MINING FUZZY SPATIAL ASSOCIATION RULES FROM IMAGE DATA

By

George Brannon Smith

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Science
in the Department of Computer of Science

Mississippi State, Mississippi

August 2001

MINING FUZZY SPATIAL ASSOCIATION RULES FROM IMAGE DATA

By

George Brannon Smith

Approved:

Susan M. Bridges Associate Professor of Computer Science (Major Professor)	Julia Hodges Professor of Computer Science (Committee Member)
Hasan Jamil Assistant Professor of Computer Science (Committee Member)	R. Rainey Little Associate Professor of Computer Science (Graduate Coordinator)

A. Wayne Bennett
Dean of the College of Engineering

Name: George Brannon Smith

Date of Degree: August 4, 2001

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Dr. Susan Bridges

Title of Study: MINING FUZZY SPATIAL ASSOCIATION RULES FROM IMAGE DATA

Pages in Study: 100

Candidate for Degree of Master of Science

The work of Agrawal et al. on the type of data mining known as association rule mining has been the basis for continuous research over the past seven years. Kuok, Fu and Wong have extended association rules with the fuzzy set theory of Zadeh to build a system more adapted to real world data. Meanwhile Bloch has recently applied this same fuzzy set theory to the area of image analysis, particularly to the task of finding relationships between image objects. Koperski and Han have been the leaders in adapting general data mining techniques, including association rules, to the domain of spatial data. This thesis describes a synthesis of these techniques to form a unified system for generalized image analysis, specifically finding general fuzzy rules about the relationships of objects in image data sets. Experimental results on synthetic data and real world data samples demonstrate that such a system is effective in producing interesting rules.

DEDICATION

To Mom & Dad - for their infinite patience

ACKNOWLEDGMENTS

The author greatly appreciates the tireless efforts of Dr. Susan Bridges, committee chairperson, for her role as editor and reviewer of this thesis. Thanks go to Sean Taylor as well for giving his valuable time to conduct code review in the experimental phases of this work. Special thanks go to Bruce Wooley, Ph.D candidate, for creating *mpiShell*, a powerful parallel computing interface, and providing documentation and invaluable live technical support. His work greatly facilitated my experiments. Finally, thanks go to the NAVO Research group based at Stennis Space Center in Bay St. Louis, MS for their support - in particular the use of sample data. This support was made possible through the following grants:

- National Science Foundation Grant #9818489
- ONR EPSCoR Grant N00014-96-1-1276
- Naval Oceanographic Office via NASA Stennis NAS1398033 DO92

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

With the rapid advancement in computing power, storage capacity, and collection techniques, more and more spatial image data are being collected. This data explosion is compounded by the fact that because the data sources, namely the earth, sea and atmosphere, are dynamic, data often have to be recollected to be of real use, creating another dimension to the data. But the raw data itself is only the first step in understanding. Analysis is required to find useful generalized information, but considering the volume of the data, human expert analysis becomes a tedious and tiresome job. This problem has given rise to many algorithms and systems seeking to extract useful generalized observations of various types. Such analysis commonly goes by the broad term *Knowledge Discovery in Databases* (KDD) or sometimes simply *Knowledge Discovery* - the finding of implicit and previously unknown or unseen information in a database by the application of computer algorithms. In fact, Fayyad et al. declare that "Knowledge Discovery is the most desirable end-product of computing" (Fayyad et al. 1996). Unfortunately, they also concede that it is " ... one of the most difficult computing challenges to do well"(Fayyad et al. 1996). Doing it well is a pursuit in which many researchers are engaged and they have developed several techniques towards that end. A useful system could be constructed by combining some of these existing techniques into a multi-layer system.

1.1 Motivation

As mentioned before, much data is collected from spatial sources such as remote sensing of the oceans and terrain. We are often interested in the nature of the many components of such images, objects and regions (e.g., a stand of trees, coral reefs, etc.) and may apply KDD techniques such as classification to extract these. But now that we know the classes of the various entities in an image or more likely a large set of images, what next? These components do not exist by themselves; they exist in an image (and on the earth in the case of real world remote sensing) along with many other objects. We would like to know the relationships among the objects. For example, "Sandy ocean bottom #235 occurs 2 miles to the right of coral reef #769" in an image (which may in turn mean east or down current in the real world). Or "Grassy meadow area #114 is located 5 miles above deciduous forest #80." However, considering the volume of data, even individual spatial relations may not be of interest to us. Rather, generalized information on aggregates of the data is more immediately useful. How does one class of region relate to other classes? An answer, courtesy of a specific KDD system, might be "Lakes (a class) tend to occur next to mountains (another class)" or "Sandy bottoms (a class) tend to occur close to and below coral formations (another class)." A more recognizably useful scenario might be:

- Discovered "Rocky bottoms usually occur to the west (having corrected for real world orientation) of bottom type #12".
- Bottom type #12 has often proven a good place to drill for oil.
- Remote sensing data shows no bottom type #12 - only a rocky bottom.
- However, knowing the general relationship of the two classes, let's focus on the area west of the rocky bottom and see what turns up.

A real world example is the joint project between the Naval Oceanographic Office (NAVO) and the Mississippi State University Computer Science Department, in which undersea side-scan SONAR data is being analyzed by a research group (Bridges et al. 1998). The image data is being processed by region growing techniques and the Bayesian classifier Autoclass, which partitions the sea-floor data into discrete regions of relatively few classes. Doing this effectively and efficiently is the current focus of the research. Generalizing these results in their spatial context is a next step. A rule induction system could be used to take the classified image set and discover the generalized spatial nature of the component parts, thereby providing more human usable information about the data.

1.2 Background

In 1993, Agrawal, Imielinski and Swami presented their theory of *association rule mining* for finding generalizations in a database, with their initial test domain being retail store transaction data. The application is not limited to this domain, but can be applied to general boolean attribute data (e.g., item is/is not purchased, person is/is not married, system is/is not operational). Indeed subsequent research has extended the technique to categorical attributes (e.g., car make in {Ford, GM, Toyota, Daimler, etc.}) and quantitative attributes (e.g., client Age in [18-24], [25-30], [30-35], etc.). Kuok, Fu and Wong have even extended that with fuzzy set theory to extract fuzzy association rules from quantitative data (e.g., if client Age in 'Young Adult' to some degree then income is 'Low' to a certain degree for some percentage of the database).

Among the researchers exploring association rules are Koperski and Han (1995) who have been applying a variety of other data mining techniques to spatial data,

data pertaining to the position of objects in a space. Koperski and Han attempt to make generalizations along the lines of: 65% of houses are west_of a lake. While they do acknowledge the mining of image or raster data, they typically work on data actually in a spatial database system in which the data has already been broken down and analyzed to a certain degree for its initial insertion into such a system.

Along those lines, Ordonez and Omiecinski (1999) have done a preliminary investigation into rule mining from true image data, using a Blobworld image analysis backend to extract objects from raster images. They then transform these blob data into a meta-database analogous to that of Agrawal et al. That is to say, an important part of their technique is the preparation of the image data to create something like a spatial database mentioned above. Standard mining techniques are applied to the meta-data to mine association rules describing the relationship between objects in the image base, namely co-occurence, e.g., images with a square tend, depending on the rule support and confidence, to also contain triangles.

Also somewhat similar is the work of Bloch, who has developed techniques for finding spatial relationships between objects, mainly directional attributes. The chief differences are:

1. She, like Ordonez and Omiecinski, is working in imagespace (such as MRI scans) rather than in object space.
2. She is using fuzzy set theory to take into account the varying morphology (i.e., size and shape) of the objects in question, which can have an impact on how best to describe the spatial relationship.
3. Her work is on the specific object relations, rather than mining generalized rules.

The result of Bloch's techniques is not a single precise number but a range which actually captures the imprecision of the relationship.

Fuzzy set theory is a generalization of classical set theory that was first proposed by Zadeh in 1965. Whereas an element of a classical or *crisp* set is either totally in a given set or not in it at all, fuzzy set theory allows for elements to be in a set to a degree ranging from total inclusion to total exclusion. The former works fine for domains where set membership is indeed a binary relationship such as belonging to a sports team or being a US state. The latter is useful for situations such as the set of all TALL people or OLD people, sets which have no sharp boundaries unless they are arbitrarily imposed. The applications above exploit this feature for their own designs. An object A is to the RIGHT of object B to some degree in a Bloch image. A person is in the 'Young Adult' set to some degree: 1.0 for a 22 year old; 0.3 for a 16 year old.

1.3 Hypothesis and Main Goals

The hypothesis of this work is that a synthesis of the fuzzy spatial relation techniques of Bloch (1999) and the fuzzy association mining methodologies of Kuok, Fu and Wong (1998) and others, enables the extraction of general fuzzy spatial rules in an image or set of images. We restrict our attention to raster data and operate in image space as opposed to the object space used in true spatial databases (Koperski, Adhikary and Han 1996). Our goal is to use these combined techniques to extract general rules describing exclusively directional relationships of image components, i.e., pixel-based objects, from segmented, classified raster data. The integration of Bloch's approach (1999) allows the implicit consideration of spatial object morphology. When used in conjunction with techniques for fuzzy association rule mining, information can be mined that describes the certainty of the extracted rules in terms of a fuzzy membership interval. This investigation focuses on two

basic fuzzy association forms describing directional relationships in two dimensions. Additional spatial data attributes, namely inter-object distance and object size (relative or absolute), are not explicitly taken into account, but are reserved for future work. The extension of the basic rule forms describing directional relationships such as "south of" to encompass higher level descriptions of relationships such as "surrounded by" is also reserved for future work.

1.4 Organization

The remaining chapters are organized as follows:

- Literature Review: A survey of the source work used for this research.
- Theory: the proposed formalism and methods upon which this research is based including
  - Relations: the nature of extracted object relations
  - Rule forms: what our spatial rules will look like and how they are achieved from the relations
  - Support and confidence: the metrics used to gauge the importance and usefulness of the rules.
- Experimental Results: details on the testbed implementing the theory, the complexity incurred, the data used to test, and results from that data; also included is a description of the application of parallelism to the process.
- Conclusion: Analysis of the results; a large part of this chapter will discuss avenues for future work - additional features, more advanced rule forms, possible improvements, etc.
- Appendices are included as well, offering some sample visualizations and tablular result data.

CHAPTER II

LITERATURE REVIEW

2.1 Fuzzy Logic/Fuzzy Sets

In 1965, Lotfi Zadeh presented his controversial theory of fuzzy sets (Zadeh 1965), and he is generally recognized as the founder of the discipline, though Black (1937) did discuss the related concept of vagueness as far back as 1937, and some have pointed out that there were considerations on the issue of a region between true and false as far back as Plato (Brule 1985). What Zadeh proposed was a generalization of classical set theory to handle the inexactness, approximation and what he calls *elasticity* of real life, such as the concepts *tallness* or *nearness*.

We will assume the reader has an understanding of the basics of classical set theory, its properties, and operations such as union and intersection, and so will not review those fundamentals here except to say a word about the notion of a *characteristic function*. One of the common ways of describing a set is by giving its characteristic function (Schmucker 1984), and in fact a set and its characteristic function are often used interchangeably. A characteristic function is one that maps the elements of the universe of discourse onto the two element set $\{0, 1\}$, sometimes called a *valuation set* (Dubois and Prade 1980). This means that given a universe X and a subset A:

$$char_A(x) = \begin{cases} 1 & \text{iff } x \in A \\ 0 & \text{iff } x \notin A \end{cases}$$

For example, given a universe $X = \{a, b, c, d, e, f, g\}$, a characteristic function could yield a subset A:

$$A = \{a/1, b/0, c/0, d/1, e/0, f/1, g/0\}$$

as a result. Customarily, false members are not shown, nor are the characteristic values (there could be only one) resulting in the more familiar roster form

$$A = \{a, d, f\}$$

(which we can do for a finite subset of a finite universe). This works fine for domains in which set membership is truly binary such as the set of all students currently enrolled at Mississippi State University, or the set of all Nobel Laureates. It does not work for less precise everyday concepts such as the set of all TALL men in the room or the set of all stores CLOSE to a residence.

To handle these imprecise scenarios, Zadeh proposes defining set membership not on the finite two-valued set $\{0, 1\}$ but rather over the real interval $[0, 1]$. Set elements can still take the old characteristic values or *membership values* of 0 and 1 - but they can also now take on say 0.7 or 0.34 indicating partial set membership or a degree of membership. This fuzzy set can be described by a different kind of characteristic function, a *membership function* denoted by $\mu_A(x)$ and taking values in $[0, 1]$.

For example, given the same universe as above $X = \{a, b, c, d, e, f, g\}$, a membership function could yield a fuzzy subset A:

$$A = \{a/0.9, b/0.23, c/0, d/0.7, e/0.4, f/1, g/0\}$$

as a result. Again, completely false members (i.e., with a 0 degree of membership) are not shown. However, other fuzzy membership values are retained. A roster form in this case would be

$$A = \{a/0.9, b/0.23, d/0.7, e/0.4, f/1\}$$

This concept of partial membership allows us to represent real world concepts more in line with the everyday manner in which we consider them. Returning to the example of the set of TALL people, classifying 7'1" Frank as TALL ($\mu_{TALL}(Frank) = 1$) and 4'10" Cathy as not TALL ($\mu_{TALL}(Cathy) = 0$) was not a problem before with classical sets. But what about 5'11" David? The typical natural language response is probably along the lines of "sort of" or "somewhat" and assigning classical set membership can be difficult and unsatisfactory. However, $\mu_{TALL}(David) = 0.7$ better captures the imprecision of the real world description. Along these same lines, the fuzzy membership function eliminates the crisp boundary between classical set inclusion and exclusion which is often capricious and arbitrary. For example the classical set RICH may be designated by a characteristic function by which all people with income at or above $80,000 are members. Someone with income of $79,999 is not RICH while someone making $80,000 is RICH. So the difference between RICH and not RICH is now a mere dollar which is unrealistic. A gradual fuzzy membership function remedies this: Warren Buffett is still 100% RICH and someone barely scraping by is still 0% RICH but the path from the one to the other is smoother, with the $79K employee being in RICH to some degree.

Exactly what the fuzzy membership function looks like is not carved in stone but is chosen by the user based on his needs and domain knowledge. There are however several common standard membership functions including trapezoidal, triangular,

Gaussian, bell curves, and sigmoid (Yen and Langari 1998). Some examples are (Yen and Langari 1998):

$$triangle(x : a, b, c) = \begin{cases} 0 & \text{iff } x < a \\ (x-a)/(b-a) & \text{iff } a \leq x \leq b \\ (c-x)/(c-b) & \text{iff } b \leq x \leq c \\ 0 & \text{iff } x > c \end{cases}$$

$$gauss(x : m, \sigma) = exp\left(-\frac{(x-m)^2}{\sigma^2}\right).$$

Again, which is used depends on the domain, and the values of the a, b, c, m, σ may depend on domain specific knowledge gathered by the user or through some other means, or whatever other information he/she wants.

Since fuzzy set theory is a generalization of classical set theory, it stands to reason that there are operations analogous to those on classical sets, including, but not limited to, complement/negation, intersection/disjunction, union/conjunction, and cardinality.

The complement of a fuzzy set is simply 1 - membership. That is:

$$\mu_{\neg A}(x) = 1 - \mu_A(x).$$

This makes sense; if an item is in set Q to degree 0.75, it stands to reason that it is NOT in set Q to degree 0.25.

Things are more difficult with union and intersection. Fuzzy set theory does not have an intersection/disjunction operation per se but rather a class of operations called triangular norms or *t-norms*. A t-norm is a two-valued function that maps pairs $[0,1] \times [0,1]$ onto a single value in $[0,1]$. A t-norm is formally defined as satisfying a certain set of axioms.

A t-norm, denoted $t(a,b)$, satisfies the following axioms for any $a, b, c, d \in [0,1]$, (where a, b, c, d could be interpreted as fuzzy membership values $\mu_A(x)$, $\mu_B(x)$, $\mu_C(x)$, and $\mu_D(x)$ respectively, where $x \in X$, the universe):

1. $t(0,0) = 0$; $t(a,1) = t(1,a) = a$ (function boundaries)
2. $t(a,b) \leq t(c,d)$ whenever $a \leq c$ and $b \leq d$ (monotonicity)
3. $t(a,b) = t(b,a)$ (commutativity)
4. $t(a,t(b,c)) = t(t(a,b),c)$ (associativity)

Also, the the inequality

$$t_d(a,b) \leq t(a,b) \leq min(a,b)$$

where t_d is the *drastic product* or

$$t_d(a,b) = \begin{cases} \min(a,b) & \text{if } max(a,b) = 1 \\ 0 & a,b < 1 \end{cases}$$

must be satisfied (Yen and Langari 1998).

In a similar fashion, fuzzy union/conjunction is implemented by a class of operations called t-conorms. For the same conditions above, a t-conorm, denoted $s(a,b)$, satisfies the following:

1. $s(1,1) = 1$; $s(a,0) = s(0,a) = a$ (function boundaries)
2. $s(a,b) \leq s(c,d)$ whenever $a \leq c$ and $b \leq d$ (monotonicity)

3. $s(a,b) = s(b,a)$ (commutativity)
4. $s(a,t(b,c)) = s(s(a,b),c)$ (associativity)

Also,the inequality

$$max(a,b) \leq s(a,b) \leq s_d(a,b)$$

where s_d is the *drastic sum* or

$$s_d(a,b) = \begin{cases} \max(a,b) & \text{if } min(a,b) = 0 \\ 1 & a,b > 0 \end{cases}$$

must be satisfied.(Yen and Langari 1998) Furthermore, these operations are pairwise related:

$$t_m(a,b) = 1 - s_m(1-a, 1-b)$$

and vice versa.

Given these operator classes, one can then say:

$$\mu_{A \cap B}(x) = t_m(\mu_A(x), \mu_B(x))$$

where t_m is some t-norm.

Since, these are classes of operations, we can choose the ones that best suit our domain, provided they adhere to the axioms and are implemented in dual pairs. For example, we may choose to use the Einstein product t-norm out of the huge number of choices (Zimmerman 1996):

$$t_{Einstein}(a,b) = \frac{a \cdot b}{2 - [a + b - a \cdot b]}.$$

However, when the time comes to apply the fuzzy set union operation, the corresponding Einstein sum t-conorm should be used for consistency.

That said, Bellman and Giertz (1973) have made a case for the *standard* fuzzy set union and intersection operations first used by Zadeh, as the best and most natural extensions of classical set theory:

$$\begin{aligned} \mu_{A\cap B}(x) = &\quad t_{std}(\mu_A(x), \mu_B(x)) = \quad min(\mu_A(x), \mu_B(x)), \text{ and} \\ \mu_{A\cup B}(x) = &\quad s_{std}(\mu_A(x), \mu_B(x)) = \quad max(\mu_A(x), \mu_B(x)). \end{aligned}$$

Another often useful operation or property of classical sets is *cardinality* which is basically a count of the number of elements in a set, assuming the set in question is finite. However, since elements of fuzzy sets are often not completely in the set, it seems incorrect to count each member regardless of degree of membership. Fortunately the answer is quite simply the sum of the fuzzy memberships (Dubois and Prade 1980):

$$|A| = \sum_{x\in X} \mu_A(x).$$

This formula also works correctly on a classical set.

As useful as these analogous operations are, it is important to point out what is noticeably absent in fuzzy sets. Returning once again to the earlier example, $X = \{a, b, c, d, e, f, g\}$, with a fuzzy subset

$$A = \{a/0.9, b/0.23, c/0, d/0.7, e/0.4, f/1, g/0\}$$

using the aforementioned complement operation yields:

$$\neg A = \{a/0.1, b/0.77, c/1, d/0.3, e/0.6, f/0, g/1\}.$$

Applying the standard fuzzy union and intersection operations results in:

$$\begin{aligned} A \cup \neg A &= \{a/0.9, b/0.77, c/1, d/0.7, e/0.6, f/1, g/1\}, \text{ and} \\ A \cap \neg A &= \{a/0.1, b/0.23, c/0, d/0.3, e/0.4, f/0, g/0\}. \end{aligned}$$

Recall from classical set theory that given a boolean set B in a universe X:

$$\begin{aligned} B \cup \neg B &= X, \text{ and} \\ B \cap \neg B &= \emptyset \end{aligned}$$

These are known as the the *Law of Excluded Middle* and the *Law of Contradiction* (Yen and Langari 1998) respectively, and they are conspicuously absent from generalized fuzzy theory. For some, this represents a failing of fuzzy logic, but for others it is thoroughly in line with expectations. The very nature of fuzzy set membership is that an element can be both in a set and in its negation *simultaneously*.

