finer level of granularity to look for a decision. That is, the algorithm further compares $Dec_1(PV_{current}[1])$ with each $Dec_1(PV_{bar\_y}[1])$, where bar_y is one of the matched historical bars found by the comparison at the previous granularity level. For this comparison, we find the K nearest neighbors of the current bar by calculating the similarity on the price movement within the time unit between the current bar and each of the bars selected as a match at the previous granularity level. If at least $\alpha\%$ of the K nearest neighbors share the same class label, then decision can be reached at this level on this partial view. Otherwise, the algorithm further goes down to the finest level of granular descriptions to look for a decision by comparing $Dec_0(PV_{current}[1])$ with each $Dec_0(PV_{bar\_y}[1])$, where bar_y is one of the matched historical bars found by the comparison at the coarsest granularity level. The computing at this finest granular level is most intensive, since each comparison between $Dec_0(PV_{current}[1])$ and $Dec_0(PV_{bar\_y}[1])$

requires computing similarity between two time series by using techniques like Dynamic Time Warping (DTW) [7]. The 4+1 Way Sequential Decision Algorithm reduces the requirement of this intensive computing by 1) trying to make decision at a coarser level of granular representation; and 2) reducing the number of times of executing intensive computing by taking advantage of the results that have been generated at a coarser level.

Similarly, the multi-level granular descriptions of $PV[2]$ on bar 1 are shown below

$Dec_2(PV_{bar\_1}[2])$: Price movement for 2 most recent time units – {Close High, Close Low}*,

**bar_1 itself closed low, the previous bar closed high*

Volume – {High}

RSI – {Median}

$Dec_1(PV_{bar\_1}[2])$: Price movement for 3 most recent time units – {Close High, Close High, Close Low},

Volume – {High}

RSI – {Median}

$Dec_0(PV_{bar\_1}[2])$: Price movement for 4 most recent time units – {Close Low, Close High, Close High, Close Low},

Volume – {High}

RSI – {Median}

Again, a 4+1 Way Sequential Decision Algorithm can be carried out on $PV[2]$ in order to make a prediction on the current bar. Finally, decision made on each partial view can be assembled to form a final decision.

6 Incorporating the MLVS Model for Biological Sequence Classification into the Proposed Decision Model

Bioinformatics methods are increasingly important for biomedical research, as advances in high throughput sequencing have drastically reduced the cost per base for genomic sequencing. For the first time, we are in an era where our ability to sequence genomes is quickly outstripping our capacity to analyze them in a useful manner. Developing novel methods to detect and analyze features of interest within biological sequences is a critical step in fully utilizing the wealth of information made available by advanced sequencing techniques [8]. A significant challenge in analyzing such data is the extraction and representation of significant features from the data. To address this challenge, we recently developed a model, called Multi-Layered Vector Spaces (MLVS), for representing biological

sequences for the purpose of characterization and classification [10, 11]. Experiments show that MLVS-based classifiers are able to outperform or perform on par with existing methods for classifying biological sequences [10, 12].

The MLVS model is based on the idea of mapping biological sequences into a set of vectors. In general, each vector contains the location of h-step ordered pairs of symbols, where a symbol is an element of the alphabet from which the sequence is constructed and h represents the number of spaces between two symbols. If all ordered pairs made up of consecutive symbols of the alphabet form 1-step pairs, U_1, then allowing multiple spaces between the elements of the ordered pair generates a set of m-step pairs, $U_1, U_2, \ldots, U_k$, forming a multi-layered space. The original sequence can thus be conceptually viewed as the union of all such ordered pairs stratified at k distinct layers. The mapping of biological sequences into such a vector space has the potential to bring out subtle local patterns that may be overlooked by existing methods. We now present a formal description of the MLVS model and illustrate how it can be successfully incorporated into the proposed granular decision making model.

A sequence S of finite length $|S|$ defined over a finite alphabet $\sum$ is viewed to have a multi-layered structure made up of a set of m-step ordered pairs (i, j), with $(i, j) \in \sum$, denoted by $U_{h|(i,j)}$, where $1 \leq h \leq k$. The parameter h stands for the number of spaces between the elements of the pair downstream in the flow (left to right) of the sequence, and k is the maximum admissible value of h. Ordered pairs made up of consecutive elements of the sequence are said to form the family of 1-step (one-step) pairs, $U_{1|(i,j)}$. Allowing multiple spaces between the elements of an ordered pair generates a multitude of m-step pairs (families) $U_1, U_2, \ldots, U_k$, creating a multi-layered k-clustering C_k made up of sets $U_{h|(i,j)}$, $m = 1,2,\ldots,k$ as follows: $C_k = \cup_m \cup_{(i,j)} U_{h|(i,j)}$. The upper bound for parameter k is $|S| - 1$. The binding factor between the elements of a particular set $U_{h|(i,j)}$ is the step size h, common for all ordered pairs making up the family. The total number of ordered pairs that can be drawn from the alphabet is $|\sum|^2$. A sequence S can be viewed as the union of all such ordered pairs at k distinct layers. The following example demonstrates how the said structures are built.

