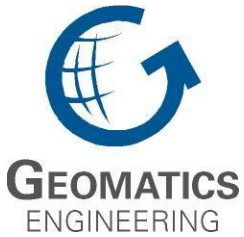

UCGE Reports
Number 20355

Department of Geomatics Engineering

Reservoir Characterization and Horizontal Well Placement Guidance Acquisition by Using GIS and Data Mining Methods

(URL: http://www.geomatics.ucalgary.ca/graduatetheses)

by

Baijie Wang

June, 2012

UNIVERSITY OF CALGARY

Reservoir Characterization and Horizontal Well Placement Guidance Acquisition by Using GIS and Data Mining Methods

by

Baijie Wang

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

June 2012

Abstract

This thesis investigates to develop and apply geographic information system (GIS) and data mining methods for reservoir characterization and horizontal well placement guidance acquisition. Reservoir characterization is a process of quantitatively assigning reservoir and fluid properties while recognizing geologic uncertainties in spatial variability. To identify reservoir properties with spatial correlation, a new density-based spatial clustering method, SEClu, is presented to group core analysis data. Further, a novel fuzzy ranking artificial neural network (FR-Neural) framework is introduced for accurately characterizing reservoir properties from well log data. SEClu and the FR-Neural framework are evaluated with synthetic and real datasets.

Horizontal well placement guidance acquisition (HWPGA) analyzes the real field data and collects guidelines for placing horizontal wells into a reservoir. In this thesis, a group of horizontal well placement attributes are defined to capture the location of horizontal wells in a heterogeneous reservoir. A customized association rule mining method, named SE-Apriori, is introduced to analyze the influences of the horizontal well placement attributes on the oil production. The SE-Apriori considers two predefined constraints from the HWPGA problem and, thus, can generate fewer association rules with less execution time. A GIS prototype containing the SE-Apriori tool was developed to help in efficiently managing petroleum field data and visualizing the association rule mining results on a map. Finally, the proposed SE-Apriori method is evaluated using a real dataset from a steam assisted gravity drainage (SAGD) project in Alberta, Canada.

Acknowledgements

I want to sincerely thank my supervisor, Dr. Xin Wang, and co-supervisor, Dr. Zhangxing (John) Chen, for their consistent support and guidance during this research. I would also like to give thanks to Dr. Michael Richter for his thoughtful advice on machine learning. This research would not have been possible without the industrial data from Husky Energy and the assistance from several individuals there. Thanks to Mr. KC Yeung and Ms. Susan Johnson for the input and sharing of their expertise in the oil and gas industry. Also, acknowledgement must be given to Mitacs, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada School of Energy and Environment for providing the research funding.

I am very grateful for my parents, Cuirong Zhang and Hanwen Wang, my sister Jie Wang, and my wife, Yabo Li. Thanks for their unfailing love, encouragement and patience.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
ANN	Artificial Neural Network
API	Application Programming Interface
ARM	Association Rule Mining
BITW8	80% Bitumen Weight
CLARANS	A Method for Clustering Objects for Spatial Data Mining
DBSCAN	A Density-based Spatial Clustering of Applications with Noise
DBRS	A Density-based Spatial clustering Method with Random Sampling
DSR	Density-spEntropy Reachable
FC	Fuzzy Curve
FR	Fuzzy Ranking
FS	Fuzzy Surface
GIS	Geographic Information System
HWPGA	Horizontal Well Placement Guidance Acquisition
ICP	Intermedium Casing Point
MLP	Multilayer Perceptron
OWC	Oil Water Contact
RC	Reservoir Characterization
RT	Reservoir Top
So50	50% Oil Saturation
SEClu	A Spatial Entropy-based Spatial Clustering Method

Chapter One: **Introduction**

1.1 Background

Professionals in the petroleum industry are facing the dilemma of making complex high-stake decisions while lacking efficient methods to manage and marshal overloaded field data (Mohaghegh, 2005). With advanced sensors installed into thousands of wells, very large amounts of field data that carry important information have been accumulated in the petroleum industry. The challenge is often how to efficiently interpret the data and benefit the decision makers by providing valuable information. However, interpretation of large volume of data through traditional analytical methods is often unsuccessful, incomplete and inadequate (Zangl and Hannerer, 2003). Hence methods for extracting important information concealed in extensive datasets are required.