2.2 Fuzzy Relative Positioning

Dr. Isabelle Bloch of the Ecole Nationale Supérieure des Télécommunications in Paris has been applying the theory of fuzzy sets to the problem of analysis of spatial relations of objects in images (Bloch 1999). We often want to describe the relationships between objects with imprecise natural language terms like "above" or "to the right of." This is made difficult by the facts that

- in real life objects are simply not always *exactly* "to the left of" other objects but can be offset.
- the shape and size, the morphology, of objects has an impact on their spatial relationship.

If a man is standing to the exact left of a woman and then shuffles forward a few inches, he does not completely cease to be to her left - just to a lesser degree. A man standing in the angle of a large L -shaped building may be both to the right and below it (assuming a bird's-eye view). As he walks away, he may continue to be to the right and be below for several yards.

Bloch offers a new methodology for dealing with this phenomenon in such domains as medical imaging, in both 3D and 2D space. Bloch actually utilizes fuzzy theory in two different ways. The main one is the directional relationship between objects in an image space as mentioned above. In other words, object A can be to the right of object B to some fuzzy degree. But objects themselves can also be defined as fuzzy sets to account for possible spatial imprecision, defined on the universe $\mathcal{S}$, the image itself. The elements of such a set are simply the points or pixels in the object. Being a fuzzy set, an object A can be represented by fuzzy membership function $\mu_A(x), x \in \mathcal{S}$ on the interval $[0, 1]$. It should be noted that since crisp sets are specializations of fuzzy sets, this definition can handle crisp objects taking only the values 0 and 1 as well.

Since Bloch's method is designed to work in 3D space, a direction is defined by two angles $\alpha_1 \in [0, 2\pi]$, a heading, and $\alpha_2 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, an elevation where $\alpha = (\alpha_1, \alpha_2)$. This then yields the unit vector

$$\vec{u}_{\alpha_1,\alpha_2} = (\cos\alpha_2 \cos\alpha_1, \cos\alpha_2 \sin\alpha_1, \sin\alpha_2)^t \tag{2.1}$$

for evaluation. If we choose to work only in 2D space, then $\alpha_2 = 0$ or we could simply let $\alpha = \alpha_1$.

Given this α, we want to find the degree to which an object A, defined by $\mu_A(x)$, is direction α of a reference object R ($\mu_R(x)$), which is simply another object in the image universe. This is done by first constructing what Bloch refers to as a *landscape* which is the area

> ... around the reference object R ... such that the membership value of each point corresponds to the degree of satisfaction of the spatial relation under examination(Bloch 1999).

This landscape is itself a fuzzy set, denoted by $\mu_\alpha(R)(x)$, where "in the direction α" is the spatial relation in question. A visualization of such a landscape is shown in Figure 2.2, generated from the sample image in Figure 2.1. Each point in the image space, excluding the points of the square object itself, takes a grey value from black to white indicating the degree of its membership in the fuzzy relation "in the direction $\alpha = 0$ of the square reference object, R". White indicates full membership, black indicates null membership, and the greys are various degrees in between. The object itself has been explicitly colored black.

Theoretically this landscape potentially extends over the entire universe, but practically speaking a full roster generation is neither computationally desirable nor necessary. We are only interested in the overlap between the landscape of R and object A, specifically a function of $\mu_\alpha(R)(x)$ and $\mu_A(x), x \in \mathcal{S}$, which will indicate the relationship between the points in A and the points in R.

The final result is not actually a single fuzzy value, but rather three values that represent an interval and a likely estimate. The values of the overlap function

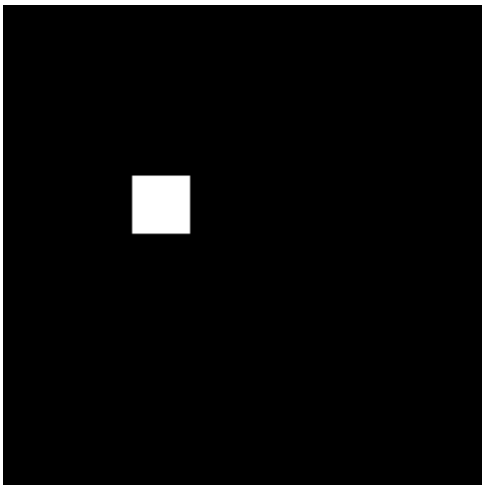

Figure 2.1: A shape object in space

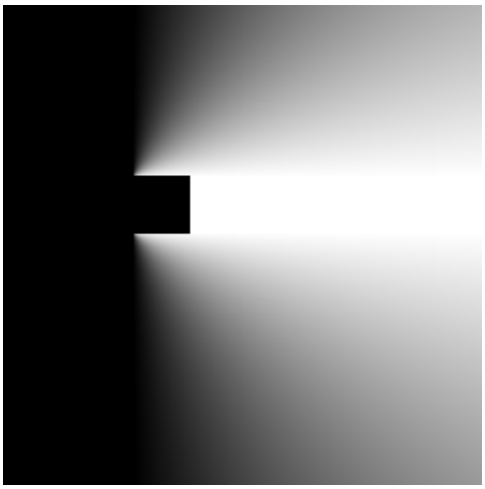

Figure 2.2: Landscape for Figure 2.1 at $\alpha = 0$

iterated over the points in the objects are evaluated in the following forms:

Possibility

$$\prod_{\alpha_1,\alpha_2}^{R}(A) = \sup_{x\in\mathcal{S}} t[\mu_\alpha(R)(x), \mu_A(x)] \tag{2.2}$$

Necessity

$$N^R_{\alpha_1,\alpha_2}(A) = \inf_{x\in\mathcal{S}} s[\mu_\alpha(R)(x), 1-\mu_A(x)] \tag{2.3}$$

Mean

$$M^R_{\alpha_1,\alpha_2}(A) = \frac{1}{|A|}\sum_{x\in\mathcal{S}} \mu_A(x)\mu_\alpha(R)(x) \tag{2.4}$$

where $t[]$ is a fuzzy t-norm, $s[]$ is a fuzzy t-conorm, supremum is least upper bound, and infinum is greatest lower bound. The *possibility* is an optimistic result - a kind of max; *necessity* is a pessimistic result - a kind of min; *mean* is the average membership degree over all the points. Together they express the value and interval $M \in [N, \Pi]$. In Bloch's opinion this can further represent the imprecision of the spatial relationship.

As stated in the section on fuzzy sets, the membership function for a fuzzy set is chosen by the user based on the various requirements of the domain. For the landscape function $\mu_\alpha(R)$ Bloch has chosen a basic linear function

$$\mu_\alpha(R)(P) = \max\left(0, 1 - \frac{2\beta_{min}(P)}{\pi}\right), P \in \mathcal{S} \tag{2.5}$$

which turns the angle β into a fuzzy value in $[0, 1]$ (Again, while $P \in \mathcal{S}$, computationally this is typically only calculated when $P \in A$, A being the other object in question). $\beta_{min}(P)$ is the minimum of all the angles $\beta(P, Q)$ where $Q \in R$, and

$$\beta(P, Q) = \arccos\left[\frac{\overrightarrow{QP} \cdot \vec{u}_{\alpha_1,\alpha_2}}{\|\overrightarrow{QP}\|}\right] \tag{2.6}$$

where $\overrightarrow{QP}$ is the vector from the point in R to the point in A. So we are essentially iterating over the points in A and finding the best angle to a point in R. This angle is compared to the specified angle α.

2.3 Association Rules

The idea of *association rule mining* as a useful KDD technique was first put forth in 1993 by Agrawal, Imielinski and Swami (1993) based on experiments with retail store transaction database testbeds.

In short, an association rule takes the form

$$antecedent \Rightarrow consequent(c\%\ confidence, s\%\ support)$$

where the antecedent consists of one or more items in the set of transactions being mined and the consequent consists of only one item not in the antecedent.

The *support* is the percentage of all transactions in the database containing both the antecedent and the consequent, and it measures the significance of the rule in the database. The *confidence* is the percentage of those transactions containing the antecedent which also contain the consequent and measures the strength of the rule. Together these two are the principal *constraints* on the association rule mining process.

A classic example of an association rule is:

$$beer \Rightarrow chips(87\%\ confidence, 3\%\ support)$$

Here 3% of all transactions (i.e., possibly a basket of goods purchased by a customer on a store visit) in the database contain both beer and chips. It should be noted that the opposite does not necessarily follow; transactions containing chips may not include beer as well. Store management may be able to use rules such as these to make decisions about store operations: where to place displays, when to have sales and on what products, etc.

Following the notation of Agrawal, Imielinski and Swami, we formally define association rules as follows:

$\mathcal{I}$ is the set of all items, aka binary attributes, in the database; $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$
T is the transaction database
t is a transaction; $t \subseteq \mathcal{I}$
X is a set of some of the items; $X \subseteq \mathcal{I}$
t satisfies X if, for each item in X, t contains that item; $X \subseteq t$
An association rule is $X \Rightarrow I_j$ *s.t.* $X \subseteq \mathcal{I} \wedge I_j \in \mathcal{I} \wedge I_j \notin X$
$X \Rightarrow I_j$ is satisfied in T with confidence $0 \leq c \leq 1$ iff at least c% of transactions in T which satisfy X also satisfy I_j.
The result is sometimes written $X \Rightarrow I_j \mid c$

Items are sometimes called binary attributes because they can be viewed from a boolean point of view. Either I_n is in a transaction or it is not. Both transactions and itemsets are simply sets of items in the universe $\mathcal{I}$.

Additional constraints can be placed on antecedent and consequent to generate more immediately useful rules such as

"Antecedent *must* contain I_n" or

"Consequent *must* be an element of a given set Q".

The application of user-defined constraints other than support and confidence is a subject for study in itself (Ng et al. 1998; Silberschatz and Tuzhilin 1996; Smith 1999).

Support is an important constraint on the resultant rule, measuring statistical significance and often business significance (given that these initial experiments are retail oriented). Low support may indicate low preferences or rules that are not worth further examination.

Agrawal et al. (1993) decompose the Association Rule Mining problem into two main subproblems:

1. Finding *large*itemsets (aka *frequent* itemsets). These are itemsets that exceed the minimum support (*minsupp*) threshold supplied by the user. These large itemsets will be used to generate the rules. They may be further pruned by additional constraints. Not surprisingly, itemsets that do not meet the threshold are *small* or *infrequent* itemsets.

2. Generating all the rules for each previously found large itemset. Formally:

   large itemset $Y = \{I_1, I_2 \ldots, I_k\}, k \geq 2$ will generate rules $X \Rightarrow I_j$ such that $X \subseteq Y$, $\mid X \mid = k - 1$ and $I_j = Y - X$, i.e., $X \Rightarrow Y - X$. Put another way, Y is the union of the antecedent and consequent of every rule derived from it.

   To enforce the user specified minimum confidence (*minconf*) threshold c:

   if $support(Y)/support(X) > c$ then the rule satisfies c.

   Obviously we must have the support of X available for this. NOTE: As for minsupp, we know that Y is large from (1), and we know that X is large because *all subsets of a large itemset are also large.*

2 Agrawal, Imielinski and Swami (1993) mainly focus on the first subproblem and offer a custom algorithm (later know as AIS), bearing in mind that

- measuring all possible itemsets on one pass could result in checking 2^m combos in a database where the number of items, m, could easily reach 1000; and
- measuring only size k itemsets on k^{th} pass, building up candidates on each could result in many passes ($k = 1000$).

In a subsequent paper, Agrawal and Srikant (1994) offer some refinements, particularly:

- A relaxation of the requirement that rule consequents be a single item - sets of items are now allowed.
- A new algorithm, Apriori, and its variants, which are significantly faster than the previous AIS algorithm.

Apriori and its brethren tackle the first subproblem previously laid out - that of finding all large itemsets from which actual rules would then be derived. It attempts to relieve problems of the AIS algorithm in which too many candidates that turn out

small are generated. Simply put, Apriori takes the set of large $k-1$ itemsets (i.e., knowledge it has *a priori*), joins them to create a new set of k-itemsets, and prunes the ones whose subsets are small. This follows from the logic that since any subset of a large set must be large then:

a) large $k-1$ itemsets can alone be used to to generate candidate large k-itemsets.

b) Any k-itemset with small $k-1$ subsets *cannot* be large.

Apriori is significant because it is efficient, relatively simple, and it scales well and has thus been used by much subsequent research as either a baseline for comparison of new techniques or as the core algorithm for the development of new techniques and domain specific modifications.

These initial forays into the realm of association rules dealt primarily with binary attributes making the results *Boolean Association Rules*. Either a transaction contains "soda" or it does not - true or false. But other kinds of data exist, specifically quantitative such as price, which have values over a range, and categorical such as brand (e.g., Dell, IBM, Compaq, etc.). Srikant and Agrawal (1996) want to mine *Quantitative Association Rules* in order to gain information about this kind of data and would like to do it with the rule infrastructure already in place. A simple approach would be to partition the ranges and categories thereby mapping them onto binary attributes, allowing us to apply the foundational binary association rule techniques discussed earlier (e.g., Apriori algorithm). For example

- $Price = 1200 / Price \neq 1200$
- $Brand = Dell / Brand \neq Dell$

or if there are many values, use intervals:

- $Price \in [1000, 1500] / Price \notin [1000, 1500]$.

A transaction in this database would now contain (or not contain) the item *Price = 1200*, which could be mined like other items. Unfortunately, this partitioning can actually cause some contradictory problems. A few large intervals can encompass many tuples and thus have high support while many smaller intervals, while being more selective, may yield low and perhaps unacceptable support. On the other hand, a small interval, say $Age \in [21, 22]$ may provide for a nice specific high confidence rule such as

$$Age \in [21, 22] \rightarrow GraduatedCollege(80\%)$$

whereas widening the interval to gain support brings in more negative tuples lowering the rule confidence:

$$Age \in [18, 22] \rightarrow GraduatedCollege(40\%)$$

not to mention losing some of the rule's focus. A tighter interval may boost confidence but lower support and vice versa.

Srikant and Agrawal attempt to counter this by mining across all values or intervals, which have been coded with a new integer representation, and then selectively combining them to increase support, producing additional, more general "items" as there are called. For example, $Price \in [500, 999]$ and $Price \in [1000, 1499]$ may both be probed for support in the database, leading to $Price \in [500, 1499]$ being examined as well. All three could have minsupp and be used to find rules. The obvious problem here is that such a sweeping approach will incur much more computing time. They try to relieve this by adding a *maximum support* constraint to limit the combinations, though a lot of computing time will still be needed overall.

The other less apparent caveat is the output of an excessive number of rules which could be either redundant or simply not useful. For example,

$$Price \in [500, 1499] \Rightarrow \quad Mem \in [64, 128] \quad (75\%sup, 80\%conf)$$
$$Price \in [500, 999] \Rightarrow \quad Mem \in [64, 128] \quad (25\%sup, 78\%conf)$$

shows that the second rule is essentially contained in the first and hardly conveys any more information. On this front Srikant and Agrawal have offered an automated *measure of interest* calculation to prune such redundancies before presentation to the user.

While quantitative rules do open up another kind of attribute to association rule mining, Kuok, Fu and Wong (1998) contend that there are still some limitations that need to be dealt with. Namely, the harsh boundaries created by the partitioning of quantitative attributes into intervals cause an unnatural division of the data and subsequently of the resultant rules. For example, the intervals $Age \in [20 - 29]$ and $Age \in [30 - 39]$ create an abrupt shift at the 29 year point, though in reality there is none. This is essentially the same problem discussed earlier regarding the term *RICH* and the crisp cutoff point ($\geq \$80,000$) for membership in that set. Kuok et al. (1998) offer a similar remedy for rule mining.

Quite simply, they propose fitting quantitative rules with fuzzy set theory to create *Fuzzy Association Rules*. Instead of the above Age intervals, the attribute Age could be partitioned into linguistic fuzzy sets still backed by the intervals at the membership function level. For example, we could have $Age \in$ *young adult* and $Age \in$ *adult*, whose boundaries are fuzzy. Quantities of 21 - 28 could still have maximal membership (1.0) in the *young adult* set with membership beginning to decrease at 20 and 29. Ages 30 and above could also be members of the set but to a

lesser and lesser degree. Likewise, ages 29 and below would be members of the *adult* set to some degree.

Kuok, Fu and Wong (1998) offer this formalism for their Fuzzy Association Rule technique (cf. Agrawal, Imielinski and Swami above):

T is transaction database of n tuples t: $T = \{t_1, t_2, \ldots, t_n\}$

$\mathcal{I}$ is the set of all items or attributes (e.g., Age, Price, etc.) in the database; $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$

Each item I_k can have membership in its own fuzzy sets:

$$\begin{aligned} F_{I_k} &= \{f^1_{I_k}, f^2_{I_k}, \cdots, f^m_{I_k}\}, \text{ e.g.,} \\ F_{age} &= \{young\ adult, adult, middle-aged, senior\} \\ F_{height} &= \{short, average, tall\} \end{aligned}$$

A Fuzzy Association Rule is

$X\ is\ A \Rightarrow Y\ is\ B\ s.t.\ X \subseteq \mathcal{I} \wedge Y \subseteq \mathcal{I} \wedge X \cap Y = \emptyset$

Also

$A = \{f_{x_1}, f_{x_2}, \ldots, f_{x_p}\} \mid x_k \in X \wedge f_{x_k} \in F_{x_k}$

$B = \{f_{y_1}, f_{y_2}, \ldots, f_{y_q}\} \mid y_k \in Y \wedge f_{y_k} \in F_{y_k}$

Just as fuzzy set theory requires its own operations analogous to classical set operations, such as union, so too does fuzzy association rule mining. Boolean association rules involve finding all *large* or *frequent itemsets* which meet a user-specified *support* threshold, support being the percentage of all database tuples which contain all elements of the itemset in question. Kuok et al. (1998) present an analogous measure called *significance* to gauge an itemset's frequency in the database. The major difference is, since we are dealing with fuzzy sets, each item's presence is weighted by its fuzzy membership degree in [0,1] before being divided

by the number of tuples in the database T. In this respect, significance is similar to a fuzzy cardinality operation. Formally, Kuok, Fu and Wong (1998) give the *significance factor*, $S_{\langle X,A\rangle}$ as:

$$
\begin{aligned}
Significance &= \frac{\text{Sum of votes satisfying } \langle X, A\rangle}{\text{Number of tuples in T}} \\
S_{\langle X,A\rangle} &= \frac{\sum_{t_i \in T} \prod_{x_j \in X} \{\alpha_{a_j}(t_i[x_j]))\}}{total(T)} \\
&\text{where} \\
\alpha_{a_j}(t_i[x_j])) &= \begin{cases} \mu_{a_j}(t_i[x_j]) & \text{iff } \mu_{a_j} \geq \omega \\ 0 & \text{otherwise} \end{cases} \quad a_j \in A
\end{aligned}
$$

One thing to note here: the function α_{a_j} is a standard fuzzy operation known as an *alpha-cut* which is simply a threshold applied to a fuzzy set, in this case μ_{a_j}, that zeroes out memberships below a certain degree, in this case $0 < \omega \leq 1$.

Since we are using a product operation, any fuzzy membership that is zero, causes the measure for the entire tuple to be zero - the desired result. An unsupported attribute makes the entire attribute set useless for the significance calculation of that tuple. All tuple products, each having been weighted by the attribute fuzzy values, are then summed and compared to the database as a whole, giving the fuzzy support or significance. Notice that if this operation were applied to crisp sets, as long as $0 < \omega \leq 1$ (which it should always be), the products will always end up as either 0 or 1 and the final result will be the classical itemset support as in the original work by Agrawal, Imielinski and Swami (1993).

If there is a fuzzy analogy to support, it stands to reason that there is also one for *confidence*. Kuok et al. (1998) offer a calculation for the *certainty factor* of a resultant rule, based on the above defined significance.

