Softly Focusing On Data

Lawrence J. Mazlack
Computer Science
University of Cincinnati
Cincinnati, Ohio 45221-0030
mazlack@uc.edu

key words: data mining, reduction, focus, non-crisp, unsupervised, dissonance, clustering*

Abstract

A computational approach providing a focus for unsupervised, reactive, data mining is suggested. In data mining, achieving focus is an important issue. This is because there are too many attributes and values in a real database to consider them all. A soft focus is suggested as both data and the focus product may be imprecise. An approach is suggested for unsupervised search controlled by progressive reduction of cognitive dissonance. Both crisp and non-crisp data are subject to discovery. Soft computing tools are needed because of the need to: granulize data and to establish crisp boundaries in a non-crisp world. Issues involve: coherence measures, granulization, user intelligible results, unsupervised recognition of interesting results, and concept equivalent formation.

1. INTRODUCTION

Databases have significant amounts of stored data. Much of the data is implicitly or explicitly imprecise. The data is valuable because it was collected to explicitly support particular enterprise activities. There could well be undiscovered, valuable relationships in the data. The issue is how to best recognize them.

The volume of stored data usually directly correlates with investment; so the larger the database, the larger the potential payoff. At the same time, the larger the database, the greater is the difficulty of analysis.

Data mining seeks to discover noteworthy, unrecognized associations between data items in an existing database. This can be thought of as discovering 'interesting' patterns in the data.

Data mining supports the creation of knowledge from collected data. Instead of the modeling and implementation of existing, explicit human knowledge; data mining techniques elicit knowledge that is implicit in the databases. In a sense, we could say that the implicit knowledge is not yet available for use.

One way to try to discover new information is to try to discover the relationships between all of the items. This may work when there are relatively few things to consider. However, in a large collection of data, the problem may be too computationally complex. When contemplating databases that capture information with thousands of records, each with hundreds of attributes, a sense of the problem's magnitude becomes apparent. Furthermore, some databases have millions of records.

There are three possible approaches to the problem of complexity: (a) develop parallel algorithms, (b) reduce the quantity of records through sampling, and (c) reduce the data to be considered through *focusing*.

A number of different techniques can be used. There are two basic approaches: unsupervised discovery, or, supervised discovery. Supervised search has the advantage of providing focus. However, supervised search limits the results as it is necessary to at least *determine in advance the subjects that are of interest*. This is counterintuitive to the broadest goals of conducting discovery to find unexpected, interesting things. In keeping with the more ambitious database discovery goals, this work considers unsupervised discovery. Most unsupervised efforts try, in one way or the other, to achieve focus.

Achieving focus is important in data mining. This is because there are too many attributes and values in a production database to consider them all. Most unsupervised data mining approaches try to achieve attribute focus by first recognizing the most interesting. This work suggests the converse; i.e., eliminating the least interesting first.

A 'soft' focus (using soft computing tools) is needed because of the often inherently imprecise process. Data may be imprecise, missing, or incomplete. The resulting product or focus object may also be imprecise.

2. BACKGROUND

A database is a collection of data from which different facts can be efficiently retrieved in response to specific queries. Numerous databases have significant amounts of stored data. It has been estimated that the amount of captured data in the world doubles every 20 years. The automation of business activities produces an ever increasing stream of data. Even simple transactions, such as a telephone call, the use of a credit card, or a medical test, are typically recorded in a computer. Scientific and government databases are also rapidly growing.

* Parts of this work were performed while the author was visiting with the Berkeley Initiative in Soft Computing, Computer Science Division, EECS Department, University of California, Berkeley.

Clearly, little of the captured data will ever be seen by human eyes; let alone intelligently analyzed by a human. If it will be understood at all, it will have to be analyzed by computers. At the same time, there is a growing realization and expectation that data, intelligently analyzed and presented, *should* be a valuable resource.

2.2 Focus

Achieving focus in database mining is an important issue. This is because there are too many attributes and values in a real database to consider them all. The most common approach uses a metric based on Shannon's information theory. However, this may not always be satisfactory as the underlying motivations of Shannon's work are not necessarily the same as those of database mining. In addition, these approaches impose a sequence dependency (i.e., a taxonomy) that may not be in the data.