Data mining is the process of discovering interesting, implicit and previously unknown knowledge from large databases (Frawley et al., 1992). It is an interdisciplinary field at the intersection of artificial intelligence, machine learning, statistics and database systems (Chakrabarti et al., 2006). Data mining has been successfully utilized in various petroleum applications, including reservoir characterization (Mohaghegh et al. 1996; Aminian and Ameri, 2005), fracture detection (El Ouahed et al., 2005), seismic analysis (Strecker and Uden, 2010; Marroquin et al., 2009) and reservoir modeling (Aulia et al., 2010; Zangl and Hannerer, 2003). At present, with the large amount of collected field data in the petroleum industry, data mining has great potential to provide better solutions

to explain the complexity of different phenomena affecting oil production and exploration.

Geographic Information System (GIS) has also shown great potential in managing the field data in the petroleum industry. GIS is a computer system for capturing, storing, querying, analyzing and displaying geospatial data (Chang, 2012). It helps with the storage of field data in a geodatabase and the visualization and management of the data geographically as part of a map. The large volume of field data in the petroleum industry has been geographically referenced to spatial locations, integrating data mining and GIS and offering a great opportunity for the provision of valuable information for decision makers.

1.2 Research Gap and Problem Statement

This thesis focuses on the development of GIS and data mining methods for two petroleum applications, i.e. reservoir characterization and horizontal well placement guidance acquisition.

1.2.1 Reservoir Characterization

Reservoir characterization (RC) is a process for quantitatively assigning reservoir and fluid properties while recognizing geologic uncertainties in spatial variability (Mohaghegh et al. 1996). The most direct field data used in reservoir characterization is from core analysis, where rock core samples taken from a reservoir are analyzed in a laboratory. Core analysis data are accurate and are, thus, widely used as benchmark or validation data in reservoir characterization.

Acquisition of rock core samples from a reservoir is, however, costly; and, core analysis data can only provide reservoir properties for several discrete points. For areas without core samples, the reservoir properties are calculated by building correlations from nearby core analysis data. The assumptions of this process are that spatial correlation exists in reservoir properties and that reservoir properties in an area are similar to nearby core analysis data. Therefore, clustering core analysis data considering the spatial correlation is helpful in selecting representative core analysis data for areas where core samples are not available. However, most existing spatial clustering methods in data mining consider spatial and nonspatial attributes independently and none of them can be applied to clustering core analysis data.

In order to acquire large-scale reservoir properties, artificial neural networks (ANNs) have recently been introduced into reservoir characterization by using core analysis and well log data (Al-Bulushi et al., 2009; Aminian and Ameri, 2005; Mohaghegh et al., 1996; Mohaghegh et al., 2000; Wong et al., 1995). Well log data record high resolution but indirect reservoir information via subterraneous sensors (Lim, 2005). In this method, well log and core analysis data with the same depth interval are paired together to train the ANN.

After proper training, the ANN can be used to predict reservoir properties based only on well log data for depth intervals where core analysis data are not available. The assumption of applying ANN to reservoir characterization is that relationships between well logs and reservoir properties can be correctly represented by training data; hence, representative training data selection is critical for the ANN's performance. However, the research on selecting representative well log data for ANN-based reservoir

characterization is limited; therefore, the modeling ability of ANN in reservoir characterization can be jeopardized.

1.2.2 Horizontal Well Placement Guidance Acquisition

Enhanced oil recovery technologies, such as steam assisted gravity drainage (SAGD), require superior horizontal well placement to achieve good oil production performance (Chen et al., 1997). Several horizontal well planning methods, with the assistance of numerical simulation, have been introduced (Chen et al., 1997; Shin and Polikar, 2007). However, these methods derived the horizontal well placement plans based only on simulated or predicted data. Without investigating real field data, the generated well placement plans have high uncertainty.

Large amounts of real SAGD field data, including geological, horizontal drilling and production data, have been consistently collected in the past decade. The data contains implicit information conveying the correlations between horizontal well placement and oil production performance. Retrieving this information will significantly help horizontal well placement planning. However, no work has investigated real SAGD field data and the retrieval of the horizontal well placement guidance.