Not surprisingly, this factor is directly related to the confidence measures of classical association rules, which can be calculated on a rule from itemset Y:

$$conf_{X \Rightarrow Y-X} = \frac{supp(Y)}{supp(X)} \text{ where } X \subset Y$$

In like fashion, Kuok et al. (1998) offer:

$$\text{Certainty} = C_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{S_{\langle Z,C\rangle}}{S_{\langle X,A\rangle}} \text{ where } Z = X \cup Y, C = A \cup B$$

for a rule 'If X is A then Y is B', which is a logical extension of the support/significance analogy.

Through application of these techniques, Kuok, Fu and Wong believe they present a more robust system for mining useful real-world rules on quantitative data, utilizing fuzzy set theory operations.

2.4 Spatial Data Mining

According to Fayyad et al.(1996), *Data Mining* is the algorithmic application step of the full KDD (*Knowledge Discovery in Databases*) process which also includes input preparation and output interpretation and utilization. Even so, it is a very broad term encompassing such subfields as clustering, neural networks, rule mining, etc. An important consideration when discussing data mining is the type of data being mined. General data mining literature usually centers on data typically found in a standard relational database system (RDBMS): retail transaction data,

personnel data, client data, and the like. For the past several years, Jiawei Han and Kris Koperski have been among the chief exponents of *Spatial Data Mining* which they describe as

> ... the extraction of implicit knowledge, spatial relations, or other patterns not explicitly stored in spatial databases (Koperski and Han 1995).

A spatial database system is generally one like an RDBMS that has support for spatial data structures (e.g., MBRs - Minimal Bounding Rectangles), spatial operations (e.g., Spatial Join), and general spatial access methods or SAMs (Koperski, Han and Adhikary 1998). That said, a spatial database would not be exclusively tasked with handling spatial data, but in RDBMS fashion, would need to hold or have access to non-spatial data related to the spatial objects as well. Simply having a spatial object A and spatial object B is not as useful as knowing also that object A is a house and object B is a lake. In fact one of the uses for spatial data mining is "discovering relationships between spatial and nonspatial data" (Koperski, Han and Adhikary 1998).

Along these lines, Koperski and Han (1995) have done work on adapting the techniques of regular database association rule mining as presented by Agrawal, Imielinski and Swami (1993) for use with spatial data. Like the originals, spatial association rules still take the form $X \Rightarrow Y(c\%)$ except rather than representing items or sets of items, X and Y now stand for spatial or nonspatial predicates(Koperski, Han and Adhikary 1998) such as $west\_of(x, y)$, spatial, since it deals with the objects location in space, or $isa(x, lake)$, nonspatial, since it describes the nature of the object but not its position. This is not far removed from the original association rule definition since these are *boolean rules* with *binary attributes* ("item

X is present or it is not"). The aforementioned predicates are essentially the same thing. Rules involving various combinations of these could be mined to uncover hidden information about the data. For example,

$$\begin{aligned} isa(x, golfcourse) &\Rightarrow contains(x, lake)(100\%) \\ isa(x, town) &\Rightarrow adjacent\_to(x, river)(81\%) \\ close\_to(x, beach) \wedge isa(x, house) &\Rightarrow price(x, high)(76\%) \end{aligned}$$

are spatial association rules saying golf courses always have lakes inside, towns are usually located on rivers, and houses by the beach are usually expensive, "usually" being a linguistic approximation of the given high confidence values. It should be noted that the concept of support still applies, so while the golf course/lake connection may be strong, there may not be many golf courses in our database making the rule rather insignificant. A strong rule would have both high support and high confidence (Koperski, Han and Adhikary 1998). Koperski and Han use algorithms similar to those of Agrawal et al. to build the requisite large itemsets or predicate sets, integrating features required by the Spatial DBMS.

Another instance of spatial data mining of interest to us is the mining of image or raster data. Contrary to the earlier quote, spatial data mining does not have to be on an explicit spatial database but can be performed on a large image database containing spatial data of a different kind, often from a remote sensing source. Koperski, Han and Adhikary (1998) say that this differs from ordinary "image processing" in scale:

> ... data mining studies very large amounts of data ... while image processing usually concentrates on analysis of single or a few images.

This kind of approach is applied when some kind of automated image analysis such as object detection or pattern matching is needed. A famous example is the work done by Burl et al. (1998) on recognizing volcanos on the planet Venus. Mining on 30,000+ radar images from the Magellan probe, they were able to achieve an 80% accuracy rate with their system. The system consisted of three major components:

1. Data focusing to examine the image pixels (the building block of the images) and their neighbors to decide on the image areas to proceed on;
2. Feature extraction to glean necessary attributes and metrics (e.g., eigenvectors) from the found areas;
3. Classification learning which actually answers the question "volcano/not volcano" or learns from human expert examples how to do this based on the attributes from the last component.

This example is important because much of the spatial data coming in from field observation (e.g., satellite radar, aerial photos, sea-floor SONAR, etc.) is indeed raw image data without the niceties of spatial objects structures, MBRs, etc. As such, much of the required work must be done here, at the image arrival point.

Even more applicable is the recent work of Ordonez and Omiecinski who have done a first investigation into association mining from raster image data (1999). Ordonez and Omiecinski are using existing image segmentation techniques, namely Berkeley's Blobworld system (Forsyth et al. 1997), customized preprocessing methods - the heart of the research, and applying traditional association rule algorithms, such at those of Agrawal et al. or others.

First the Blobworld process is applied to each image in the test image-base to extract the various shapes so that each shape in an image now has its own blob. The important step now it the preprocessing using similarity matching. Ordonez and Omiecinski use the various object attribute metrics (color, shape, position, etc.)

to calculate a similarity coefficient a given object versus any other objects, with the weight for each metric being user definable. Objects that are similar enough are considered to be the same object, e.g., the yellow square in image X is the same as the yellow square in image Y. In terms of the research in this thesis, the objects could be considered to be in the same class, and in terms of foundational association rules, they are essentially the same item.

This basically describes what Ordonez and Omiecinski are doing. As stated in their research:

> With typical basket-market analysis, the data is usually constrained to a [preexisting] fixed set of items that are explicitly labeled (Ordonez and Omiecinski 1999).

No such set of items exist here; the image data domain requires the itemset to be generated. By using similarity to constrain the shape objects to a small group, they create a shape itemset analogous the that of a retail itemset. Having its constituent objects so tagged, each image in the image base becomes a transaction analog, containing not beer and chips but shapes as its items. Once this meta-transaction base is built, neither the images or the blobs are needed any longer, except for visualization. Traditional association rule mining can now proceed to generate rules such as:

$$\{\ 4\ \} \Rightarrow \{\ 2\ \}\ (0.50, 1.00)$$
$$\{\ 4\ 7\ 11\ \} \Rightarrow \{\ 2\ \}\ (0.40, 1.00)\ \text{(Ordonez and Omiecinski 1999)}$$

which say that an image with a hexagon(4) will also have a triangle(2) with 100% confidence for the first, and for the second more complex rule, that an image with a

hexagon(4), a square(7), and a circle(11) will also contain a triangle(2), again with 100% confidence but lower support.

This whole approach by Ordonez and Omiecinski is somewhat similar to others such as Srikant and Agrawal (1996). That is, standard association rule techniques are designed for boolean attribute data, however we want to mine on non-boolean data such as quantitative or image data. If a method can be found to map the initial data into a boolean context, such as the similarity matching of Ordonez and Omiecinski or the range mapping of Srikant and Agrawal then the techniques from the established association rule tool box can be easily and effectively applied.

CHAPTER III

THEORY

3.1 Problem

Given a partitioned raster image, i.e., an image containing a set of discrete objects, or several such images, we wish to extract information about the relationships among these various objects in the image or across images. In other words, we wish to mine association rules describing the *general* spatial nature of the image components. Again, the focus here is exclusively on the direction component of a spatial relationship. Not currently taken into account are other attributes such as the distances between objects or the sizes of objects, whether relative to each other or absolute in the image space. Rather, we are currently interested only in simple rules constrained to the directions between objects. Furthermore, due to the inherent imprecision in directional relationships, in part due to morphology, as laid out by Bloch, we wish such rules to be fuzzy - indeed we cannot escape this fact. To investigate such a system we will need available: an object detection scheme, a spatial relation form, association rule form(s), including the appropriate support and confidence metrics, or reasonable analogs thereof, and a mining algorithm.

The object detection scheme will make available objects (crisp for our purposes) tagged with their size, location, and, most importantly for this investigation, their *class*. Yet while knowing the identity of the class is necessary, what exactly said class means is in and of itself not so important to the process but depends on the specific domain to which the process is being applied. And while it is true that we may

use natural numbers or alphabetic characters to denote classes, these are categorical attributes and not ordinal attributes. Class 2 is no closer to class 1 than class 7; class H is no closer to class G than class V. As such, we can interpret class as a simple categorical attribute as described in Srikant and Agrawal (1996). Extracting the relational meta-data from the objects requires what is for the most part an implementation of Bloch's techniques, with the output customized for our purposes. After the meta-data has been extracted, we will apply an association rule mining algorithm to these intermediate spatial relationship data to mine fuzzy association rules describing the *general* relationships among the image components. However, as mentioned in the Literature Review, the existing fuzzy rule mining techniques (Kuok, Fu and Wong 1998) are designed for the general milieu of "traditional" non-spatial databases (e.g., age, price, etc.). With that in mind we will have go back to basics and apply a simpler algorithm to provide useful and interesting generalized information at this stage.

Such a system will tackle some of the various unaddressed issues in the previously discussed literature via this integration. The fuzzy spatial relation work of Bloch usefully functions at the individual relation level, a specific object R related to a specific object A. The proposed system will mine these relationships to extract generalizations. The fuzzy association rules of Kuok, Fu and Wong add fuzziness to the classic associations of Agrawal, Imielinski and Swami, but principally apply it to ordinary non-spatial data. The proposed system will operate in a spatial/image domain. Ordonez and Omiecinski offer a first step into image data mining, transforming image data into traditional boolean transactions, thus enabling the application of regular retail-style rule mining, but they mainly focus on simple object co-occurrence, more basic than object spatial relationships, and they do not

utilize fuzzy set theory. And while Koperski, Han and Adhikary do offer a spatial variation of association rule mining, it does not utilize fuzzy set theory. Indeed they cite a "(f)uzzy sets approach (1998)" as one of the many future directions of spatial data mining. The system proposed here takes up that challenge.

3.2 Objects

We will assume that the source images are already partitioned and classified through some means, making available to us the constituent objects tagged with their size, location, and class. The class in question could theoretically have some domain specific significance important to the user, but that is not relevant for this work. We should mention here that implied in the object size and location is its morphology. Because we are dealing with raster image data, our objects are simply collections of pixels rather than the nested objects found in spatial databases. The algorithms we are using will need to perform operations on these various pixels. Therefore, we do not just want to know the location of an object, but the location of all its constituent pixels as well which define the object's morphology. As such the BlobWorld (Forsyth et al. 1997) used by Ordonez and Omiecinski (1999) would not be applicable here since objects are transformed into blobs and by definition lose their morphological identity.

3.3 Relations

Recall that we are using Dr. Isabelle Bloch's fuzzy spatial techniques (1999) for 2D space applied to partitioned image data (i.e., containing discrete objects) to produce orientation meta-data about the objects, or more specifically object pairs, contained therein. This meta-data takes the form of *fuzzy spatial relations*. However, rather than describing these fuzzy relations with single fuzzy membership values,

there are three values per relation forming an interval and a likely average estimate. The resultant values of Bloch's algorithm applied to the points in the object pair are evaluated in the forms 2.2, 2.3, and 2.4 seen on page 17 in the Literature Review. Given these resultant values as well as the input values (direction, object IDs, etc.) and adjunct data, namely the object classes, we can form descriptive relations about pairs of objects and their relative position. These descriptions will be used in turn to generate association rules. We describe the form of such relations as follows:

Given:

$\mathcal{O}$ = $\{o \mid o$ is an object in $S\}$, set of all objects in space.

$\mathcal{C}$ = $\{c \mid c$ is the class of $o \in \mathcal{O}\}$, set of all object class labels.

The relation:

$$isa(o_x, c_i) \text{ where } o_x \in \mathcal{O} \wedge c_i \in \mathcal{C}$$

is a classical, crisp relation associating an object o_x with a class label c_i.

$$D_\alpha = \{((o_r, o_q), N_\alpha^{o_r}(o_q), M_\alpha^{o_r}(o_q), \prod_\alpha^{o_r}(o_q)) \mid (o_r, o_q) \in \mathcal{O} \times \mathcal{O} \wedge o_r \neq o_q\}$$

is a fuzzy relation, D = "in direction of", or more specifically "o_q is in direction α of o_r." In contrast to typical fuzzy relation notation, we have explicitly replaced the single membership value, μ, with three values: the pessimistic necessity, N, the optimistic possibility, Π, and the overall mean, M. These are the same as 2.2, 2.3, and 2.4 discussed on page 17 with $R = o_r$ and $A = o_q$ in this case. They have the property:

$$0.0 \leq N \leq M \leq \Pi \leq 1.0.$$

In other words, the necessity and possibility form an interval in which is found the Mean - $M \in [N, \Pi]$ - in short, a *membership interval*. According to Bloch, this arrangement captures the inherent imprecision in the spatial relationship (1999).

For a given image-space, since we are examining the relationships between pairs of objects, we will find such relation tuples for all 2-permutations of the image's object

set $\mathcal{O}$. Thus, the number of such tuples is:

$$P(|\mathcal{O}|, 2) = \frac{|\mathcal{O}|!}{(|\mathcal{O}| - 2)!} = |\mathcal{O}| \cdot (|\mathcal{O}| - 1)$$

Note that we want permutations - not combinations since the spatial relationships between objects are not commutative.

Furthermore, we also do not want to limit ourselves to a single α direction but rather several *primitive* directions, such as *right*, *above*, *left* and *below* or

$$D = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$$

radians which can better describe the space in general. Moreover, we will eventually want to analyze not just a single image but a set of images. This gives us a transaction database of $(|\mathcal{O}_i| \cdot (|\mathcal{O}_i| - 1) \cdot |D|)$ tuples for each image i, where $\mathcal{O}_i$ is the set of objects for image i. For now, however, we will focus on applying our techniques to a single image populated with several objects and thus several object pairs.

3.4 Rule Forms

To actually find rules we will need some possible *rule forms* to use. Additionally, an important factor in these rules will be how the *Support* and *Confidence*, or the fuzzy spatial analogs thereof, will be determined. Recall that in traditional binary association rules, the support of a rule $X \Rightarrow Y$ is the percentage of all tuples in the database that contain the items XY, indicating the significance of the rule in the database at large, while the confidence is the percentage of all tuples containing X which also contain Y, indicating the strength of the rule itself.

It should be noted here that, as seen above, traditional association rules take the form of an implication, and as such many equate the term "rule" with implication. We will adopt a broader definition of association rules to potentially include a wider variety of logical assertions.

In the following section we explore several possibilities. We establish some rule forms and offer some examples using hand generated data. The forms fall under two categories which we call aggregated and traditional. We present the more radical aggregated approach and accompanying rule form first and work towards a more traditional form.

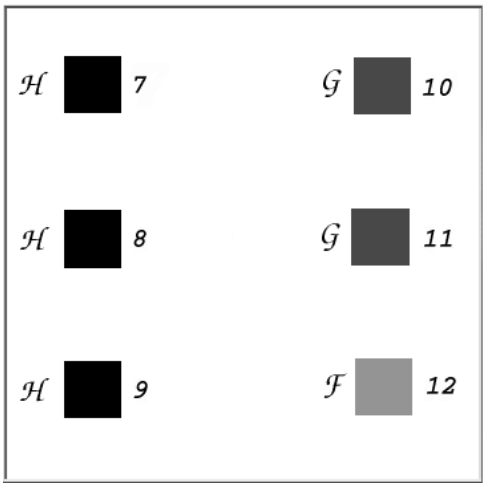

Figure 3.1: Sample objects with classes

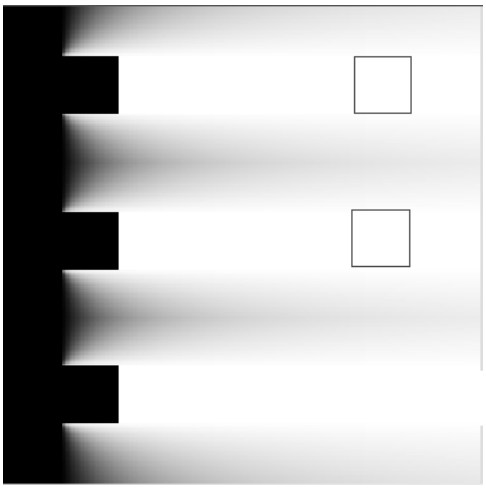

Figure 3.2: Landscape of Class $\mathcal{H}$in direction $\alpha = 0.000$ and phantom Class $\mathcal{G}$

3.4.1 Aggregated

In simplest terms association rules make generalizations about a relatively large amount of data. They in effect aggregate the data into like chunks, i.e., the rules. We take this concept a step further and do some aggregation *before* the

explicit rule generation stage. Separate objects of like classes are aggregated into non-contiguous pseudo-objects (or "class-objects") before the fuzzy spatial relation extraction stage, giving us one pseudo-object per class. Bloch's algorithm is then applied to these pseudo-objects as though they were normal objects (the algorithm makes no distinction) to generate tuples. Figure 3.2 depicts a landscape generated from Figure 3.1 through such means.

The result of this is that for any two given object classes in an image, there is one and only one tuple describing that pairing in a given direction. In essence the mining is done prior to or concurrent with tuple generation. As for the forms such rules take, a possibility is:

Class-Class

$$\begin{aligned} \forall x \forall y, \quad & class(A) \wedge pixel(x) \wedge inClass(x, A) \wedge \\ & class(B) \wedge pixel(y) \wedge inClass(y, B) \\ & \Rightarrow inDirection(\alpha, x, y)\ (S, C) \\ S \quad = \quad & Sum(NP_A, NP_B)/NP_{Obj} \\ C_N \quad = \quad & \text{Necessity} \\ C_M \quad = \quad & \text{Mean} \\ C_\Pi \quad = \quad & \text{Possibility} \end{aligned} \tag{3.1}$$

Or in more natural language:

> Given *any* class A pixel and a *any* class B pixel, it follows that the class B pixel will be located in direction α of the class A pixel with a certain confidence and certain standing in the database.

This a fairly strict rule due to the use of *any* (or *all* or *every*) which holds *all* such pixels to the arrangement.

One important issue here is the fact that since objects were aggregated into class-objects, it is difficult to even talk about true objects anymore - they no longer exist. Instead, we must talk in terms of the pixels that make up the objects, hence the predicate *pixel* simply meaning "is a pixel". So the rule part is rather simple: given *any* two pixels, x and y of the classes A and B, y is in the direction α, whatever it may be, of x. The issue then becomes the measurement of this rule, the S and C figures (these aren't exactly the same as traditional support and confidence, but we keep the initials to emphasize that there is an analogous relationship).

First, we have posited a sum calculation for S, namely a ratio of the sum of class A and B pixels to the total number of pixels in all objects (vs. total pixels in the image), NP_{Obj}. While we have lost individual object identities, we still have statistics about them like these. This seems to be closest to the traditional concept of support, being similar to XY itemsets presented above. But there may be other possibilities that might be more in line with Bloch spatial relations in which size can have an effect on a relation.

$$S_2 = Min(NP_A, NP_B)/NP_{Obj}$$

$$S_3 = Max(NP_A, NP_B)/NP_{Obj}$$

both produce smaller support values dependent on the size of the participating classes. For example a rule might be produced with high confidence (more on this momentarily) where one of the classes is very large in the image space, conferring high S upon the rule. However the other class might be very tiny and thus unimportant, at least in the given domain of discourse. As such, perhaps the cautiousness of allowing a small class to scale down a rule is merited - at least as an option.

As for C, if confidence indicates the strength of a rule, then the fuzzy memberships for the base tuples seem to be logical candidates for this metric. Moreover, in contrast to the S choices above, these are presented here as three values which should *all* be presented to the end user, rather than giving only one value as the final output. Recall once again that Bloch characterizes this membership interval as capturing the imprecision of a relation. As such, retaining all three values (or derivatives thereof) can be viewed not as noncommittal but rather informative.