In general, focusing starts with some input set S that is described by some additional characteristics C_s,. These characteristics may contain information about types, ranges, noise, etc. -- as far as known in advance, usually from a database dictionary. Focus aims at retrieving some sample*s* that is smaller than *S*. Focusing may be iterative. Whenever evaluation by an evaluation criteria E_L yields insufficient results, focusing starts again. The evaluation criteria are most often quantitative; but qualitative criteria are also possible.

There are two basic focusing techniques: *forward selection* and *backward selection*. Forward selection is the most common and it attempts to recognize the most interesting features first. Backward selection in explored in this work. It seeks to achieve focus by reducing the data through discarding undesired attributes and data set partitioning.

2.2 Supervision

Searching databases to sift out information is a relatively new area of endeavor. Most efforts have some expectation of the nature of the information to be discovered. (Thus, much of the existing database mining work has used supervised discovery.) The expectation may be the items of possible interest and/or the form of the information.

However, supervised search limits the of the results as it is necessary to determine in advance the subjects that are of interest. Consequently, the heart of this work is the search for a concept to support unsupervised mining.

One problem with unsupervised search is combinatoric explosion. To consider fully the interrelationships between all the attributes is computationally prohibitive in a typical database. Possible heuristic help comes from the realization that many of the combinations do not appear to have much information value. This leads to this work's controlling heuristic focusing on increasing coherence and decreasing dissonance. The heuristic assumption is that reducing cognitive dissonance increases useful information. The speculation is that database exploration can be accomplished through a progressive reduction of cognitive dissonance.

Unsupervised mining usually tries to achieve focus by computationally identifying attributes of greatest interest. Most often, this is based on a metric drawn from Shannon's information theory. In large data sets, because of metric complexity, training sets are used. (The training sets are records sampled from the database.) Result quality is dependent both on training set quality as well as the method's underlying appropriateness.

An alternative unsupervised approach is suggested. The idea is to first eliminate the uninteresting. Informally, the approach is to form cohesive, comprehensible information groups by sifting out attributes that provide less information; and, at the same time, concentrate what is discovered. Metaphorically, the data is concentrated into information 'nuggets'. The final or intermediate results are not necessarily symmetric with methods seeking to first recognize the interesting.

2.3 Uncertainty

Lack of crispness is an issue. Most mining techniques implicitly assume that the data is clean and that the data can eventually be effectively clustered on a precise metric. For example, Fisher [1995] suggests that while it might be initially desirable to defer cluster boundaries, it eventually can be done. However, the reality is that data is often imperfect, that some values are inherently imprecise. Another concern is that the boundaries between groups of data may not be crisp.

One class of solutions considers impreciseness to be an impediment to correctness and uses a variety of techniques to reduce data variability. The other approach considers some data to be inherently non-crisp and seeks to work with the data while retaining its non-crispness.

We use soft computing tools to work with imprecise data. Of the approaches to handle non-crisp data, fuzzy methods are the most mature. There is a considerable history in using fuzzy techniques to form clusters [Zadeh, 1976] [Bezdek, 1992]. Other complementary approaches include Dempster-Shafer Theory [Shafer, 1976] and Rough Sets [Pawlak, 1991].

A particular problem of data mining is that multiple intra-item distance measures must be utilized to separate database records. Database records may have hundreds of different data attributes. Some of these attributes are scalar; some are not. Establishing a combined distance metric is difficult. A combined distance metric is needed if taxonomic structure is to be avoided.

2.4 Increasing Cohesion

Gaining focus by using a cohesion enhancement paradigm has the benefit of avoiding the intractability of combinatorial complexity that arises in attempting to discover relationships between all elements. Cohesive information is also easily understood. The most cohesive groups of data

can be thought of as information 'nuggets'. This is illustrated by Figure 2.4.

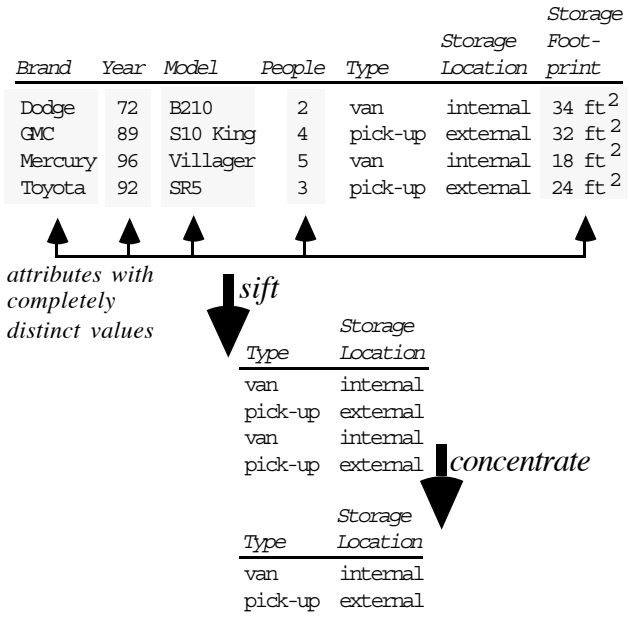

Figure 2.4 Example illustrating discovery through cohesive data concentration.