1.2.3 GIS with Data Mining in Field Data Management

Field data from thousands of wells in the petroleum industry have accumulated for decades. A growing number of oil and gas companies have implemented GIS to efficiently manage the data (Coburn and Yarus, 2000). Data mining methods have also been introduced to marshal the overload of data (Zangl and Hannerer, 2003).

Most of the field data and data mining results are, however, in a numerical or text format. Without proper visualization, it is very difficult for engineers to work with the data and understand the data mining results. With a large volume of field data being recently geographically referenced to spatial locations, integrating data mining methods into GIS shows great potential in the petroleum industry and is becoming a new research topic. The problem is often the provision of a user-friendly system for efficiently managing the large volume data and visualizing the data mining results.

1.3 Research Objectives

The literature review and discussions in Section 1.2 lead to the following objectives:

1. Develop a spatial clustering algorithm that accounts for spatial correlation,
2. Incorporate an automatic well log data selection step into reservoir characterization and develop a new ANN-based reservoir characterization framework,
3. Develop an efficient data mining method to identify horizontal well placement guidance from real SAGD field data, and
4. Integrate data mining tools into GIS and develop a system prototype that can manage, analyze and visualize the petroleum field data.

1.4 Research Contribution

The main contributions of this thesis can be summarized as:

1. This thesis proposes a new density-based spatial clustering method (SEClu), which clusters the core analysis data considering the local non-spatial similarity

and spatial correlation. The new method is evaluated using both synthetic and real datasets.

2. A novel fuzzy ranking-artificial neural network (FR-Neural) framework is introduced for reservoir characterization from well logs. In the new framework, fuzzy ranking selects the representative well log variables for neural inputs with the objective of specific reservoir property characterization, which implicitly increases the modeling ability of the ANN in reservoir characterization. The new method is evaluated using real data from three wells in Alberta, Canada.
3. This thesis formalizes the horizontal well placement guidance acquisition (HWPGA) problem and presents a customized association rule mining method to solve it. In order to characterize the location of horizontal wells in a heterogeneous reservoir, 40 horizontal well placement attributes are defined. A customized association rule mining method, named SE-Apriori, is introduced for analysis of the interesting horizontal well placement patterns from real SAGD field data. The proposed method is evaluated using a real dataset from a SAGD project in Alberta, Canada.
4. A system prototype, named PetroData-GIS, which incorporates the SE-Apriori tool into a GIS, is developed to efficiently manage the large volume of field data in the petroleum industry and visualize the association rule mining results.

1.5 Thesis Outline

Chapter Two gives a literature review of density-based spatial clustering methods, using ANN in reservoir characterization, horizontal well planning and association rule mining

methods. Chapter Three starts by introducing the SEClu spatial clustering algorithm. The FR-Neural framework is proposed for reservoir characterization from well logs. Finally, the SEClu method and FR-Neural reservoir characterization framework are evaluated through synthetic and real datasets. Chapter Four formalizes the HWPGA problem and introduces a customized association rule mining method to solve it. A group of horizontal well placement attributes are defined and the SE-Apriori association rule mining method is introduced. The development of PetroData-GIS system containing the SE-Apriori tool is described. Finally, the SE-Apriori method is evaluated through a real SAGD dataset. Chapter Five draws conclusions and states future works of this thesis.

Chapter Two: **Related Work**

This chapter presents the literature study in the following areas: First, previous works on density-based spatial clustering are reviewed. Second, research works that use an artificial neural network in reservoir characterization are presented. Third, reviews of horizontal well placement planning and association rule mining are given.

2.1 Density-Based Spatial Clustering

In order to discover arbitrarily shaped clusters in core analysis data, density-based spatial clustering methods are used. Several density-based spatial clustering methods have been proposed that consider the spatial and non-spatial attributes in the data.

2.1.1 DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) was the first proposed density-based spatial clustering method. In order to form a new cluster or extend an existing cluster in DBSCAN, the density around a point *p* must surpass the predefined values for which a neighbourhood around *p* with radius *Eps* must contain at least a minimum number of points (*MinPts*). The greatest advantages of DBSCAN are that it can find arbitrarily shaped clusters and it requires only a distance function and two input parameters (Wang and Hamilton, 2003).