Table 3.1: Objects

id	class	size	first
7	$\mathcal{H}$	961	(31, 26)
8	$\mathcal{H}$	961	(31, 110)
9	$\mathcal{H}$	961	(31, 192)
10	$\mathcal{G}$	961	(187, 109)
11	$\mathcal{G}$	961	(188, 27)
12	$\mathcal{F}$	961	(189, 190)

Table 3.2: Pseudo-Objects

id	class	size	first
13	$\mathcal{H}$	2883	(31, 26)
14	$\mathcal{G}$	1922	(187, 109)
15	$\mathcal{F}$	961	(189, 190)

Object #13 is aggregation of objects #7-9;
Object #14 is aggregation of objects #10,11;
Object #15 is aggregation of object #12;

An example using this form:

Class-Class Example (using tables 3.2 and A.2 from Figure 3.1)

$$
\begin{aligned}
\forall x \forall y, \quad & class(\mathcal{H}) \wedge pixel(x) \wedge inClass(x, \mathcal{H}) \wedge \\
& class(\mathcal{G}) \wedge pixel(y) \wedge inClass(y, \mathcal{G}) \\
& \Rightarrow inDirection(0.0, x, y)\ (S, C) \\
S \quad = \quad & Sum(2883, 1922)/5766 = 4805/5766 = 83.33\% \\
S_2 \quad = \quad & Min(2883, 1922)/5766 = 1922/5766 = 33.33\% \\
S_3 \quad = \quad & Max(2883, 1922)/5766 = 2883/5766 = 50.00\% \\
C_N \quad = \quad & 99.59\% \\
C_M \quad = \quad & 99.99\% \\
C_\Pi \quad = \quad & 100.00\%
\end{aligned}
$$

In words:

> Given *any* class $\mathcal{H}$pixel and *any* class $\mathcal{G}$pixel, it follows that the class $\mathcal{G}$pixel will be located in direction 0 (i.e., to the right) of the class $\mathcal{H}$pixel with a necessary (or minimum) confidence of 99.59% a possible (or maximum) confidence of 100.00% and a mean confidence of 99.99%. This is supported by 83.33% of the pixels in the database (i.e., image space).

The advantages of this approach are that it is fast, requires little intermediate storage, and incorporates the mining stage directly into relation extraction; rule generations becomes a trivial reformatting of the corresponding relations. The main disadvantage is the loss of individual object information. Another possible

disadvantage, or at least an unresolved problem, is how to manage rule mining across multiple images in a set. If objects are in different spaces, objects of a given class cannot all be aggregated and then spatially related to objects not in their same space. This could result in multiple tuples which will require a more explicit mining step again.

So while our agglomeration technique can streamline the mining process, the cost is a loss of individual object identity which could be a limiting factor for further knowledge discovery.

3.4.2 Traditional

A traditional mining approach entails viewing the various object-object relations as analogs to itemsets, though not necessarily to the degree of Ordonez and Omiecinski (1999). These individual object relations would then be generalized into association rules through an explicit mining stage.

One option is to use a modification of the basic form presented above:

Object-Object form #1

$$\begin{aligned}
\forall x \forall y, \quad & class(A) \wedge object(x) \wedge inClass(x, A) \wedge \\
& class(B) \wedge object(y) \wedge inClass(y, B) \\
& \Rightarrow inDirection(\alpha, x, y)\ (S, C)
\end{aligned} \tag{3.2}$$

$$S_p = Sum(NP_A, NP_B)/NP_{Obj}$$

$$S_o = Sum(NO_A, NO_B)/NO$$

$$C_i = \frac{\sum_{x \in A} \sum_{y \in B} (inDir_i(\alpha, x, y))}{NO_A + NO_B}$$

$$i \in \{N, M, \Pi\}$$

$$C_{pixel} = \textit{see future work}$$

Or:

> Given *any* class A object and a *any* class B object, it follows that the class B object will be located in direction α of the class A object with a certain confidence and certain standing in the database.

However, some things have changed. For one, we still have individual objects (notice the object predicate) so we have the option of making statements in terms of objects. If we need to work with pixels as in the Class-Class version, we would require some slightly different formulae. S_p could be the same as with Class-Class but the C_{pixel} series would require some additional calculation to be discussed in future work. Recall from Section 3.4.1 we did not have to actually do such a calculation.

Overall though, form 3.2 is the same as before except with the additional object variant. This is because, depending on the domain, we may not be very interested in object size, which number of pixels conveys, but simply object populations - one object, one vote, like the senate versus the house of representatives. Also, we could still use the minimum and maximum variants as well, but for the rest of this work we will concentrate on the sum version since, as said earlier, it is most like the traditional support measure. As for the min/max versions, we will be content to know that they are available if domain specific conditions call for them and their support dampening features.

The other major change is the determination of C. Because we have to do explicit mining to aggregate the tuple information we must do some summations and averaging of the fuzzy memberships. And it is essentially an average, summing over the objects, weighted by fuzzy membership, then dividing by the number of objects. What is still preserved is the notion of passing on the imprecision measures by generating three such C measures as a membership interval.

An example [1] using this form:

[1]NOTE: The problem with this example is that because the objects in the test image are of uniform size the two sets of S measures come out the same. This will not always be the case.

Object-Object 1 Example (using tables 3.1 and A.1 from Figure 3.1)

$$
\begin{aligned}
\forall x \forall y, \quad & class(\mathcal{H}) \wedge object(x) \wedge inClass(x, \mathcal{H}) \wedge \\
& class(\mathcal{G}) \wedge object(y) \wedge inClass(y, \mathcal{G}) \\
& \Rightarrow inDirection(0.0, x, y)\ (S, C) \\
S_p \quad = \quad & Sum(2883, 1922)/5766 = 4805/5766 = 83.33\% \\
S_o \quad = \quad & Sum(3, 2)/6 = 5/6 = 83.33\% \\
C_N \quad = \quad & 4.5438/6 = 75.73\% \\
C_M \quad = \quad & 4.8215/6 = 80.36\% \\
C_\Pi \quad = \quad & 5.0727/6 = 84.55\%
\end{aligned}
$$

Or:

> Given *any* class $\mathcal{H}$object and a *any* class $\mathcal{G}$object, it follows that the class $\mathcal{H}$object will be located in direction 0.0 (i.e., to the right) of the class $\mathcal{G}$object with a necessary (or minimum) confidence of 75.73% a possible (or maximum) confidence of 84.55% and a mean confidence of 80.36%. This is supported by 83.33% of the *objects* in the database (i.e., image space).

The major advantage here is we now have access to the objects again so we can potentially say more about a given rule, or we could relate an object to more than one other object in more than one direction at a time. The disadvantages are more processing time, and more storage. Also notice there is the unresolved problem of

a good C measure in pixel mode. Unlike the earlier example, there is no longer a single value (or three values) to simply use - one must be constructed.

The straight Object-Object tuple based mining also offers the possibility of additional rule forms such as the one given here:

Object-Object form #2

$$\begin{aligned}
\forall x \exists y, \quad & class(A) \wedge object(x) \wedge inClass(x, A) \wedge \\
& class(B) \wedge object(y) \wedge inClass(y, B) \wedge \\
& inDirection(\alpha, x, y)\ (S, C) \\
S_o \quad = \quad & Sum(NO_A, NO_B)/NO \\
S_p \quad = \quad & Sum(NP_A, NP_B)/NP_{Obj} \\
C_i \quad = \quad & \frac{\sum_{x \in A} \max_{y \in B}(inDir_i(\alpha, x, y))}{NO_A} \\
i \quad \in \quad & \{N, M, \Pi\} \\
C_{pixel} \quad = \quad & \textit{see future work}
\end{aligned} \tag{3.3}$$

Or:

> Given *any* class A object, we assert that there exists *some* class B object located in direction α of the class A object with a certain confidence and certain standing in the database.

The obvious difference is that this rule form is less demanding being existential rather than universal on the other object in question. That is to say, we are not concerned with every object in relation to the reference object - just the best one.

The other difference along with that is the absence of the implication in favor of a simple conjunction.

The existential rule also requires some different methods for finding the S and C measures. We could try to use one of the possible forms from the previous rule forms as shown in $S_{o,p}$ above, but in this case the NO_B is troubling. If we only need one instance of class B to exist, we should not count all the B objects, since this would be closer to standard support. We could go to the trouble of holding out the best y that exists per x and use just those to create a measure not presented above but that seems difficult. Since the focus is really on all A objects and whether they have a B object, maybe all that should really be counted is the A objects as in:

$$\begin{aligned} S_{2o} &= NO_A/NO \\ S_{2p} &= NP_A/NP_{Obj}. \end{aligned}$$

In some sense it is that last approach we adopt for the C measures, placing NO_A in the denominator. The numerator is similar to that of the previous rule form except that, because of the existential form, it requires only the max, i.e., the best that exists, membership value of the pair relations in question. However we still pass on the three value membership interval as before.

An example using this form:

Object-Object 2 Example (using tables 3.1 and A.1 from Figure 3.1)

$$
\begin{aligned}
\forall x \exists y, \quad & class(\mathcal{H}) \wedge object(x) \wedge inClass(x, \mathcal{H}) \wedge \\
& class(\mathcal{G}) \wedge object(y) \wedge inClass(y, (\mathcal{G}) \wedge \\
& inDirection(0.0, x, y)\ (S, C) \\
S_o \quad = \quad & Sum(3, 2)/6 = 5/6 = 83.33\% \\
S_p \quad = \quad & Sum(2883, 1922)/5766 = 4805/5766 = 83.33\% \\
S_{2o} \quad = \quad & 3/6 = 50.00\% \\
S_{2p} \quad = \quad & 2883/5766 = 50.00\% \\
C_N \quad = \quad & 2.6805/3 = 89.35\% \\
C_M \quad = \quad & 2.7588/3 = 91.96\% \\
C_\Pi \quad = \quad & 2.8233/3 = 94.11\%
\end{aligned}
$$

Or:

Given *any* class $\mathcal{H}$object, we assert that there exists *some* class $\mathcal{G}$object located in direction 0.0 (i.e., to the right) of the class $\mathcal{H}$object with a necessary (or minimum) confidence of 89.35% a possible (or maximum) confidence of 94.11% and a mean confidence of 91.96%. This is supported by 83.33% of the *objects* in the database (i.e., image space) based on *all* objects in the two classes.

3.4.3 Previous Work

In general, the resultant rules generated with these forms are similar to those discussed by Kuok, Fu and Wong (1998). Their work establishes a convenient framework for interpreting directional attributes. Recall from Section 2.3, page 25 that they describe a set of fuzzy sets

$$F_{i_k} = \{f^1_{i_k}, f^2_{i_k}, f^3_{i_k}, f^n_{i_k}\}$$

that are associated with an attribute i_k in the set of attributes I. In our case, the set D of primitive directions represents a partitioning of the attribute *direction* aka D_α into such fuzzy sets

$$F_{D_\alpha} = \{f^1_{D_\alpha}, f^2_{D_\alpha}, f^3_{D_\alpha}, f^4_{D_\alpha}\} = \{Right, Above, Left, Below\}$$

The direction field of ordered pair of objects will have some degree of membership in each of these four fuzzy sets. Such an arrangement is visualized in Figure 3.3.

For the sake of consistency, one could also view the $isa(object_x, class_i)$ relation this way with sets for the class categorical attribute:

$$F_{class} = \{f^1_{class}, f^2_{class}, \ldots, f^n_{class}\} \text{ where } n = |\mathcal{C}|, \text{ the number of classes.}$$

Of course, *isa()* is a classical relation so the above actually represents the crisp set specialization of a fuzzy set where membership is in $\{0, 1\}$ rather than in $[0, 1]$. This leads to a slight difference in form. For a given tuple, the class attribute can have full membership in one and only one of these sets, e.g., $Object_{12}$ cannot be in $Class_{\mathcal{H}}$

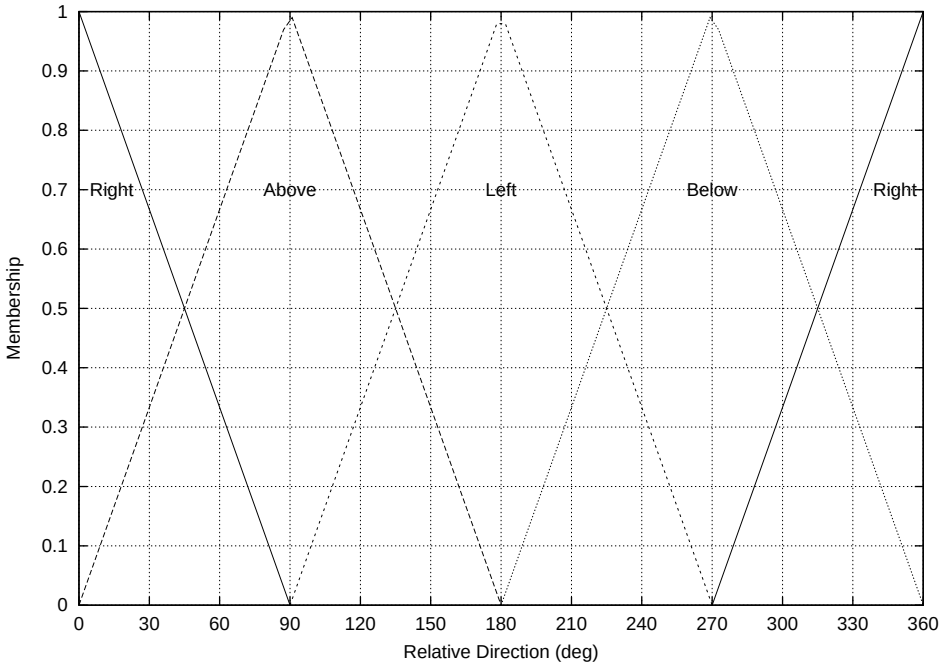

Figure 3.3: Primitive direction fuzzy sets

AND $Class_{\mathcal{M}}$. Thus there is never more than one candidate for an objects class unlike fuzzy direction.

3.4.4 Mining

Moving from the relation forms to the above rule forms requires some sort of mining algorithm. In some sense, the algorithms used here have already been presented in the guise of the S and C metrics and their formulae, since for the purposes of this particular investigation, the main task of the mining stage is calculation of these figures. The Class-Class form in 3.2 requires no mining at all but merely reformatting of the aggregate relations into the rule form with the S and C values pulled directly from the relation and object lists. The Object-Object forms on the other hand are not pre-aggregated and thus require something more like a traditional mining algorithm to make our generalized association rules. That

said, this preliminary investigation is looking at very simple small rules so a highly optimized miner with high-level pruning capabilities is not used. Moreover, due to this basic nature of the rules, their skeletal forms are already known; a reference class, some other class, and a compass point uniquely determine a simple rule for a given form (such a combination can occur for each form used). Based on this, the mining algorithm collects the relevant tuples, i.e., those of like class-class-dir, and using their fuzzy metrics and general statistics from all the relations in an image or images, aka the relation base, applies the formulae presented in forms 3.2, 3.2, and 3.4 to calculate the proper S and C values. Those data can then be formatted to the form specifications and presented to the user as basic spatial associations.

CHAPTER IV

EXPERIMENTAL RESULTS

To test the theories given in Chapter III a series of experiments were run demonstrating the possibilities of basic fuzzy spatial association rules. A small system was built and run on SunOS and Linux/Intel platforms. We were fortunate to be able to make use of parallelism via MPI on a Linux cluster which sped up the test runs with minimal additional effort. We generated several data sets for testing: a few hand constructed for preliminary testing, most programmatically generated by a custom synthetic data generator, but also a real world example using categorized seafloor data from another project. Below we present some details about the test platform and its components, the data used for testing, and results of those tests.

4.1 Test Platform

The main components of our test system are the *Fuzzy Spatial Relation Extractor* (FSRE), the *Rule Miner*, and a *Synthetic Data Generator* (SDG). The first two are discussed below while the SDG is discussed in Section 4.4. We also make extensive use of an external application called *mpiShell*, which was developed by Bruce Wooley (2000), to facilitate parallelism. It is discussed in Section 4.3.

The main component is the FSRE which, as the name implies, extracts fuzzy spatial relations from classified raster image data. It is a custom object-oriented C++ implementation of Bloch's landscape approximation algorithm or propagation algorithm (1999) for 2D crisp objects. This is a greedy algorithm that

dynamically bypasses much of the calculation of vector angles and generates an approximate landscape that differs very little from an exhaustively calculated one, while benefitting from a potential *20*× increase in speed according to some of Bloch's tests (1999). The FSRE iterates over all the shape objects treating each one in turn as a reference object and approximating the landscape for the reference object for each specified direction. The intersection of every other shape object with each landscape is then isolated "cookie cutter" style and the membership interval of the relation is calculated to arrive at the fuzzy spatial relation. As mentioned earlier, an object aggregation capability is built into the FSRE to offer the style of mining from Section 3.4.1. To keep the memory footprint low, only one landscape is allocated at any given time. The output is a table of fuzzy spatial relation meta-data suitable for mining. One could also consider the object detector to be a component, but since object detection is not the focus of this work, a very simple object detector has been folded into the FSRE and runs as a preprocessing stage of that program.

The RuleMiner is a Perl script that assembles the relations generated with FSRE into our proposed rule forms. It is a basic implementation of the mining approach detailed in Section 3.4.4. It will autodetect the relation source, e.g., Class-Class aggregated vs. Object-Object, and in the case of the former will drop into a simple formatting mode to change aggregate relations directly in associations. Otherwise it will gather relations with the same class-class-direction fingerprint and derive the S and C metrics.

4.2 Complexity

Recall from section 3.3 on page 38, we are extracting $s \cdot (s-1)$ tuples (every shape object related to every other shape object) and we are doing it for d directions.

This yields $O(d \cdot s^2)$ time complexity. However, the most computationally expensive step is landscape generation. We only have to do landscape generation once per object per direction so that is actually $O(d \cdot s)$. Since d is typically a small constant, e.g., $d = 4$, the complexity is $O(s)$.

According to Bloch (1999), the propagation landscape generation algorithm has a complexity of $O((1 + 2n_V)N)$ or $O(N \cdot n_V)$ where N is the number of pixels in the image and n_V is the neighborhood of each point. If we always use a typical 2D neighborhood of 8, then this reduces to $O(N)$. Relation extraction for a given object A is linear, $O(n_A)$ where n_A is the number of points in A.

From this we can say that for a given reference object, generating the landscape and finding all the relations in one direction costs $O(N+s \cdot n_A)$, the $O(s)$ being taken from above. This assumes that all the objects are size n_A which may not always be the case. However, since we are doing this for multiple objects, we could substitute for n_A the mean object size, m_A. Doing this for all reference objects results in $O(s(N + s \cdot m_A))$. Either way, we could say that $s \cdot m_A$ (or equally the sum of all n_A pixel counts over all s shapes) is simply the total number of object pixels less those of the reference object, $N_O - n_R$, giving us $O(s(N + (N_O - n_R)))$ or $O(s(N + N_O))$. If our image space is totally populated with objects, i.e., almost all pixels are in objects with little or no void (classless) pixels, then N_O approaches N. In any case, $N_O \leq N$ always holds. This gives us $O(s(N + N))$ or $O(sN)$, holding d and n_V constant at 4 and 8 respectively. In other words, relation extraction is on the order of the number of shape objects times the number of pixels in the image space.

As for the actual miner, it is rather simple. We simply collect all the like relations, i.e. those of matching alpha, and classes and calculate the association. This can actually be done in linear $O(R)$ time where R is the number of relations.

However, this is understating the true complexity since the algorithm implementation requires the input to be already sorted in memory, so in reality we can only get $O(RlogR)$. The space requirement is still $O(R)$, since in the current implementation, we keep the full relation table in memory.