3. APPROACH'S CONCEPTUAL OVERVIEW

The underlying research question is the extent to which data mining methods must necessarily employ special domain knowledge. The speculation is that developing focus by increasing cohesion will aid in data mining.

3.1 Rationale

There are a number of different techniques that can be used in mining. Many of them use some form of supervised search. For example, in a medical database, the question could be identifying anomalous physician services.

However, supervised search limits the results because it is necessary to *determine in advance the subjects that are of interest*. This is almost counter-intuitive to the general goal of conducting data mining of finding unexpected, interesting things. The implementing program's name, *Heraclitus*, is indicative. Heraclitus (a Greek philosopher) observed "If you do not expect the unexpected, you will not find it."

On the other hand, unsupervised search has a problem with combinatorial explosion. In a typical database, there may be hundreds of attributes. To consider fully the interrelationships between all of them is prohibitive. Possible heuristic help comes from the realization that many of the combinatoric combinations do not appear to have much information value.

Often, the goal of a data mining effort is top-down classification. This investigation approaches the question from the other side; i.e., trying to identify coherent collections of information. This is done by progressively partitioning the data to reduce intra-item dissonance within the resulting partitions.

The main tool in this effort is focus developed by reducing dissonance *within* a partition. This is opposed to the more common machine learning approaches that attempt to discover metrics to best discriminate *between* individual data items as with Quinlan's [1986] entropy disorder metric. Quinlan sought to determine the attribute *sequence* that most efficiently discriminates among the database *items*. This work is not particularly interested in capturing the partitioning sequence, although it may be a process artifact. Similarly, this work is less interested in intra-item discrimination; but, more interested in recognizing commonalties.

The speculation is that discovery can be accomplished through increasing focus within a data partition by a progressive reduction of cognitive dissonance. The heuristic assumption is that reducing cognitive dissonance increases information. Another way of saying this is that finding groups of records that have a high degree of internal coherence may produce interesting results. This is done by progressively discarding attributes that have limited information value and by partitioning the data to increase cohesion within the partition.

3.2 Partitioning

An issue is whether it is better to attempt to functionally combine all items of a table into a single disorder measure; or, to use a metric of individual, distinct attributes. Using the coherence of only a single attribute may be satisfactory. Or, it might be better to consider combinations of attributes. The most straight forward, comprehensible, and computationally attainable is to use the distinctiveness of the values within one attribute.

3.2.1 Partitioning On Crisp Data

If a table of data $\boldsymbol{T}$ is made up of elements $\boldsymbol{t}_{i,j}$ where i represents the row (or tuple) of data and j represents an attribute of the database, $\boldsymbol{T}$ is partitioned by placing the rows into different partitions. The partitions are constructed so that the coherence of the resulting partitions is greater than the coherence of the initial data.

Partitions can be formed using the distinctness of attributes with crisp data attributes. $\boldsymbol{T}$ can be partitioned on the distinctiveness of attributes so that each partition only contains only a single value for a particular attribute. For example, the table in *Figure 3.2.1* is split into two sub-partitions on $\boldsymbol{t}_{i,2}$. $\boldsymbol{T}$ could have also been partitioned into two different sub-partitions on $\boldsymbol{t}_{i,6}$. However, partitioning on $\boldsymbol{t}_{i,2}$ also partitions on $\boldsymbol{t}_{i,7}$; thus, the partitioning is accomplished on two attributes as opposed to one. This suggests a partitioning heuristic of partitioning on the maximum count of attributes. (Three partitions could have been formed using $\boldsymbol{t}_{i,1}$.) The preferable partition count (i.e., more, less, some count) is a research question.

original

$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$	$t_{i,5}$	$t_{i,6}$	$t_{i,7}$
a	b	c	d	z	w	g
t	b	c	h	e	p	g
k	b	c	r	f	w	g
k	m	n	s	h	p	j
t	m	t	s	x	w	j
a	m	v	s	d	p	j

partitioned

$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$	$t_{i,5}$	$t_{i,6}$	$t_{i,7}$
a	b	c	d	z	w	g
t	b	c	h	e	p	g
k	b	c	r	f	w	g
k	m	n	s	h	p	j
t	m	t	s	x	w	j
a	m	v	s	d	p	j

Figure 3.2.1 Partitions formed on crisp attributes $t_{i,2}$ and $t_{i,7}$.