Given a dataset *D*, a distance function *dist* and parameters *Eps* and *MinPts*, the following definitions are used to define DBSCAN (Ester et al, 1996).

- **Definition 2.1 (Eps-neighbourhood)**: The *Eps*-neighbourhood of point *p*, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p)=\{q \in D \mid dist(p,q) \leq Eps\}$.

- **Definition 2.2 (directly density-reachable)**: A point p is directly density-reachable from a point q w.r.t *Eps* and *MinPts* if (1) $q \in N_{Eps}(p)$ and (2) $N_{Eps}(p) \geq$ *MinPts.*
- **Definition 2.3 (density-reachable)**: Point p is density-reachable from a point q w.r.t *Eps* and *MinPts* if there is a chain of points $p_1,\ldots,p_n$, $p_1=q$, $p_n=p$ such that p_{i+1} is directly density-reachable from p_i.
- **Definition 2.4 (density-connected)**: A spatial point p is density-connected to a point q w.r.t *Eps* and *MinPts* if there is a point o such that p and q are density-reachable from o.
- **Definition 2.5 (density-based cluster)**: A density-based cluster C w.r.t *Eps* and *MinPts* is a non-empty subset of D satisfying the following conditions: (1) $\forall p, q \in D$, if $p \in C$ and q is density-reachable from p; (2) $\forall p, q \in C$, p is density-connected to q.

Once *Eps* and *MinPts* are defined, DBSCAN starts to group the data from an arbitrary point, q. It begins by performing a neighbourhood query, which finds the neighbourhood of point q. If the neighbourhood is sparsely populated, i.e. contains fewer than *MinPts* points, q is labeled as noise; otherwise, a cluster is created, and all points in q's neighbourhood are placed in this cluster. The neighbourhood of each of q's neighbours is then examined to see if it can be added to the cluster. If so, the process is repeated for every point in this neighbourhood, and so on. If a cluster cannot be expanded further, DBSCAN chooses another arbitrary unlabeled point and repeats the process until all the points in D have been assigned into a cluster or identified as noise.

Although DBSCAN gives extremely good results and is efficient in many datasets, it is not suitable for cases where the non-spatial attributes play a role in the determination of the desired clusters, since it does not take into consideration the non-spatial attributes in the dataset (Ester et al., 1996; Wang and Hamilton, 2003).

2.1.2 Spatial Clustering Considering Non-spatial Attributes

Very few spatial clustering algorithms have been proposed for dealing with both spatial and non-spatial attributes. An option is the handling of non-spatial and spatial attributes in two separate steps, as described in CLARANS (Ng and Han, 2002). The other option is to deal with the non-spatial attributes and spatial attributes together in the clustering process. The similarity functions for non-spatial attributes and the distance functions for spatial attributes are handled simultaneously, in order to define the overall similarity between objects. Algorithms that have taken this approach include generalized DBSCAN (GDBSCAN) (Sander et al., 1998) DBRS (Wang and Hamilton, 2003) and clustering of multi-represented objects (Kailing et al., 2004).

GDBSCAN (Sander et al., 1998) takes into account the non-spatial attributes of an object as a weight attribute, which is defined by the weighted cardinality of the singleton containing the object. The weight can be the size of the area of the clustering object or a calculated value from several non-spatial attributes.

DBRS (Wang and Hamilton, 2003) introduces the concept of purity to determine the categorical attributes of objects in the neighbourhood. Purity is defined as the percentage of objects in the neighbourhood, with the same characteristic for a particular non-spatial attribute as the centre object. For non-spatial attributes, this can avoid the

creation of clusters of points with different values, even though these points may be close to each other. However, purity is only defined for categorical non-spatial attributes.

Clustering of multi-represented objects (Kailing et al., 2004) extends DBSCAN by retrieving information from one attribute to multiple attributes. Within a set of attributes, either spatial or non-spatial, density reachability is defined as the union or intersection of the selected attributes. For example, the union form requires that the summation of the neighbourhood in every attribute space be larger than a particular threshold.