4.3 Parallelism

The approximation of a landscape for a given reference object (or reference pseudo-object) R is completely independent of any other object. The subsequent relation extraction from that landscape is dependent on the availability of other objects (with which R is related), but it is still independent of the other landscapes. In other words, the analysis of reference object R with all other objects in direction D can be viewed as an independent procedure. This means each R, D pair can be potentially assigned to a different CPU without concern about further communication. This is embarrassingly parallel. Usually, implementing for parallel processing using the Message Passing Interface (MPI) standard can be a daunting prospect, since the function calls and flow of control can be quite complex. An embarrassingly parallel problem is much less so since internodal communication is minimal. In this case, such implementation was further facilitated by utilizing the services of the *mpiShell*, an MPI wrapper program by Bruce Wooley (2000). The mpiShell obviated the need for extensive changes in the FSRE implementation, instead requiring only the addition of MPI size and rank parameters. These allow a given FSRE process to use mod and div operations to decide which of the pool of shape objects to work on and which to leave to other nodes. The mpiShell abstracts away the various MPI scatters, gathers, broadcasts, etc., and greatly accelerates the development cycle. With this in place, it became rather simple to use 8, 12, 16,

or more CPUs from the Linux cluster for a test image and greatly enabled us to complete more experiments in a shorter period of wall clock time. Actual speedup achieved was far from linear with a 16 CPU job generally finishing about 4 times as fast a single CPU run on the same machines. But that timing encompasses the entire run including:

- copying the image data across the network,
- object detection which must be run on *all* CPUs for the whole image,
- copying and concatenating all the results back across the network.

Redundant object detection and more importantly file operations can be detrimental to the timing. That said, 2m30s is a lot less waiting time than 13m15s as seen in time tests, especially considering the minimal development time required to take advantage of mpiShell's services.

4.4 Test Data and Results

4.4.1 Simple Hand-Constructed Data

We began by running the system on small simple hand-constructed images with a few objects of a few known classes in readily apparent spatial configurations. One such image containing six objects (on an empty classless background) in three classes was presented as Figure 3.1. Some sample rules were also presented.

In Figure 4.1 we present another simple example still with three classes but now with slightly more objects. Certain aspects of the spatial arrangement are very apparent to the human eye. We expect strong rules in the left and right directions. In order to present an overall picture of the number of rules generated and the support and confidence for those rules, we have generated a scatter plot for the rules where the x-axis is support and the y-axis is confidence. High support and confidence rules are in the upper right. This is what we mean by "plotting the rules" - plotting the

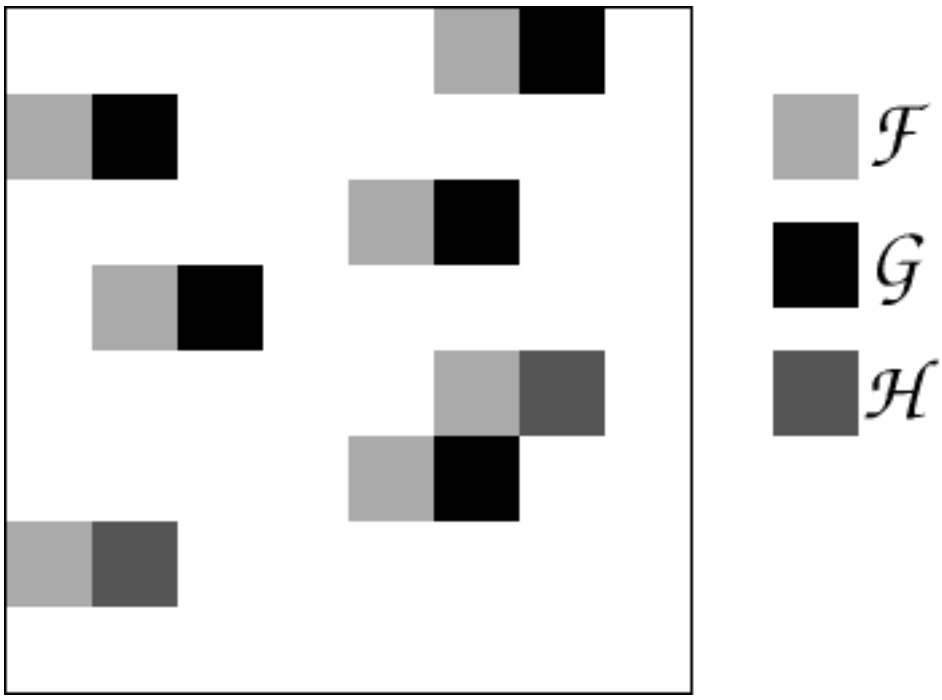

Figure 4.1: A hand-constructed three-class image

metrics of the rules. We have found this to be an effective way of visualizing and evaluating the large numbers of spatial association rules produced. The graph for rules about Figure 4.1 is shown in Figure 4.2. While we have our three values to use for confidence, plotting all three of them for all rules would be difficult to read, especially as we generate more rules in larger images, so for these plots we have chosen to show only C_M, the mean value within the interval. There are two main point sets, one for the more rigorous Universal rule set (form 3.2) and another for the Existential (3.4). Because the Universal rules, whose form starts $\forall x \forall y$, are more rigorous in that they require all A to be related to all R in a certain direction, their plots tend to cluster at the bottom of the graphs while the Existential rules, $\forall x \exists y$, are more lenient requiring only on A class object and so can achieve higher confidence values. Figure 4.2 exhibits this characteristic as well. This image particularly emphasizes

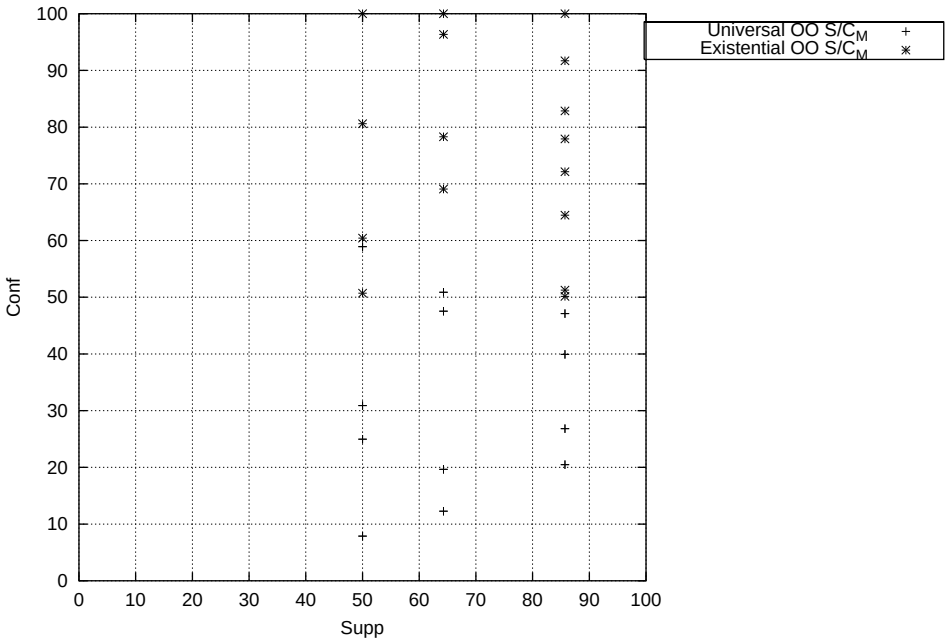

Figure 4.2: Graph of rule metrics, S & C_M for rule mined from Figure 4.1. Rules are in two series corresponding to the two forms used: *Universal* for form 3.2 and *Existential* for form 3.4.

the Existential rules; the image shows objects with "existing" companion objects and several existential rules have scored well on the graph. A rule one would expect, that the objects of the class visualized in light grey (call it $\mathcal{F}$) have black class objects (call them class $\mathcal{G}$) to their right, i.e.,

$$\begin{aligned}\forall x \exists y, \quad & class(\mathcal{F}) \wedge object(x) \wedge inClass(x, \mathcal{F}) \wedge \\ & class(\mathcal{G}) \wedge object(y) \wedge inClass(y, \mathcal{G}) \wedge \\ & inDirection(0.000, x, y)(85.71, 86.15 \leq 91.69 \leq 99.66),\end{aligned}$$

does score well at the second plot down in the upper-right of graph at $C_M = 91.69$. However this is not the highest scoring rule. That one is actually the opposite, i.e., reference class $\mathcal{G}$objects with class $\mathcal{F}$objects to the *left*, i.e.,

$$\begin{aligned}\forall x \exists y, \quad & class(\mathcal{G}) \wedge object(x) \wedge inClass(x, \mathcal{G}) \wedge \\ & class(\mathcal{F}) \wedge object(y) \wedge inClass(y, \mathcal{F}) \wedge \\ & inDirection(\pi, x, y)(85.71, 100.00 \leq 100.00 \leq 100.00),\end{aligned}$$

which scores $C_M = 100$. While every reference object in $\mathcal{G}$there does indeed exist another object in $\mathcal{F}$, the same cannot be said for the other direction. It is peculiarities like these that the rule miner can extract.

4.4.2 Synthetic Data

The hand-constructed images can only demonstrate so much and constructing sufficiently populated images is rather tedious. Therefore we constructed a Synthetic Data Generator (SDG) to produce additional images. C++-style pseudocode of the algorithm is presented in Figure 4.3. The SDG operates on the concept of *bias*,

specifically *directional bias*. The user specifies two classes, reference class (R) and other class (A) out of some total number of classes (C), a direction between them (α) and a percentage bias (b) for this direction. While the code offered in Figure 4.3 is for the direction $\alpha = 0$ (right), the only difference for the other three directions is the order of grid traversal (lines 16 and 17) which effectively controls direction. Objects are randomly generated on a grid of user-defined size, with each class, including R and A having an equal probability $1/C$ of appearing at a grid point. However when an object of class R appears, the bias goes into effect and the next object in the designated direction has a biased probability $b/100$ of being in class A, otherwise it becomes one of the other classes (including R) with probability $(100 - b)/100$. Each individual remaining class accounts for

$$\frac{100 - b}{100} \cdot \frac{1}{C - 1} = \frac{100 - b}{100 * (C - 1)}$$

of total probability.

An additional option called "Extended" is offered. When this is activated, the bias remains in full effect from the first occurrence of R until the edge of the image in the specified direction. Without it, the bias is only in effect for the object *immediately* after the R class object. This option should cause differences in overall support since more class A objects should be generated in extended mode. This could also have effects on the confidence interval of Existential (form 3.4) rules versus Universal (form 3.2) rules, since the additional class A objects could boost the Universal figures.

We also incorporated another algorithm called "Half" which produces spatial relations more readily visible to the human observer. Pseudocode is offered in Figure 4.4. When activated, A objects are biased to one half of the image and disallowed from the other half in the specified direction. This is a much simpler algorithm since

```
// C++ pseudocode algorithm for generating a biased synthetic image
// in the right or east direction. The other three directions
// (Left, Above, Below) are analogous, the main difference being
// the order of grid traversal
void SDGenerator::generate_Right()
{
   int R_class;    // Our reference class
   int A_class;    // Our other "biased" class
   int numClasses; // total number of classes including R & A
   int bias;       // Our probability %age in [1..100]                    10
   int biasRoll;   // Roll of a 100 sided die for bias percentage
   int classRoll;  // Roll for chosing classes
   bool useBias; // Flag indicating Reference class has been seen
                 // and therefore bias is in effect
   useBias = false;
   for(int y = 0; y < rows; y++) { // Generate from Top to Bottom
      for(int x = 0; x < cols; x++) { // Generate from Left to Right
         if(useBias) {
            biasRoll = Roll_a_100_sided_die();
            if(biasRoll <= bias)  // Did it fall within influence of bias?  20
               classRoll = A_class; // ...then force our biased A_class
            else // Pick a class OTHER than A_class
               // Probability of OTHER classes is (100-bias)/100*(numClasses-1)
               classRoll = Roll_a_numClasses-1_sided_die(); // no A_class
         } else // no bias
            // Probability of EACH class is 1/numClasses
            classRoll = Roll_a_numClasses_sided_die(); // Pick any class

         image[x][y] = classRoll; // Put our chosen class in the grid
                                                                          30
         if(classRoll == R_class) useBias = true; // Flag if we just saw R
         else if(NOT EXTENDED MODE) useBias = false; // reset if no extension
         // ELSE we ARE in EXTENDED MODE and the bias can persist for the
         // duration of this row, provided it is already set
      }
      useBias = false; // reset for next row - extended doesn't matter here
   }
}
```

Figure 4.3: Synthetic algorithm pseudocode

```
// C++ pseudocode algorithm for generating a biased synthetic image
// in the right or east direction, using image half method. The other three
// directions (Left, Above, Below) are analogous, the main difference being
// the order of grid traversal
void SDGenerator::generateHalf_Right()
{
   int R_class;    // Our reference class
   int A_class;    // Our other "biased" class
   int numClasses; // total number of classes including R & A
   int bias;       // Our probability %age in [1..100]                    10
   int biasRoll;   // Roll of a 100 sided die for bias percentage
   int classRoll;  // Roll for chosing classes

   // Generate from Left to Right
   // — Generate left half first - random
   for(int x=0; x < cols/2; x++) // Generate from Left to midpoint
      for(int y=0; y < rows; y++) // Generate from Top to Bottom
         image[x][y] = Roll_a_numClasses−1_sided_die(); // excludes A_class

   // — Generate right half - A_class biased                              20
   for(int x=cols/2; x < cols; x++) // Generate from midpoint to Right
      for(int y=0; y < rows; y++) // Generate from Top to Bottom
      {
         biasRoll = Roll_a_100_sided_die();
         if(biasRoll <= bias)  // Did it fall within influence of bias?
            classRoll = A_class; // ...then force our biased A_class
         else  // Pick a class OTHER than A_class
            // Probability of OTHER classes is (100-bias)/100*(numClasses-1)
            classRoll = Roll_a_numClasses−1_sided_die(); // excludes A_class
                                                                           30
         image[x][y] = classRoll;
      }
}
```

Figure 4.4: Synthetic half algorithm pseudocode

bias is not based on the class went before but simply where in the image we are, i.e., what half. The biased roll of $b/100$ is continuously applied in that half of the image. Again the remaining classed each account for

$$\frac{100-b}{100} \cdot \frac{1}{C-1} = \frac{100-b}{100 * (C-1)}$$

of total probability in that half. In the other half, A is disallowed and the other classes each get probability $1/(C-1)$. In this case, R is not a factor; rather this is a demonstration built around A. It is easy to see how this should bias the resultant rules, especially the Universal form, when objects are positioned in space with more stringent conditions.

The goal of the SDG experiments is to see if the FSRE/RuleMiner combination identifies rules that reflect the specified biases. There are several complicating factors when analyzing the results of these experiments.

1. *Small Number Effect*: The bias is simply a probability and not a strict proportion. An input of 95% does not guarantee a precise proportion of 95% class R's in direction α of class A's, and thus rules of exact 95% confidence. Since the numbers of objects in these experiments is relatively small (64), the Law of Large Numbers does not come into play, and the results vary from run to run.

2. *Edge Effect*: Also contributing is the problem of image edges. Say we have an class R object on the right edge of our image and we are investigating the "right of" relationship. This object can never have such a relation with other objects.

3. *Bias Side Effects*: Because these are images of several objects of several classes (say six), there is the high potential for *spillover*. This term refers to the tendency for a specified bias to influence class relationships other than those requested via the SDG - an unavoidable effect in a multi-class image.

   (a) *Co-directional Effect*: In one kind of side effect, demonstrated in Figure 4.5, a user-specified R class object X may have a user-specified A class object Y to its right which is the desired effect. But another unspecified

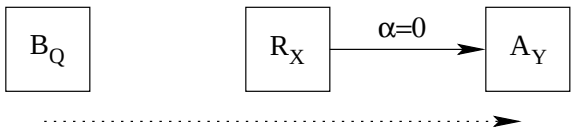

Figure 4.5: Co-directional example: Object A_Y is in direction $\alpha = 0$ of R_X - and of B_Q as well.

object Q of class B to the *left* of the first object X also has object Y to its right, emphasized with the dotted line. This does not necessarily interfere with the XY relationship, but it does produce the unintended side-effect of a QY relationship, and thus a more confident rule on QY (or rather classes BA).

(b) *Population Effect*: This spillover extends to direction as well. By biasing the generation of the A class we are increasing its overall population, and thus support, in the image. It is inevitable that there will be several R objects in directions other than the specified one thus boosting rules in that direction. Again these are unspecified side effects.

(c) *Counter-bias Effect*: The bias does not prevent the existence of relationships counter to the bias. That is, asking for a 90% "right

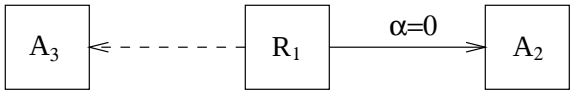

Figure 4.6: Counter-bias example: Object A_3 is not in direction $\alpha = 0$ of R_1

of" bias between two classes, R and A in no way prevents the occurrence of "left of" relations between those classes. An example is depicted in Figure 4.6. Object A_2 is in specified direction $\alpha = 0$ of R_1 - but A_3 is not. This is a perfectly valid arrangement while contrary to the requested bias. This can downgrade rule confidence, mainly with the Universal rule forms, because the opposing direction has low fuzzy membership, and by extension, confidence, in the specified direction fuzzy set.

These complicating factors must be considered when evaluating results. Being aware of these caveats, mainly that the bias is not hard and fast, we can still use the SDG to create synthetic test data.

For our tests we generally use 256×256 images with 32×32 pixel objects for a total of $8 \times 8 = 64$ objects. For most of our images, we consistently use six classes and a direction of $\alpha = 0$ with the same classes, which we will call $\mathcal{G}$and $\mathcal{H}$, chosen as our biased reference class R and other class A respectively.[1] These classes are consistently visualized as the same colors across examples. Having adopted those specifications, several sets of synthetic images were examined with the FSRE/Miner. The following is a representative sample of those tests.

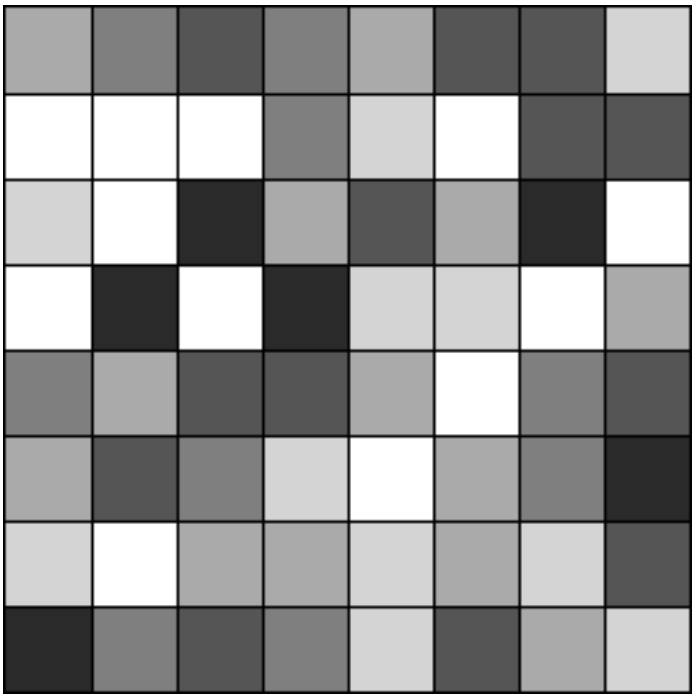

Figure 4.7: A randomly generated, i.e., no bias, six-class image

First we show an image that is not a synthetic dataset per se, at least not a biased one. Figure 4.7 is a simple random image with no biased direction or classes.

[1]We are using R and A in the *general* sense of Reference and Other class - that is, as *variables* in our system. We are using $\mathcal{G}$and $\mathcal{H}$as *specific* instances of these in our particular examples, i.e., as *constants*. So $R = \mathcal{G}$ and $A = \mathcal{H}$.