Table 1 Sample Sequence

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
g	c	t	g	g	g	c	t	c	a
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
g	c	t	a	a	t	g	a	g	c

Example-1: Given the alphabet $\sum = \{a, c, g, t\}$, with $|\sum| = 4$, $|\sum|^2 = 16$, and the following biological sequence S defined over $\sum$, with $|S|$ =20: $S = [g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c]$.

Table-1 shows the step locations of the elements making up the sequence. The following are sample h-step pairs: 1-step ordered pairs for (g, c) are located at step locations $[1,2], [6,7], [11,12]$, and $[19,20]$; 1-step ordered pairs for (g, g) are located at step locations $[4,5]$, and $[5,6]$; 2-step ordered pairs for (g, t) are located at step locations $[1,3], [6,8]$, and $[11,13]$; 4-step ordered pairs for (c, g) are located at step locations $[2,6]$, and $[7,11]$. For a selected value of h and a given ordered pair $(i, j) \in \sum$, the sequence of anchor positions is taken as forming the scalar components of an q-dimensional feature vector $V_{h|(i,j)}$ associated with the ordered pair (i, j). The union of such vectors for all ordered pairs (for a given h) forms a vector cluster $\check{Z}_h$ at step size h, $\check{Z}_h = \cup_{(i,j)} V_{h|(i,j)}$, providing a single-step representation for the sequence.

The union of vector clusters $\check{Z}_h$ provides a multi-layered feature vector space $\check{Z}_k = \cup_m \cup_{(i,j)} V_{h|(i,j)}$, one layer for each value of h, for the original sequence. The grand vector space $\check{Z}_k$ provides the option of controlling the accuracy and resolution of the solution space by selecting m, the step size for ordered pairs, and q, the dimensionality of the vectors $V_{h|(i,j)}$ in an appropriate manner. Vector $V_{h|(i,j)}$, functioning as a feature vector in this paper, represents the sequential positions of the leading anchor elements of ordered pairs throughout the entire sequence.

Feature vectors for each h-step ordered pair can be structured in at least two different ways. One approach is to simply record the step (spatial index) locations of anchor positions as Boolean values $(1,0)$. This approach is suitable for collections of equal length sequences. An alternative approach is to partition a sequence into q equal segments and record the number of anchor positions that fall into each segment. The number of segments q will determine the dimension of the vectors thus formed. The size of q can be adjusted to meet restrictions or expectations on resolution and accuracy. This approach has the advantage of mapping sequences of unequal length into fixed length feature vectors.

Using the alphabet and sequence from the previous example (Table 1), the following are sample feature vectors for a select group of ordered m-step pairs: Anchor positions of 1-step ordered pairs for (g, c) are located at step (index) locations $[1,6,11,19]$; vector $V_{1|(g,c)}$, is represented by the Boolean vector $< 1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0 >$ if step anchor locations are used directly as vector components. If we instead partition the sequence into 4 equal segments ($q = 4$), the vector $V_{1|(gc)}$, is represented by the 4D vector $< 1,1,1,1 >$ with vector components representing the number of anchor elements in each segment; anchor positions of 1-step ordered pairs for (g, g) are located at step (index) locations $< 4,5 >$; vector $V_{1|(g,g)}$ is represented by the Boolean vector $< 0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 >$ or by the 4D vector $< 2,0,0,0 >$; anchor positions of 2-step ordered pairs for (g, t) are located at step (index) locations $[1,6,11]$; vector $V_{2|(g,t)}$ is represented by the Boolean vector $< 1,0,0,0,0,1\ ,0,0,0,0,1,0,0,0,0,0,0,\ 0,0,0 >$ or by the 4D vector $< 1,1,1,0 >$.

The MLVS model, with its correspondingly very large vector space, is an excellent domain in which to apply the proposed granular model for decision making. There are numerous ways in which to define partial views in the context of the MLVS model. These include the construction of partial views based on individual ordered pairs specified across one or more step sizes, and partial views based on combinations of multiple ordered pairs specified across one or more step sizes. To illustrate, consider the classification of protein sequences. Such sequences are typically defined in terms of an alphabet consisting of twenty amino acids. Thus, a given protein sequence can be transformed into four hundred MLVS feature vectors in which individual vectors correspond to a specific ordered pair / step size combination. Each vector can be regarded as a partial view and collectively all vectors for a given sequence x represent the set of partial views, $PV_x = \{x_1, x_2, \ldots, x_{400}\}$. A classification of the protein sequences can be obtained based on the output of the four hundred independent decision making processes defined by the partial views. As described in Section 4, each decision making process would start with analyzing MLVS vectors represented at a coarse level of granularity (i.e. low dimension vectors) and, if necessary, repeat the decision making process using vectors represented at finer levels of granularity (i.e. higher dimensional vectors). The results of the individual decision processes are subsequently combined using an ensemble approach to obtain an overall classification of the protein sequences.