Limited studies have been conducted on applying spatial clustering methods on geological studies. Tutmez and Tercan (2007) utilized spatial clustering methods to identify reservoir heterogeneity and geological uncertainty. In their experiments, 27 wells are grouped into 4 clusters based on their porosity values. As a post processing step, for each identified cluster the spatial correlation was evaluated by using the semivariogram plot. The authors concluded that there was a close relationship between uncertainty and spatial variability. However, in (Tutmez and Tercan, 2007) the spatial and nonspatial attributes in the geological data were treated independently in the spatial clustering process. In geological data where strong spatial correlation exists, the clustering result may lose credibility.

2.2 Reservoir Characterization

For acquiring large-scale reservoir properties, previous methods have used regression analysis to build linear or nonlinear correlations between well log data and various reservoir properties. For less heterogeneous reservoir, where the reservoir properties vary

not very much across reservoir space, these correlations work fairly well. However, as the degree of heterogeneity of the reservoir increases, these correlations lose accuracy (Aminian and Ameri, 2005; Helle et al., 2001; Lim, 2005; Mohaghegh et al., 1996).

2.2.1 ANN in Reservoir Characterization

Attempts to use artificial neural networks (ANNs) in reservoir characterization began in the mid 1990s, when multilayer perceptron (MLP) neural networks with back propagation (BP) algorithms were gradually accepted as new intelligent reservoir characterization tools (Aminian and Ameri, 2005; Helle, et al., 2001; Wong et al., 1995). Mohaghegh et al. (1996) built a three-layer MLP neural network for porosity and permeability characterization. The same neural inputs, including three well log variables (gamma ray, bulk density and depth induction), were used for both porosity and permeability characterizations. Results showed that ANNs had great potential, since they exceeded traditional statistical methods for accurately predicating reservoir properties for heterogeneous reservoirs. However, their work did not discuss how the three well log variables were selected as the neural inputs.

Independently, Helle et al. (Helle H.B. et al. 2001) concluded that using ANN in reservoir characterization has a number of advantages over conventional methods, despite requiring efforts to select good representative training data. In this work, Helle et al. built two MLP neural networks to predict porosity and permeability using two different sets of well log variables as inputs. Well log variables from sonic, density and resistivity categories were used for neural inputs for porosity characterization; well log variables from density, gamma ray, neutron and sonic categories were used for permeability characterization. All the neural inputs were manually and carefully selected. As stated by

the authors, although the final predicted porosity and permeability values via MLP were accurate enough to meet most practical needs, the main drawback of using ANN in reservoir characterization was that it is difficult to select a representative collection of training samples.

2.2.2 Data Selection for ANN in Reservoir Characterization

As a data driven approach, the modeling ability of an ANN relies heavily on the quality of its neural inputs, and any inappropriate well log data selection decreases ANN's performance in reservoir characterization.

Several studies have shown that fuzzy analysis can help identify the optimal set of independent variables for an ANN by addressing the uncertainties of neural inputs (Mohaghegh, 2005, 2000). Lim (2005) used a fuzzy curve analysis to select neural inputs from well logs for use in ANN-based reservoir characterization. In this work, 5 and 6 out of 8 candidate well log variables were selected via the fuzzy curve for permeability and porosity characterizations, respectively. Based on the selected neural inputs, MLP neural networks were built, and their modeling ability was compared with that of multiple regression analysis. Results showed that the modeling ability via the combination of fuzzy curve and MLP significantly exceeded that of multiple regression analysis. However, the well log data redundancy problem was not discussed. Additionally, Lim (2005) did not give a comprehensive evaluation on the proposed method, because the proposed ANN model was trained and tested using the same dataset.

2.3 Horizontal Well Placement Planning

Given the importance of horizontal well placement, both static and dynamic methods have been proposed for well placement planning.

2.3.1 Static and Dynamic Horizontal Well Placement Planning

Static horizontal well placement methods determine the horizontal well placement plan with only a geological model while they do not account for the dynamics of fluid flow in the reservoir (Norrena and Deutsch, 2002). The static methods deliver the well placement plan by maximizing a predefined objective function.