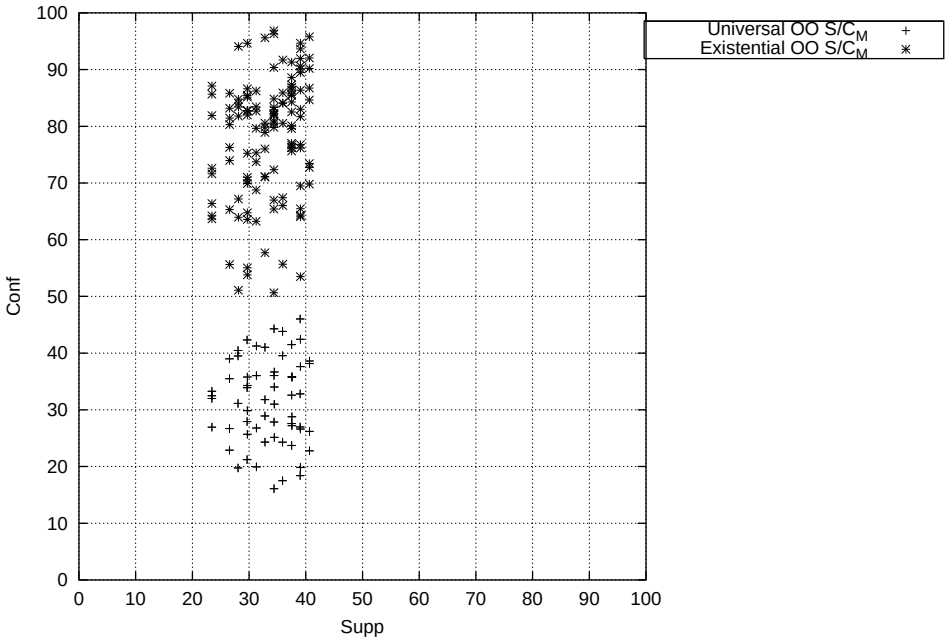

Figure 4.8: Scatter plot of rule metrics for Figure 4.7, a random biasless image. As before, there are two plot series, one for each of the rule forms we are focusing on.

The results of rule mining are plotted in Figure 4.8 with support on the x-axis and confidence on the y-axis. Again we have chosen the C_M value for graphing and have two point sets. Most of these random rules cluster in the relatively low 30-40% support range. This biasless sample establishes a basis for comparison for our biased samples below.

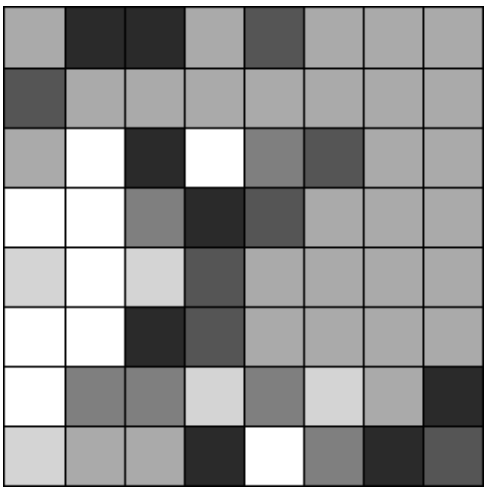

Figure 4.9: Image generated with specifications 90% extended bias, $R = \mathcal{G}$ and $A = \mathcal{H}$

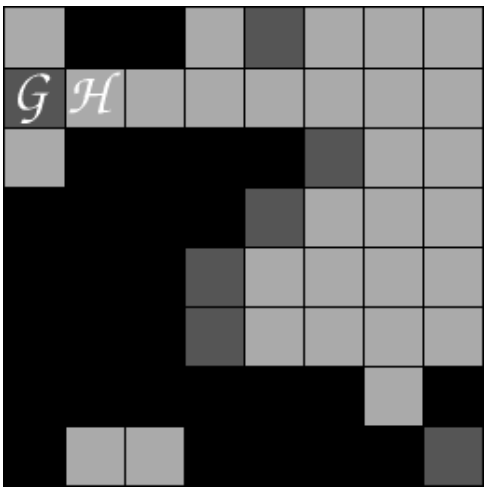

Figure 4.10: Isolation of classes $\mathcal{G}$and $\mathcal{H}$from Figure 4.9 with the other classes changed to empty space to give a better of view of their relationship.

Figure 4.9 was generated with a high bias of 90%, between the two classes with the added option of bias extension activated. The results of this can be easily seen when the classes in question are isolated in Figure 4.10. In this respect the bias has performed closer to 100% with the A class generated fully from R objects rightward to the edge. The class A objects not to the right of an R class object, such as those on the bottom, were randomly generated with the general object population.

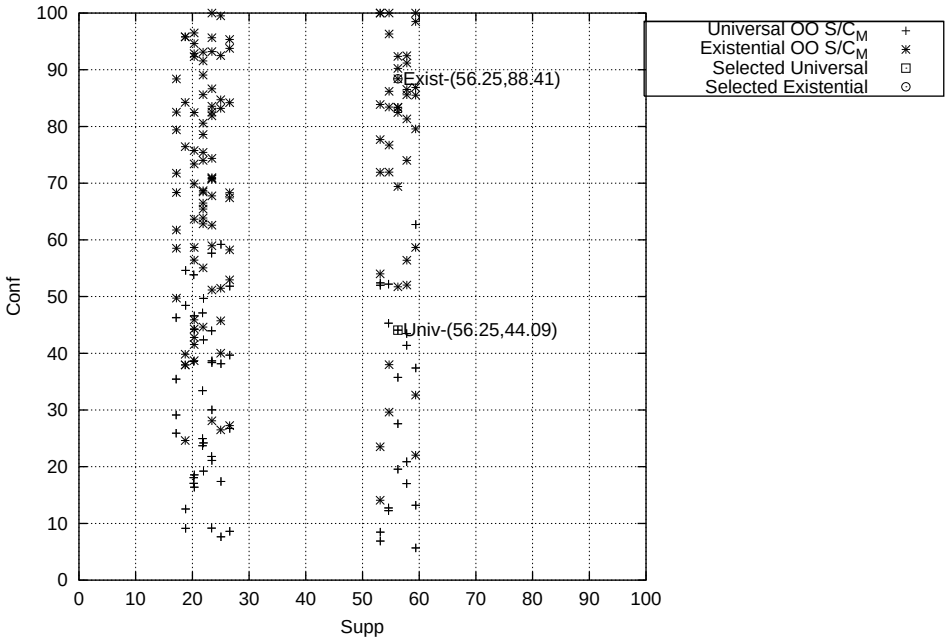

Figure 4.11: Scatter plot of rule metrics for Figure 4.9. In addition to a full plot series for each of the two rule forms, the rule plot points for both forms corresponding to the SDG specifications have been *selected* out as separate respective series and labeled with S and C_M.

Even with this high extended biased performance, the rules we are focused on, identified by the specified *R,A,Right* signature, i.e., the parameters used as SDG input, did not achieve maximum confidence and this can be traced to item #2 of the caveats mentioned above - the *Edge Effect*. The reference class object in the bottom right corner has no A class object to the right but it still plays a role in rule confidence calculation. Consequently only six of the seven R objects fulfill the rule. $6/7 \approx 86\%$ which is close to the C_M value of 88 in the Existential rule.

The Universal rule is helped by the extended bias which can make it more likely that *all* A class objects will be to the right of R class objects. However, the Universal rule is hindered by the existence of A objects on the wrong side, i.e., to the left as was discussed in item #3c above - the *Counter-bias Effect*. But considering the Universal form is a more rigorous rule it did turn out relatively strong at $C_M = 44\%$.

One interesting effect to notice is the banding seen in Figure 4.11. This is attributable to the side-effects mentioned in items #3b, the *Population Effect*, and #3a,. *Co-directional Effect*. The high bias and more importantly the extension option have boosted the overall population of class A objects in the image. This in turn has elevated the support of all rules that include class A, either as reference objects or other objects because for these experiments we compute it as a sum. The *Co-directional Effect*. manifests itself in the rules with confidences even higher that our focus rule. Class objects in these rules generally occur further to the left or possibly below the other classes and thus get a better "view" of the class A objects in their landscapes, thereby boosting their C values. Also some of these classes do not suffer from edges and that detriment to their rule metrics. Rules with class A as the reference class can benefit from their own high population for rules going

in the other directions (left and down) but do suffer from its edge objects in those directions.

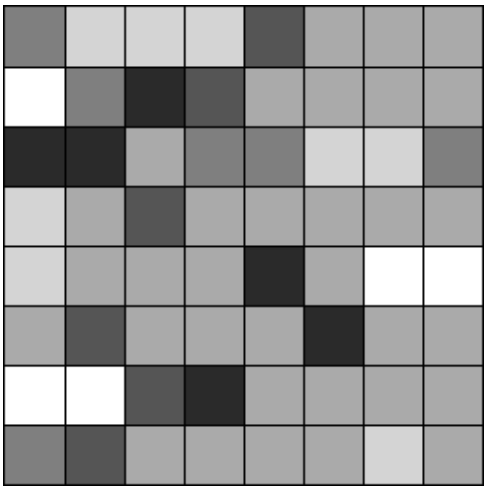

Figure 4.12: Image generated with specifications 80% extended bias, $R = \mathcal{G}$ and $A = \mathcal{H}$

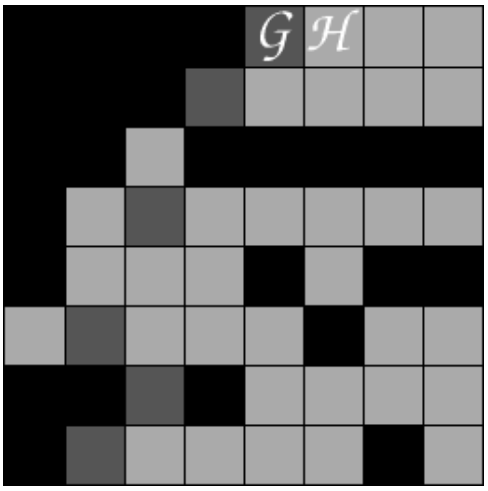

Figure 4.13: Isolation of classes $\mathcal{G}$and $\mathcal{H}$from Figure 4.12 with the other classes changed to empty space to give a better of view of their relationship.

Figure 4.12 also derives from a high extended bias, although not as high as the last example, using 80%. The isolated visualization in Figure 4.13 shows this with not all spaces to the right of our R objects being populated with A objects, most notably on the lower three rows. Ironically, this lower bias example has resulted in higher confidence rules, with both the stricter Universal form and the more lenient Existential form occurring at the top of their sets, the Existential one scoring $C_M = 100\%$. This is due to the fact that the R class has no object on the right edge to bring it down and that the Existential rule form only looks for the best A class relation for any given R object. The A class objects on the very left do not trouble the Existential form but are rejected. The extreme left A objects do negatively affect the Universal

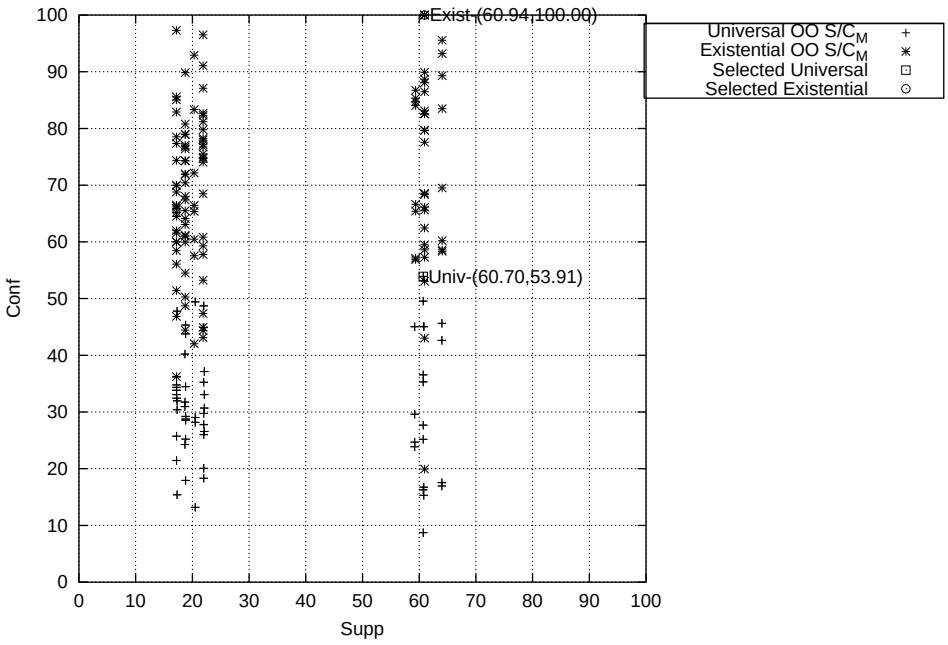

Figure 4.14: Scatter plot of rule metrics for Figure 4.12. In addition to a full plot series for each of the two rule forms, the rule plot points for both forms corresponding to the SDG parameters have been *selected* out as separate respective series and labeled with their S and C_M measures.

rule, and keep it from scoring as high as it could have, but the extended occurrence of A objects on the right do boost it and it still scores very well at $C_M \approx 54\%$. In general, this also illustrates side-effect #1, the *Small Number Effect*. That is, at these sizes, the lower 80% bias input here will not always manifest itself in worse rule output versus the higher 90% bias in the previous example. Overall this Figure 4.14 again displays the banding caused by the extended bias.

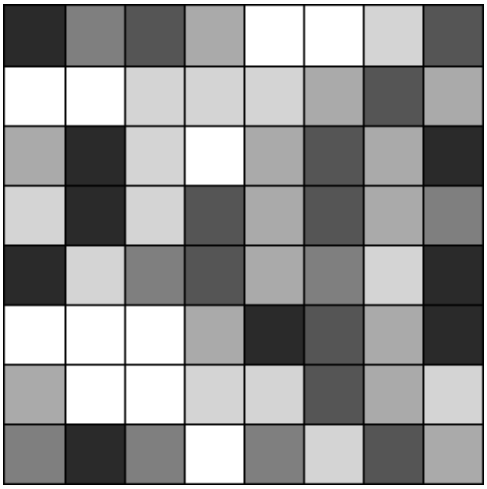

Figure 4.15: Image generated with specifications 95% bias,$R = \mathcal{G}$ and $A = \mathcal{H}$

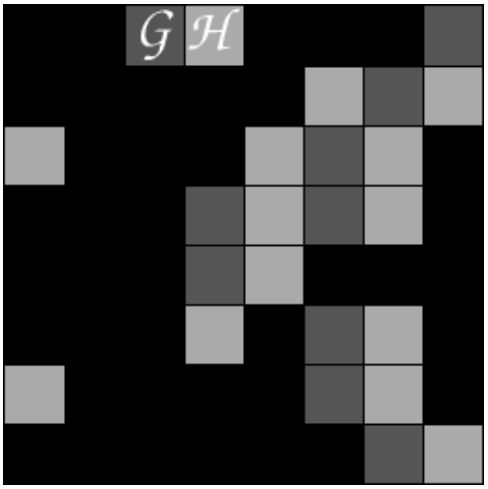

Figure 4.16: Isolation of classes $\mathcal{G}$and $\mathcal{H}$from Figure 4.15 with the other classes changed to empty space to give a better of view of their relationship.

Figure 4.15 displays a very high bias of 95% but in this case the extended option has been deactivated; only objects immediately to the right of R class objects are biased towards class A, then the bias resets as generation proceeds. Thus the banding formed from extension has vanished from the graph in Figure 4.17 with support ending up lower across the board and clustering together. Also because of this the Universal rule of our specified focus ($\mathcal{G}$, $\mathcal{H}$, Right) is no longer distinguished

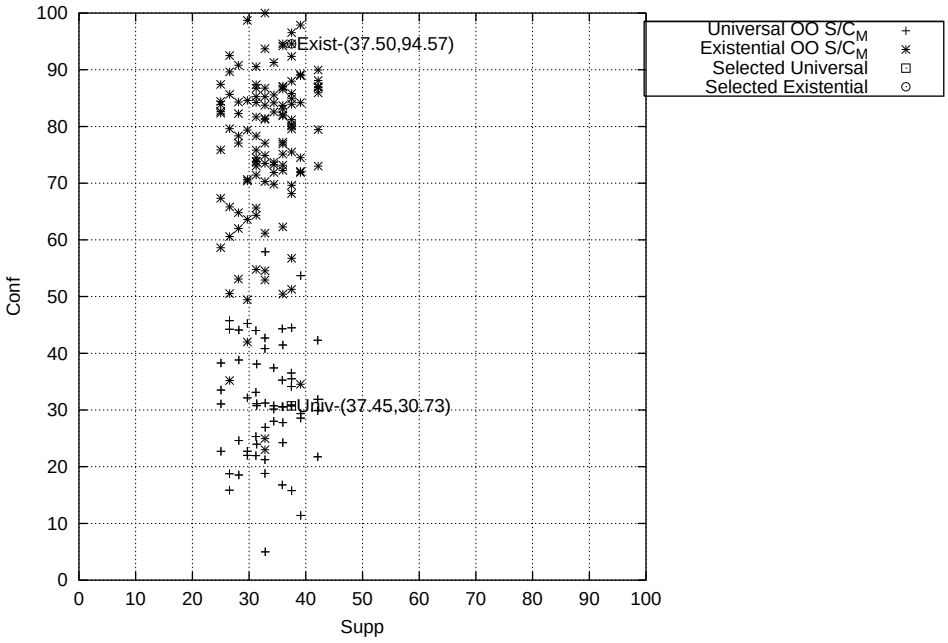

Figure 4.17: Scatter plot of rule metrics for Figure 4.15. In addition to a full plot series for each of the two rule forms, the rule plot points for both forms corresponding to the SDG parameters have been *selected* out as separate respective series and labeled with their S and C_M measures.

in its plot set; there is no high population to the right to boost it. In contrast, the Existential rule in question is still quite high in support and confidence, lowered mainly by the upper right edge class object. This example demonstrates the different demands of these two forms.

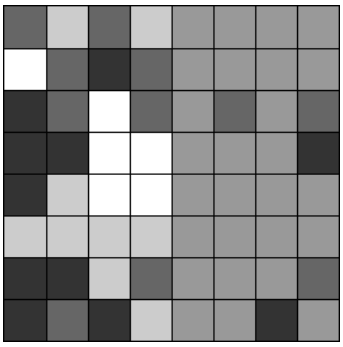

Figure 4.18: Image generated with 85% bias for class $\mathcal{I}$in the right ($\alpha = 0$) half

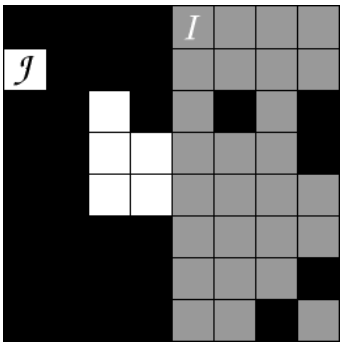

Figure 4.19: Isolated classes $\mathcal{J}$and $\mathcal{I}$from Figure 4.18

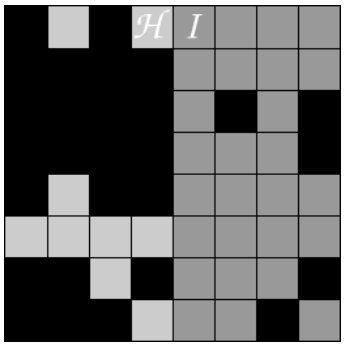

Figure 4.20: Isolated classes $\mathcal{H}$and $\mathcal{I}$from Figure 4.18

In the final synthetic example of Figure 4.18 we have activated the "Half" feature which, rather than biasing the occurrence of the A objects relative to R objects, biases them in an absolute half of the image, in this case the right half with a bias of 80%. Since the bias is with respect to a space, there is no particular chosen reference class. Furthermore, this arrangement demonstrates a situation which the Universal rule form is suited to since all A objects are more likely to be in the chosen direction of R classes. We should see increases using several reference classes. However, while the Existential rules using that A class to the right should generally all achieve 100% confidence as seen in the Figure 4.21 graph, not all Universal rules will reach the same level. We have extracted two class pairs in Figures 4.19 and 4.20 to show this. The $\mathcal{JI}$ rule has lower support than the $\mathcal{HI}$ one because there

are slightly fewer $\mathcal{J}$ objects than class $\mathcal{H}$. But more interestingly, $\mathcal{JI}$ rule has high confidence than the $\mathcal{HI}$ one, though both have high confidences than was seen in the previous examples. This is because of the reference object positions and the nature of landscapes. Recall that landscapes which visualize fuzzy spatial relation membership have a conical or spotlight nature with membership higher in the center of the cone and lower near the edges (see Figure 2.2). Because the $\mathcal{J}$ objects are nearer the center of the image, more $\mathcal{I}$ can fall in the higher membership portions of the landscape and, just as importantly, fewer will fall in the lower membership areas. In this way, the $\mathcal{I}$ objects as a group are more to the right of $\mathcal{J}$ objects than are of $\mathcal{H}$ objects. This is basically what the rule tells us through its C_M measure.