3.2.2 Partitioning On Non-Crisp Data

Partitioning may be extended to include non-crisp values by using linguistic values and by granulation. Granulation can be used to aggregate ordered data values. After granulation, partitions with granulized data can be subjected to the partitioning process. In an ordered sequence of linguistic variables, a possible partitioning heuristic is to partition in a linguistic variable ordered between two other linguistic variables.

Granulation has two advantages:

- Attributes containing granulized data can be more easily partitioned into large partitions. Partitioning may proceed as before with attributes of granulized data more easily partitioned into wholly uniformed valued partitions. This is because the increased granularity reduces attribute distinctiveness. For example, in *Figure 3.2.2a*, attribute $t_{i,*}$ cannot be simply partitioned because all of its values are distinct. However, attribute $t_{k,*}$ can be partitioned into fewer groups (2) because granulation reduced attribute distinctiveness.

... $t_{i,*}$ ...	... $t_{k,*}$ ...
1	L
2	L
8	H
9	H

Figure 3.2.2a Granulized reduction of attribute distinctiveness allowing simple partitioning

- If the granules are soft, partitioning may proceed by placing the same granule of an attribute value granule into neighboring partitions for different records. Potentially placing the same attribute value into alternative partitions may be accomplished either by fuzzy or rough sets. One possible fuzzy technique is to define membership functions that can be placed into either of two neighboring partitions. In an ordered sequence of granules, a possible partitioning heuristic using fuzzy granules is to form the partition in a granule ordered between two other granules. For example, given *Figure 3.2.2b,* the partitions may be formed as shown in *Figure 3.3.2d.*

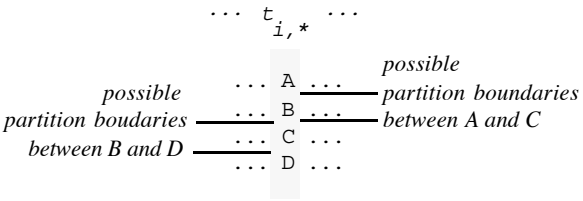

Figure 3.2.2b Possible partitions formed on a data attribute $t_{i,*}$ containing linguistic variables.

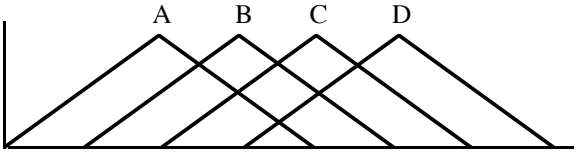

Figure 3.2.2c Overlapping membership functions supporting linguistic granules in multiple partitions.

The neighboring partitioning heuristic's use becomes evident when considering data tables. For example, in *Figure 3.2.2d,* the partition on the maximum count of attributes is on $t_{i,1\&2\&4}$. This partitions on one attribute with crisp values and two attributes with non-crisp values. If only attributes with crisp values were used, the partitioning would be on $t_{i,4\&5}$.

$t_{i,1}$	$t_{i,2}$	$t_{i,3}$	$t_{i,4}$	$t_{i,5}$	
A	D	x	r	t	
A	D	x	s	v	
B	B	x	r	t	
B	E	x	s	v	*possible paritition*
C	E	y	r	t	
C	F	y	s	v	
C	F	y	r	t	
C	F	y	s	v	
non-crisp	*non-crisp*	*crisp*	*crisp*	*crisp*	

Figure 3.2.2d Partitions formed on both crisp and non-crisp data attributes. Partitioning on $t_{i,1}$, $t_{i,2}$, $t_{i,3}$

Forming partitions on multiple attributes in a large database is not computationally simple. When adding granulized semantic values to the mix, the problem becomes computationally more complex. Help may be sought from the operations research discipline. Computationally sifting through the derived information for the most interesting re-

sults is speculative. Various heuristics will be examined to recognize computationally the most interesting.