The benefits using the proposed granular model for classifying MLVS feature vectors are twofold. First, classification accuracy may improve as a result of processing data at different levels of granularity by minimizing the curse of dimensionality phenomena. This benefit is particularly significant when analyzing high dimensional MLVS vectors. Second, the process of classifying MLVS vectors can be easily distributed across multiple computing processors and therefore provide a reduction in processing time.

7 Conclusion and Future Work

The major challenges of data analytics comes from the complexity and volume of the data. In this chapter, we propose a comprehensive granular model for decision making in order to tackle both challenges. There are different types of data complexity. One is that the structure of a data instance is complex. For this type of complexity, this chapter suggests that a complex data instance is first decomposed to multiple partial views, each of which has simpler structures. Then the proposed decision model can be applied to those partial views for decision making. Another type of data complexity is that each data instance is associated with multiple heterogeneous features with different natures, formats, and scales. For this type of data, we can first categorize those features into different partial views, based on which the proposed granular model can be applied for decision making. This chapter uses protein sequence classification and stock price movement predication to illustrate how to apply the proposed decision model for both types of complexity.

For complex data with large volume, the proposed decision model first distributes the overall decision making process onto different partial views. On each partial view, the decision making starts with coarsest granular descriptions, which typically requires much less intensive computation. For those data instances where more intensive computation is needed at a finer level of granular representation, often the number of data instances that need to be involved in intensive computation can be reduced by using the results generated at some previous coarser level as filters. We plan to conduct large-scale experimental studies on various types of complex data, including stock data and protein sequence data, to further refine the proposed decision model. Another future research work is to implement a computational framework that supports the proposed decision model on big data computing platforms such as Spark [13, 14].

References

1. Bargiela, A., Pedrycz, W. (eds.): Human-Centric Information Processing Through Granular Modeling. Springer, Berlin (2009)
2. Yao, Y.Y.: Perspectives of Granular Computing. In: Proceedings of the 2005 IEEE International Conference on Granular Computing, pp. 85–90 (2005)
3. Yao, Y.Y.: Granular Computing: Past, Present, and Future. In: Proceedings of the 2008 IEEE International Conference on Granular Computing, pp. 80–85 (2008)
4. Yao, Y.Y.: Granular Computing: Basic Issues and Possible Solutions. In: Proceedings of the 5th Joint Conference on Information Sciences, vol. 1, pp. 186–189 (2000)
5. Yao, Y.Y.: Granular Computing and Sequential Three-Way Decisions. In: Lingras, P., Wolski, M., Cornelis, C., Mitra, S., Wasilewski, P. (eds.) RSKT 2013. LNCS, vol. 8171, pp. 16–27. Springer, Heidelberg (2013)
6. Yao, Y.Y.: An Outline of a Theory of Three-Way Decisions. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 1–17. Springer, Heidelberg (2012)
7. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping to massive datasets. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 1–11. Springer, Heidelberg (1999)
8. Lee, S.J., Jeong, S.J.: Trading Strategies based on Pattern Recognition in Stock Futures Market using Dynamic Time Warping Algorithm. Journal of Convergence Information Technology 7(10), 185–196 (2012)
9. Desai, N., Antonopoulos, D., Gilbert, J., Glass, E., Meyer, F.: From genomics to metagenomics. Current Opinion in Biotechnology 23(1), 72–76 (2012)
10. Akkoç, C., Johnsten, T., Benton, R.: Multi-layered Vector Spaces for Classifying and Analyzing Biological Sequences. In: Proceedings of International Conference on Bioinformatics and Computational Biology, pp. 160–166 (2011)
11. Raghavan, V.V., Benton, R.G., Johnsten, T., Xie, Y.: Representations for Large-scale Sequence Data Mining: A Tale of Two Vector Space Models. In: Ciucci, D., Inuiguchi, M., Yao, Y., Ślęzak, D., Wang, G. (eds.) RSFDGrC 2013. LNCS, vol. 8170, pp. 15–25. Springer, Heidelberg (2013)

12. Johnsten, T., Fain, L.A., Fain, L.E., Benton, R., Butler, E., Pannell, L., Tan, M.: Exploiting Multi-Layered Vector Spaces for Signal Peptide Detection. International Journal of Data Mining and Bioinformatics (to appear)
13. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: Cluster Computing with Working Sets. In: Proceedings of the 2nd USENIC Conference on Hot Topics in Cloud Computing, pp. 10–16 (2010)
14. https://spark.apache.org