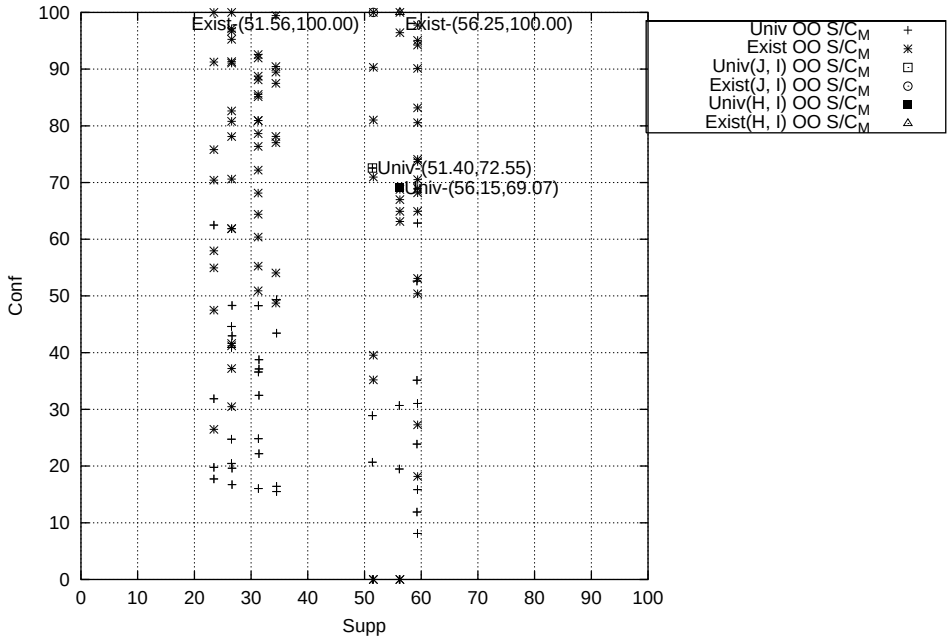

Figure 4.21: Scatter plot of rule metrics for Figure 4.18. In this case we have generated two additional subseries per each full series to illustrate those particular rules.

4.4.3 Real World Example

For a real world test we ran the system on some selected classified seafloor SONAR data from the OKEANOS project (Bridges et al. 1998), the spatial arrangement of which was unknown. One such images is shown in Figure 4.22. This image represents a classified sonar image. Before the test the exact spatial arrangement of the class regions is not known.

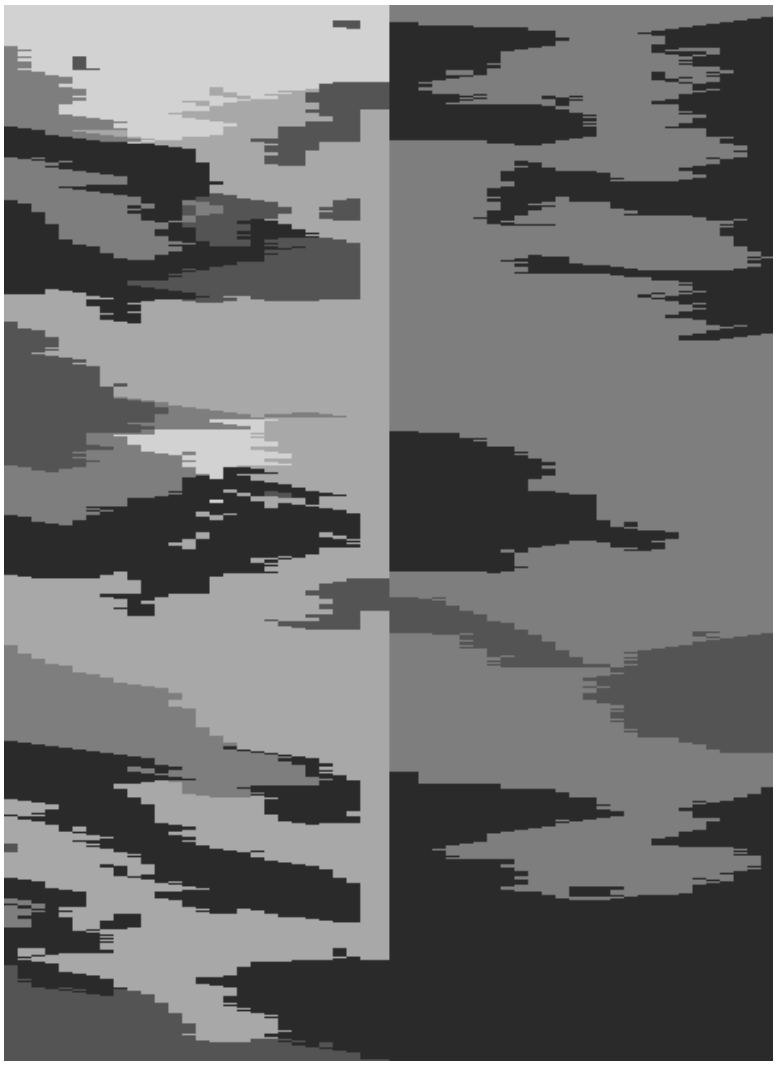

Figure 4.22: Classified seafloor image

This image was run through the system and the gathered rules have been plotted in Figure 4.23. Again we chose the mean C measure for display. While there are

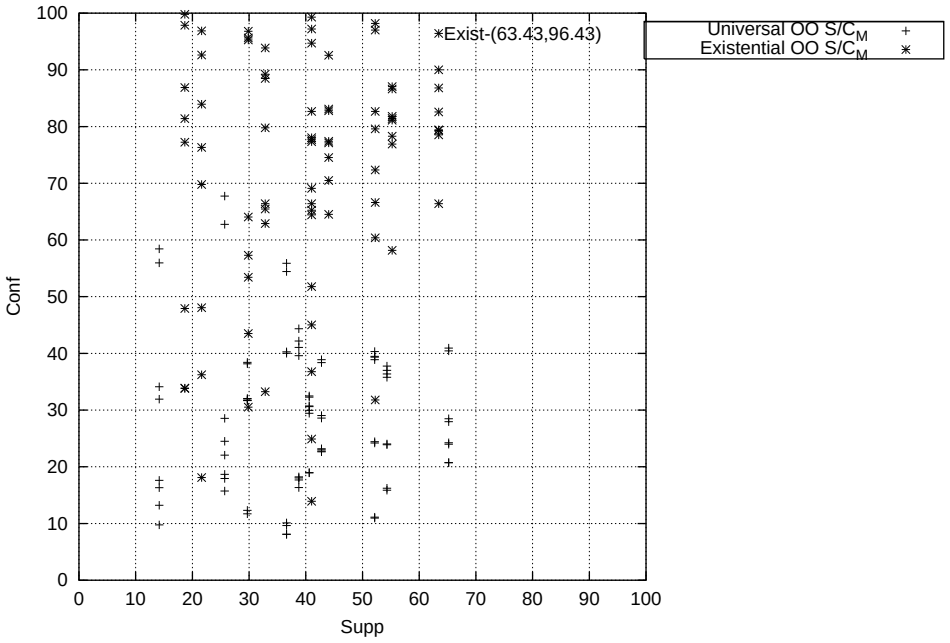

Figure 4.23: Scatter plot of rule metrics for Figure 4.22

several rules that could be useful here, one in particular looks promising from the graph and that is the Existential (marked *) rule in the top-right of the plot. This plot corresponds to rule #148,

$$\begin{aligned}\forall x \exists y, \quad & class(\mathcal{K}) \wedge object(x) \wedge inClass(x, \mathcal{K}) \wedge \\ & class(\mathcal{L}) \wedge object(y) \wedge inClass(y, \mathcal{L}) \wedge \\ & inDirection(\frac{3\pi}{2}, x, y)(63.43, 94.50 \leq 96.43 \leq 100.00),\end{aligned}$$

which states that objects of class $\mathcal{K}$will have some object of class $\mathcal{L}$below them with a very high confidence interval and a high support sum as well. In Figure 4.24 we have

isolated the two relevant classes and set the rest to void, with class $\mathcal{K}$, the reference class, visualized in a medium grey and class $\mathcal{L}$, the other class, visualized in white for better contrast. It now becomes more apparent, especially on the left side, that the rule does accurately describe these object classes in the image. Moreover, while distance is currently not taken into account, the corresponding objects would satisfy such a test as well since they are almost always immediately touching.

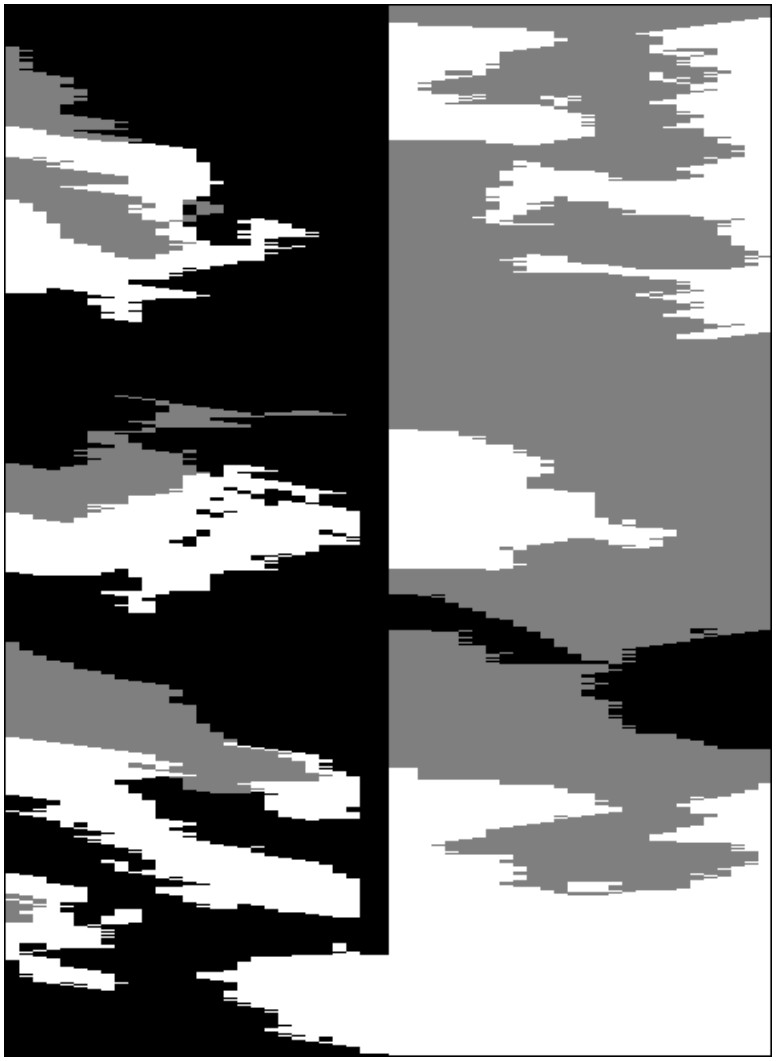

Figure 4.24: Isolated classes from seafloor image

This shows that a fuzzy spatial rule mining system such as this one does have potential usefulness in applications to real world data to discover hitherto unknown associations.

CHAPTER V

CONCLUSION

5.1 Summary

We have shown that the field of spatial rule mining, particularly fuzzy spatial rule mining, is still in its youth and ripe for new development especially regarding spatial positioning rules. A literature review was done in Chapter 2 in several related subfields to reviewing the current state of the science. Although there is a wealth of research in the area of image analysis, the application of newer data mining techniques, such as association rule mining, established in 1993 by Agrawal et al., to image data is relatively unexplored. Work has been done in spatial databases, or object space, as noted by Koperski, Adhikary and Han (1996) and preliminary forays into the image domain have been done by the likes of Ordonez and Omiecinski (1999). Likewise Srikant and Agrawal (1995) and others have extended the foundational association rules of Agrawal, Imielinski and Swami (1993), and Kuok, Fu and Wong (1998) have added fuzzy set theory, but only as applied to conventional data - not to image space. In an attempt to fill in this gap, we presented a theoretical framework for fuzzyspatial association rule mining.

We have adapted the techniques of Bloch to transform images into intermediate meta-data, a little like what Ordonez and Omiecinski do, except that where the concern of Ordonez and Omiecinski is simple object co-occurrence, we are exploring the more complex relationship of spatial arrangement. We can then mine rules from the meta-data describing the general arrangement of classified objects in the image

space. Just as with Kuok, Fu and Wong, a key is defining rule forms that will produce useful and interesting rules, and we offer rule forms somewhat like theirs using a set of fuzzy sets. In our case the fuzzy sets describe fuzzy directional relations between pairs of objects. We constructed two main rule forms for generalizing object relations: one which utilizes the logic of the existential form which attempts to find the best relationships between class and a more rigorous one which enforces a universal logic, i.e., one that all object relations must satisfy. We also offered a more radical albeit more limited rule form that operates at the class level rather than the object level by making use of pre-mining object aggregations. Vital to these rule forms are the measures of confidence and support which require a departure from the methods of Kuok, Fu and Wong, which did not suit our needs. We adopted the fuzzy membership interval of Bloch, and figures derived thereof as confidence. For support, object population sums were used.

We conducted a series of experiments to demonstrate the effectiveness of our rule forms and the properties they display. The experiments were run mainly on synthetic data generator by a custom program designed expressly for that purpose. The program produced images with a bias for specific class objects in specific directions and specific proportions. In essence the images were "loaded" and it was the system's job to find with what they were loaded. The experiments showed that the rule forms could generate useful rules that detected the planted relationships fairly reasonably. The two forms' performance varied on different types of images highlighting the distinction between the forms themselves. We also noted where the rule forms could stumble and offered reasons and caveats for this. A final test was run on a real world classified SONAR image with unknown relationships to see what could be extracted. A particularly good rule was mined and found to be accurate and

interesting, demonstrating the potential for such a system's applicability to unknown real-world data.

This investigation presented one possible preliminary framework for effectively analyzing spatial relationships in raster image data. Such a system could be used to mine useful and interesting fuzzy spatial association rules. Possible future avenues of exploration in this discipline based on this investigation are presented below.

5.2 Future Work

The above forms are not an exhaustive enumeration of all possible forms. The traditional object-object relation model would appear to offer several more possibilities. For example, in the forms given we generally only deal with two classes at a time, but since an object occurs several times in the table in relation to other objects of several classes, we could create more complex forms involving multiple classes, e.g., something to the effect of "Objects of class A have objects of class B OR class C to their left" or "Objects of class A are found to the right of class B AND below class C objects". We can check a given object to see if it conforms to such multiple conditions. The latter example also brings to mind an additional extension - that of considering multiple directions in a rule. The basic form of Kuok, Fu and Wong only allows for one fuzzy set per attribute. To work around this, instead of viewing direction as one attribute with four associated fuzzy sets, we could view it as four attributes D_{α_z} each with its own single fuzzy set in which it has some degree of membership. This would allow the attribute set Y to potentially contain several directional attributes simultaneously and give a rule taking them all into account. Such combination might allow the analysis of intermediate directions such as $\pi/4$ or perhaps more exotic object arrangements such as the case where an object

is completely surrounded by another in the form of a ring. Bloch mentions the possibility of complex relationship derivation from the four primitive relationships (1999). The same approach could possibly yield more complex rules as well.

There is also the potential of the membership interval which became the *confidence interval*. The most likely candidate for use in rule mining, at least initially, is the mean, M, which is most akin to a single membership degree, being an average of such values and thus the one we have used in presenting our examples. However, since we do generate the other two values which form the membership range of the spatial relation, they do have meaning and thus could be useful for mining. These could be split off to form *sub-attributes* with their own analogous fuzzy sets in the style of Kuok, Fu and Wong (1998):

$$\begin{aligned} F_{D_\alpha N} &= \{f^1_{D_\alpha N}, f^2_{D_\alpha N}, f^3_{D_\alpha N}, f^4_{D_\alpha N}\} \\ F_{D_\alpha M} &= \{f^1_{D_\alpha M}, f^2_{D_\alpha M}, f^3_{D_\alpha M}, f^4_{D_\alpha M}\} \\ F_{D_\alpha \Pi} &= \{f^1_{D_\alpha \Pi}, f^2_{D_\alpha \Pi}, f^3_{D_\alpha \Pi}, f^4_{D_\alpha \Pi}\} \\ &= \{Right, Above, Left, Below\} \end{aligned}$$

each one using the membership value indicated. Fuzzy rules could then be generated taking some or all of these factors into account, e.g., "Class A objects are minimally to the right of Class D object AND maximally to the right of Class E objects."

A side effect of these two above extensions is that if we do indeed separate the directions this way and also use the previously discussed membership partitioning for necessity and possibility, we could end up with direction becoming twelve separate, albeit related, attributes:

	Right	Above	Left	Below
Necessity	$D_{\alpha_0}N$	$D_{\alpha_{\pi/2}}N$	$D_{\alpha_\pi}N$	$D_{\alpha_{3\pi/2}}N$
Mean	$D_{\alpha_0}M$	$D_{\alpha_{\pi/2}}M$	$D_{\alpha_\pi}M$	$D_{\alpha_{3\pi/2}}M$
Possibility	$D_{\alpha_0}\Pi$	$D_{\alpha_{\pi/2}}\Pi$	$D_{\alpha_\pi}\Pi$	$D_{\alpha_{3\pi/2}}\Pi$

This expansion could make mining more difficult, but on the other hand it could allow for more expressive rules.

In addition a distance attribute would also make the rules more descriptive. In particular it would offset the effects of the infinite landscape which says that an object X 100 units distant from object Y can be equally to the right of object Y as and object Z only one unit away. This could be another fuzzy attribute: $F_{proximity} = \{adjacent, near, medium, far\}$. With distance included, spatial rules could be more useful.

There is the matter of image edges seen in the Chapter 4 examples. Are the detrimental effects of edge objects on rules a valid concern? By the same token, could we dampen the effects of edge objects and still have valid rules? Such edge compensation might be a feature worthy of investigation.

On a more practical level there is the matter of efficiency or rather inefficiency caused by excessive mining of weak rules. Traditional association rule mining algorithms all use some kind of itemset pruning based on the support of a given itemset in the database. Each pruned itemset means that much less mining is required. The current implementation of this system does not take a support measurement into account in the initial phases. Is there a characteristic analogous to small itemsets in our system? Yes and no. We are *not* dealing with itemsets per se and thus itemset support cannot be measured. All objects are spatially related to

all other objects; such a relation may simply have rather low membership. However, we can know a class's support level in a space before spatial relation extraction, and by extension we can know the support of a pair of classes with our present sum definition of support. If such a support level fell below a desired threshold, we could eliminate those relations before they are even generated. On the other hand though, relation extraction in and of itself is not the most expensive operation in this framework. Rather it is landscape generation that is computationally expensive, even with an approximation algorithm. Pruning may not offer relief if it only comes after landscape generation. For example, $supp(AB)$, the sum of class A and class B objects, may be too low but $supp(C)$ may be large making $supp(AC)$ and $supp(BC)$ also large. This happens because our support is union based rather than traditional intersection based support. This requires the generation of class A and class B landscapes so that AC and BC relations can be extracted even though AB and BA relations are unnecessary. This could make support level pruning very difficult to implement.

Finally, there is an alternative approach more directly like Agrawal's original association rule mining, still based on Bloch relations and (Kuok, Fu and Wong 1998) fuzzy association rules but also directly on (Ordonez and Omiecinski 1999). Just as Ordonez and Omiecinski use rule mining to generalize simple object co-occurrence: "Images containing a class U object also contain a class V object" with some support and confidence across an image base. We could extend that to rules about *object spatial relation* co-occurrence. Since the source relations are fuzzy, we would explicitly require the techniques of (Kuok, Fu and Wong 1998) to handle the mining. Note that this would not generate the same kind of rules as this investigation,

which attempts to associate one class of object with another in a direction, but rather ones that associate relations with other relations:

> A class I object has a class J object to its Right $\Rightarrow$ A class K object has a class L object Above it (S, C).