3.2.3 Granule Formation (Size, Count)

Granule size is an important issue, both for human cognitive understanding and for unsupervised partitioning that is to take advantage of fuzzy granules. The finer the grains, the greater is the partitioning difficulty. Alternatively, if the grains are too large, useful information may be lost. Zadeh has informally observed (in 1996 seminars) that experience indicates that the appropriate count of membership functions is generally between 5 and 7. While experience may eventually dictate otherwise, without other information, this would appear to be a good starting point.

It is an open question whether granules for data mining should be the same size (i.e., either scalar dimensionality or member count). It may be more convenient to arbitrarily divide a domain into equal scalar segments; but, it may not be the most satisfactory. For example, if ages are recorded in tenths; i.e., 0.0, 0.1, 0.2, … ; a reasonable, useful human granulization might be [infant, adolescent, adult, senior] or [infant, adolescent, young-adult, adult, senior, aged]. In both cases, the granules overlap and the range of values is differentially sized. This case is different from the upward hierarchy granule recognition from data already implicitly placed into linear groups. For example, granulizing student Grade Point Averages (GPAs) of [0.0, 0.1, … , 3.9, 4.0] into A Œ [3.0, … , 4.0], B Œ [2.0, … , 3.0), etc.

Difficulties with unsupervised recognition of granules in ordered data lie with (a) the number of granules to be used and (b) the range of values to be included in a particular granule. There are known ways of clustering data when the number of clusters is known [Bezdek, 1981] and initial cluster seeds are known. However, in unsupervised database mining, this information is usually not available.

Unsupervised clustering is difficult. One possible approach is to use a variation of the 'mountain method' to estimate the cluster centers [Yager, 1993] [Chiu, 1994]. Then, use these results as the starting point to another method (such as fuzzy c-means clustering) to refine further the results. The 'mountain method' procedure uses a metric to identify where there is the greatest density of values. This is the initial cluster center. Then, reduce the density metric for other points in the neighborhood and find the next point with the highest (adjusted) density and consider this to be the next cluster center. The process is recursively repeated.

3.3 Algorithm

Data drawn from a data warehouse is mined. No computationally usable semantic information will be captured. Attributes are described as to label, type, range, enumeration, and other basic typing information typically available from a DBMS data dictionary.

Mining is done by reductive clustering based on reducing dissonance. The approach is essentially a method of categorization by most common members. A high level description is:

- Place a normalized database into a table data structure that can easily be manipulated. Once dissonance reduction begins, the results need not be kept normalized.
- Recursively increase focus by partitioning the database to reduce dissonance internal to the resulting partitions.
- Aggregate data through granulization.
- Compress duplicate rows.

4. SUMMARY

Potentially the most useful data mining products may come from unsupervised data mining. To be successful, achieving focus is important. This is because there are too many attributes and values in a real database to consider them all. A concern is that the data as well as the products are often inherently imprecise. For this reason, a soft focus developed by soft computing tools may be useful.

This work suggests an approach using bottom-up, reductive clustering. To reduce complexity, data with less information value is progressively eliminated. Approximate reasoning techniques to address inherently non-crisp data, granules, and product objects are considered.

REFERENCES

J.C. Bezdek (1981) **Pattern Recognition With Fuzzy Objective Functions**, Plenum Press, New York

J.C. Bezdek, S.K. Pal, eds. (1992) **Fuzzy Models For Pattern Recognition**, IEEE Press, New York

S.L. Chiu (1994) *A Cluster Estimation Method With Extension To Fuzzy Model Identification,* Proceedings of Third IEEE International Conference On Fuzzy Systems, Orlando, p 1240-1245

D. Fisher (1995) *Iterative Optimization and Simplification Of Hierarchical Clustering,* Technical Report CS-95-01, Vanderbilt University

Z. Pawlak (1991) **Rough Sets: Theoretical Aspects Of Reasoning About Data**, Kluwer Academic Publishers, Boston

G. Shafer (1976) **A Mathematical Theory Of Evidence**, Princeton University Press, Princeton, N.J.

R.R. Yager (1993) *Learning Of Fuzzy Rules By Mountain Clustering*, Proceedings Of The SPIE, v 2061, p 246-254

L.A. Zadeh (1965) *Fuzzy Sets*, Information Control, v 8, p 338-353

L.A. Zadeh (1976) *A Fuzzy-Algorithmic Approach To The Definition Of Complex Or Imprecise Concepts,* International J Man-Machine Studies, v 8, p 249-291