A system implementing this might make a nice complement to the one tested here.

REFERENCES

Agrawal, R., T. Imielinski, and A. Swami. 1993. Mining associations between sets of items in massive databases. In *Proceedings of the 1993 ACM SIGMOD int'l conference on management of data held in Washington, DC, May 26-28, 1993*, 207–216. ACM Press.

Agrawal, R., and R. Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of 20th international conference on very large data bases (VLDB'94) held in Santiago, Chile, September 12-15, 1994*, edited by J. B. Bocca, M. Jarke, and C. Zaniolo, 487–499. Morgan Kaufmann.

Bellman, R., and M. Giertz. 1973. On the analytic formalism of the theory of fuzzy sets. *Information Sciences* 5:149–156.

Black, M. 1937. Vagueness: An exercise in logical analysis. *Philosophy of Science* 4:427–455.

Bloch, I. 1999. Fuzzy relative position between objects in image processing: A morphological approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(7):657–664.

Bridges, S., J. Hodges, B. Wooley, D. Karpovich, and G. B. Smith. 1998. Knowledge discovery in an oceanographic database. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies* 11:135–48.

Brule, J. 1985. *Fuzzy systems - a tutorial.* http://life.csu.edu.au/complex/tutorials/fuzzy.html (Accessed 12 Jul 2000).

Burl, M. C., L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler, and J. Aubele. 1998. Learning to recognize volcanoes on venus. *Machine Learning* 30:165–194.

Chan, K. C. C., and W.-H. Au. 1997. Mining fuzzy association rules. In *Proceedings of the sixth international conference on information and knowledge management (CIKM'97) held in Las Vegas, Nevada, November*

10-14, 1997, edited by F. Golshani and K. Makki, 209–215.

Chen, M.-S., J. Han, and P. S. Yu. 1996. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6):866–883.

Cheung, D. W.-L., J. Han, V. Ng, and C. Y. Wong. 1996. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proceedings of the twelfth international conference on data engineering (ICDE'96) held in New Orleans, February 26 - March 1, 1996*, edited by S. Y. W. Su, 106–114. IEEE Computer Society.

Dubois, D., and H. Prade. 1980. *Fuzzy sets and systems: Theory and applications*, Volume 144 of *Mathematics in science and engineering*. New York: Academic Press, Inc.

Fayyad, U. M., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds. 1996. *Advances in knowledge discovery and data mining*. Menlo Park, CA: AAAI/MIT Press.

Forsyth, D., J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. 1997. *Finding pictures of objects in large collections of images*. Berkeley, CA: U.C. Berkeley, CS Division. Technical report.

Güting, R. H. 1994. An introduction to spatial database systems. *VLDB Journal* 3(4):357–399.

Han, J., K. Koperski, and N. Stefanovic. 1997. Geominer: A system prototype for spatial data mining. In *Proceedings of the 1997 ACM SIGMOD international conference on management of data held in Tucson, Arizona, May 13-15, 1997*, edited by J. Peckham, 553–556. ACM Press.

Holsheimer, M., and A. Siebes. 1994. *Data mining: The search of knowledge in databases*. Amsterdam: CWI. Technical Report CS-R9406.

Knorr, E. M., and R. T. Ng. 1996. Finding aggregate proximity relationships and commonalities in spatial data mining. *IEEE Transactions on Knowledge and Data Engineering* 8(6):884–897.

Koperski, K., J. Adhikary, and J. Han. 1996. Knowledge discovery in spatial databases: Progress and challenges. In *Proceedings of the 1996 ACM SIGMOD workshop on research issues on data mining and knowledge*

discovery (DMKD'96) held in Montréal, June 2, 1996, 55–70. IRIS/Precarn.

Koperski, K., and J. Han. 1995. Discovery of spatial association rules in geographic information databases. In *Advances in spatial databases: Proceedings of the 4th international symposium on large spatial databases (SSD'95) held in Portland, Maine, August 6-9, 1995*, edited by M. J. Egenhofer and J. R. Herring, 47–66. Springer.

Koperski, K., J. Han, and J. Adhikary. 1998. Mining knowledge in geographical data. ftp://ftp.fas.sfu.ca/pub/cs/han/kdd/geo_survey98.ps (Accessed 12 Jul 2000).

Koperski, K., J. Han, and N. Stefanovic. 1998. An efficient two-step method for classification of spatial data. In *Proceedings of the 8th international symposium on spatial data handling (SDH'98) held in Vancouver, July 12-15, 1998*, 45–54.

Kuok, C. M., A. W.-C. Fu, and M. H. Wong. 1998. Mining fuzzy association rules in databases. *SIGMOD Record* 27(1):41–46.

Luo, J., and S. M. Bridges. 2000. Mining fuzzy association rules and frequency episodes for intrusion detection. *International Journal of Intelligent Systems* 15(8):687–703.

Miyajima, K., and A. Ralescu. 1994. Spatial organization in 2d segmented images: Representation and recognition of primitive spatial relations. *Fuzzy Sets and Systems* 65:225–236.

Ng, R. T., L. V. S. Lakshmanan, J. Han, and A. Pang. 1998. Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the 1998 ACM SIGMOD international conference on management of data held in Seattle, Washington, June 2-4, 1998*, edited by L. M. Haas and A. Tiwary, 13–24. ACM Press.

Ordonez, C., and E. Omiecinski. 1999. Discovering association rules based on image content. In *Proceedings of the 1999 IEEE forum on research and technology advances in digital libraries held in Baltimore, Maryland, May 19-21, 1999*, 38–49. IEEE.

Ramaswamy, S., S. Mahajan, and A. Silberschatz. 1998. On the discovery of interesting patterns in association rules. In *Proceedings of 24th international conference on very large data bases (VLDB'98) held in New York, August 24-*

27, 1998, edited by A. Gupta, O. Shmueli, and J. Widom, 368–379. Morgan Kaufmann.

Schmucker, K. J. 1984. *Fuzzy sets, natural language computations, and risk analysis*. Rockville, MD: Computer Science Press.

Silberschatz, A., and A. Tuzhilin. 1996. User-assisted knowledge discovery: How much should the user be involved. In *Proceedings of the 1996 ACM SIGMOD workshop on research issues on data mining and knowledge discovery (DMKD'96) held in Montréal, June 2, 1996*, 89–94. IRIS/Precarn.

Smith, G. B. 1999. User constraints in data mining. Mississippi State University Computer Science Dept.

Srikant, R., and R. Agrawal. 1995. Mining generalized association rules. In *Proceedings of the 21st international conference on very large databases (VLDB'95) held in Zurich, Switzerland, September 11-15, 1995*, edited by U. Dayal, P. M. D. Gray, and S. Nishio, 407–419. Morgan Kaufmann.

Srikant, R., and R. Agrawal. 1996. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD international conference on management of data held in Montréal, June 4-6, 1996*, edited by H. V. Jagadish and I. S. Mumick, 1–12. ACM Press.

Wooley, B. 2000. *mpishell documentation*. http://www.cs.msstate.edu/~bwooley/software/mpiShellDoc.html (Accessed 02 May 2001).

Yen, J., and R. Langari. 1998. *Fuzzy logic: Intelligence, control, and information*. Upper Saddle River, NJ: Prentice Hall.

Zadeh, L. A. 1965. Fuzzy sets. *Information and Control* 8(3):338–353.

Zadeh, L. A., K.-S. Fu, K. Tanaka, and M. Shimura, eds. 1975. *Fuzzy sets and their applications to cognitive and decision processes*. New York: Academic Press, Inc.

Zimmerman, H.-J. 1996. *Fuzzy set theory - and its applications*, 3rd ed. Boston: Kluwer Academic Publishers.

APPENDIX A

TABLES

Table A.1: Obj-Obj Relations

α	RO#	RC#	OO#	OC#	necess	avg	poss
0.00000	7	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.1362	0.3228
0.00000	7	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.0630	0.1382
0.00000	7	$\mathcal{H}$	10	$\mathcal{G}$	0.6887	0.7590	0.8233
0.00000	7	$\mathcal{H}$	11	$\mathcal{G}$	0.9959	0.9999	1.0000
0.00000	7	$\mathcal{H}$	12	$\mathcal{F}$	0.4881	0.5474	0.6058
1.57080	7	$\mathcal{H}$	8	$\mathcal{H}$	1.0000	1.0000	1.0000
1.57080	7	$\mathcal{H}$	9	$\mathcal{H}$	1.0000	1.0000	1.0000
1.57080	7	$\mathcal{H}$	10	$\mathcal{G}$	0.3113	0.3867	0.4654
1.57080	7	$\mathcal{H}$	11	$\mathcal{G}$	0.0041	0.0714	0.1524
1.57080	7	$\mathcal{H}$	12	$\mathcal{F}$	0.5119	0.5709	0.6287
3.14159	7	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.1362	0.3228
3.14159	7	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.0630	0.1382
3.14159	7	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	7	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	7	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
4.71239	7	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	7	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	7	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
4.71239	7	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0627	0.1429
4.71239	7	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
0.00000	8	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.1362	0.3228
0.00000	8	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.1401	0.3331

Table A.1: Obj-Obj Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
0.00000	8	$\mathcal{H}$	10	$\mathcal{G}$	0.9959	0.9999	1.0000
0.00000	8	$\mathcal{H}$	11	$\mathcal{G}$	0.6904	0.7603	0.8242
0.00000	8	$\mathcal{H}$	12	$\mathcal{F}$	0.7016	0.7712	0.8345
1.57080	8	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
1.57080	8	$\mathcal{H}$	9	$\mathcal{H}$	1.0000	1.0000	1.0000
1.57080	8	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0631	0.1440
1.57080	8	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
1.57080	8	$\mathcal{H}$	12	$\mathcal{F}$	0.2984	0.3733	0.4519
3.14159	8	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.1362	0.3228
3.14159	8	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.1401	0.3331
3.14159	8	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	8	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	8	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
4.71239	8	$\mathcal{H}$	7	$\mathcal{H}$	1.0000	1.0000	1.0000
4.71239	8	$\mathcal{H}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	8	$\mathcal{H}$	10	$\mathcal{G}$	0.0041	0.0719	0.1536
4.71239	8	$\mathcal{H}$	11	$\mathcal{G}$	0.3096	0.3846	0.4629
4.71239	8	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
0.00000	9	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.0630	0.1382
0.00000	9	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.1401	0.3331
0.00000	9	$\mathcal{H}$	10	$\mathcal{G}$	0.6887	0.7590	0.8233
0.00000	9	$\mathcal{H}$	11	$\mathcal{G}$	0.4842	0.5434	0.6019

Table A.1: Obj-Obj Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
0.00000	9	$\mathcal{H}$	12	$\mathcal{F}$	0.9919	0.9996	1.0000
1.57080	9	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
1.57080	9	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
1.57080	9	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
1.57080	9	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
1.57080	9	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0581	0.1371
3.14159	9	$\mathcal{H}$	7	$\mathcal{H}$	0.0000	0.0630	0.1382
3.14159	9	$\mathcal{H}$	8	$\mathcal{H}$	0.0000	0.1401	0.3331
3.14159	9	$\mathcal{H}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	9	$\mathcal{H}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	9	$\mathcal{H}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
4.71239	9	$\mathcal{H}$	7	$\mathcal{H}$	1.0000	1.0000	1.0000
4.71239	9	$\mathcal{H}$	8	$\mathcal{H}$	1.0000	1.0000	1.0000
4.71239	9	$\mathcal{H}$	10	$\mathcal{G}$	0.3113	0.3867	0.4654
4.71239	9	$\mathcal{H}$	11	$\mathcal{G}$	0.5158	0.5748	0.6325
4.71239	9	$\mathcal{H}$	12	$\mathcal{F}$	0.0081	0.0753	0.1560
0.00000	10	$\mathcal{G}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	10	$\mathcal{G}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	10	$\mathcal{G}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	10	$\mathcal{G}$	11	$\mathcal{G}$	0.0078	0.1491	0.3422
0.00000	10	$\mathcal{G}$	12	$\mathcal{F}$	0.0157	0.1604	0.3567
1.57080	10	$\mathcal{G}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000

Table A.1: Obj-Obj Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
1.57080	10	$\mathcal{G}$	8	$\mathcal{H}$	0.0041	0.0719	0.1536
1.57080	10	$\mathcal{G}$	9	$\mathcal{H}$	0.3113	0.3867	0.4654
1.57080	10	$\mathcal{G}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000
1.57080	10	$\mathcal{G}$	12	$\mathcal{F}$	0.9843	0.9994	1.0000
3.14159	10	$\mathcal{G}$	7	$\mathcal{H}$	0.6887	0.7590	0.8233
3.14159	10	$\mathcal{G}$	8	$\mathcal{H}$	0.9959	0.9999	1.0000
3.14159	10	$\mathcal{G}$	9	$\mathcal{H}$	0.6887	0.7591	0.8233
3.14159	10	$\mathcal{G}$	11	$\mathcal{G}$	0.0000	0.1314	0.3239
3.14159	10	$\mathcal{G}$	12	$\mathcal{F}$	0.0000	0.1247	0.3196
4.71239	10	$\mathcal{G}$	7	$\mathcal{H}$	0.3113	0.3867	0.4654
4.71239	10	$\mathcal{G}$	8	$\mathcal{H}$	0.0000	0.0631	0.1440
4.71239	10	$\mathcal{G}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	10	$\mathcal{G}$	11	$\mathcal{G}$	0.9922	0.9998	1.0000
4.71239	10	$\mathcal{G}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
0.00000	11	$\mathcal{G}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	11	$\mathcal{G}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	11	$\mathcal{G}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	11	$\mathcal{G}$	10	$\mathcal{G}$	0.0000	0.1314	0.3239
0.00000	11	$\mathcal{G}$	12	$\mathcal{F}$	0.0039	0.0686	0.1458
1.57080	11	$\mathcal{G}$	7	$\mathcal{H}$	0.0000	0.0627	0.1429
1.57080	11	$\mathcal{G}$	8	$\mathcal{H}$	0.3096	0.3846	0.4629
1.57080	11	$\mathcal{G}$	9	$\mathcal{H}$	0.5158	0.5748	0.6325

Table A.1: Obj-Obj Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
1.57080	11	$\mathcal{G}$	10	$\mathcal{G}$	0.9922	0.9998	1.0000
1.57080	11	$\mathcal{G}$	12	$\mathcal{F}$	0.9961	0.9999	1.0000
3.14159	11	$\mathcal{G}$	7	$\mathcal{H}$	0.9959	0.9999	1.0000
3.14159	11	$\mathcal{G}$	8	$\mathcal{H}$	0.6904	0.7603	0.8242
3.14159	11	$\mathcal{G}$	9	$\mathcal{H}$	0.4842	0.5434	0.6019
3.14159	11	$\mathcal{G}$	10	$\mathcal{G}$	0.0078	0.1491	0.3422
3.14159	11	$\mathcal{G}$	12	$\mathcal{F}$	0.0000	0.0602	0.1367
4.71239	11	$\mathcal{G}$	7	$\mathcal{H}$	0.0041	0.0714	0.1524
4.71239	11	$\mathcal{G}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	11	$\mathcal{G}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
4.71239	11	$\mathcal{G}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
4.71239	11	$\mathcal{G}$	12	$\mathcal{F}$	0.0000	0.0000	0.0000
0.00000	12	$\mathcal{F}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	12	$\mathcal{F}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	12	$\mathcal{F}$	9	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	12	$\mathcal{F}$	10	$\mathcal{G}$	0.0000	0.1247	0.3196
0.00000	12	$\mathcal{F}$	11	$\mathcal{G}$	0.0000	0.0602	0.1367
1.57080	12	$\mathcal{F}$	7	$\mathcal{H}$	0.0000	0.0000	0.0000
1.57080	12	$\mathcal{F}$	8	$\mathcal{H}$	0.0000	0.0000	0.0000
1.57080	12	$\mathcal{F}$	9	$\mathcal{H}$	0.0081	0.0753	0.1560
1.57080	12	$\mathcal{F}$	10	$\mathcal{G}$	0.0000	0.0000	0.0000
1.57080	12	$\mathcal{F}$	11	$\mathcal{G}$	0.0000	0.0000	0.0000

Table A.1: Obj-Obj Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
3.14159	12	$\mathcal{F}$	7	$\mathcal{H}$	0.4881	0.5474	0.6058
3.14159	12	$\mathcal{F}$	8	$\mathcal{H}$	0.7016	0.7712	0.8345
3.14159	12	$\mathcal{F}$	9	$\mathcal{H}$	0.9919	0.9996	1.0000
3.14159	12	$\mathcal{F}$	10	$\mathcal{G}$	0.0157	0.1604	0.3567
3.14159	12	$\mathcal{F}$	11	$\mathcal{G}$	0.0039	0.0686	0.1458
4.71239	12	$\mathcal{F}$	7	$\mathcal{H}$	0.5119	0.5709	0.6287
4.71239	12	$\mathcal{F}$	8	$\mathcal{H}$	0.2984	0.3733	0.4519
4.71239	12	$\mathcal{F}$	9	$\mathcal{H}$	0.0000	0.0581	0.1371
4.71239	12	$\mathcal{F}$	10	$\mathcal{G}$	0.9843	0.9994	1.0000
4.71239	12	$\mathcal{F}$	11	$\mathcal{G}$	0.9961	0.9999	1.0000

Table A.2: Class-Class Relations

α	RO#	RC#	OO#	OC#	necess	avg	poss
0.00000	13	$\mathcal{H}$	15	$\mathcal{F}$	0.9919	0.9996	1.0000
0.00000	13	$\mathcal{H}$	14	$\mathcal{G}$	0.9959	0.9999	1.0000
1.57080	13	$\mathcal{H}$	15	$\mathcal{F}$	0.5119	0.5709	0.6287
1.57080	13	$\mathcal{H}$	14	$\mathcal{G}$	0.0041	0.2290	0.4654
3.14159	13	$\mathcal{H}$	15	$\mathcal{F}$	0.0000	0.0000	0.0000
3.14159	13	$\mathcal{H}$	14	$\mathcal{G}$	0.0000	0.0000	0.0000
4.71239	13	$\mathcal{H}$	15	$\mathcal{F}$	0.0081	0.0753	0.1560
4.71239	13	$\mathcal{H}$	14	$\mathcal{G}$	0.3113	0.4807	0.6325
0.00000	15	$\mathcal{F}$	13	$\mathcal{H}$	0.0000	0.0000	0.0000

Table A.2: Class-Class Relations *cont.*

α	RO#	RC#	OO#	OC#	necess	avg	poss
0.00000	15	$\mathcal{F}$	14	$\mathcal{G}$	0.0000	0.0924	0.3196
1.57080	15	$\mathcal{F}$	13	$\mathcal{H}$	0.0000	0.0251	0.1560
1.57080	15	$\mathcal{F}$	14	$\mathcal{G}$	0.0000	0.0000	0.0000
3.14159	15	$\mathcal{F}$	13	$\mathcal{H}$	0.4881	0.7727	1.0000
3.14159	15	$\mathcal{F}$	14	$\mathcal{G}$	0.0039	0.1145	0.3567
4.71239	15	$\mathcal{F}$	13	$\mathcal{H}$	0.0000	0.3341	0.6287
4.71239	15	$\mathcal{F}$	14	$\mathcal{G}$	0.9843	0.9996	1.0000
0.00000	14	$\mathcal{G}$	13	$\mathcal{H}$	0.0000	0.0000	0.0000
0.00000	14	$\mathcal{G}$	15	$\mathcal{F}$	0.0157	0.1604	0.3567
1.57080	14	$\mathcal{G}$	13	$\mathcal{H}$	0.0000	0.3407	0.6325
1.57080	14	$\mathcal{G}$	15	$\mathcal{F}$	0.9961	0.9999	1.0000
3.14159	14	$\mathcal{G}$	13	$\mathcal{H}$	0.6887	0.9196	1.0000
3.14159	14	$\mathcal{G}$	15	$\mathcal{F}$	0.0000	0.1248	0.3196
4.71239	14	$\mathcal{G}$	13	$\mathcal{H}$	0.0000	0.1499	0.4654
4.71239	14	$\mathcal{G}$	15	$\mathcal{F}$	0.0000	0.0000	0.0000