McLennan et al. (2006) applied an exhaustive calculation scheme to determine the optimal elevation of a pair of SAGD wells, where the oil recovery factor was maximized. In this work, a SAGD well pair was placed in between two geological surfaces, top and bottom continuous bitumen surfaces. After scanning all possible plans, the optimal well placement plan was determined by maximizing the oil recovery factor. Although this method is straightforward, the assumption that the oil recovery is only related to the well elevation without consideration of other geological complexities is unrealistic. Moreover, the computational cost of the exhaustive calculation is very high, which is not suitable for solving multiple horizontal well placement problems.

Norrena and Deutsch (2002) introduced a simulated annealing method to optimize the well placement subject to a predefined objective function. They argued that simulated annealing was computationally efficient and capable of delivering the globally optimized result. Four different objective functions related to geological and economical constraints in well placement were discussed. This method was tested on the optimization of the

elevation of a SAGD well pair, where the connected pore volume fraction above the producer was maximized. However, as stated by the authors, the optimization result cannot be directly applied to the field, since the uncertainty of the results is still high.

Dynamic horizontal well placement planning refers to the application of a numerical simulator in assisting the determination of well plans by considering both the geological model and fluid flow (Norrena and Deutsch, 2002).

Chen et al. (1997) examined different SAGD well placement plans in a reservoir with a overlying gas cap and an underlying aquifer. The producer was fixed at the bottom of the reservoir, and the injector was systematically moved from the top gas zone toward the producer in a homogeneous model. Simulation results suggested that the SAGD oil recovery was drastically reduced for situations when the injector was placed lower than the approximate midpoint of the oil pay section. The simulation model used in the research assumed a homogeneous reservoir; thus, its results lose credibility when the reservoir heterogeneity increases. In addition, although a homogeneous model was used, each simulation took 12 hours, leading to a very high computation cost.

From a two-dimensional (2D) simulation, Shin and Polikar (2007) found that increasing the distance between the producer and the injector within a SAGD pair increased the thermal recovery efficiency. However, Albanhlani and Babadagli (2008) argued that Shin and Polikar's work did not provide a fair evaluation, since the time-varied thermal efficiency with different producer/injector spacing was not compared. Moreover, a 2D simulation model was used to represent a real three-dimensional (3D) reservoir to cut down on the simulation computation; however, this jeopardizes the results reliability due to over simplification.

Both static and dynamic horizontal well placement planning methods rely on accurate geological data. Static methods do not consider the fluid flow, and the reliability of the results is dependent on the objective function definitions. As for the dynamic methods, the application of numerical simulation to determine the optimal well placement plan is impractical due to the computational requirement (Norrena and Deutsch, 2002). In practice, these methods can be used to deliver the best possible horizontal well placement plans, and the final well placement plan is subject to the engineers' experience.

2.4 Association Rule Mining and The Apriori Algorithm

Association rule mining is one of the well-developed data mining methods for discovering interesting correlations between variables in large datasets. For the first time, it is introduced in this thesis to analyze the correlations between horizontal well placement attributes and oil production performance from real historical field data.

2.4.1 Association Rule Mining

Association rule mining (ARM) was first introduced by Agrawal and Srikant (1993) to analyze the transactional database and derive association rules (Han and Kamber, 2006; Wu X., 2007). A typical example is the market basket analysis. This process analyzes customer consumption patterns by finding associations between different items which customers purchase. For instance, if the customer buys milk, how likely does he/she buy bread at the same time? Such information can be helpful in improving marketing activities, such as shelf space placement. Further, ARM is applied to many applications including marketing (Sohn and Kim, 2008; Jiao and Zhang, 2005), bioinformatics (Creighton and Hanash, 2003) and reservoir analysis (Aulia et al., 2010).

With the definition of the association rule in Agrawal et al. (1993), let *D* be the set of all items, and *X* and *Y* be two subsets of *D* such that $X, Y \subset D$. An association rule with respect to *X* and *Y* could be in the form of:

$$X \Rightarrow Y, \text{ such that } X, Y \subset D, \; X, Y \neq \varnothing \text{ and } X \bigcap Y \neq \varnothing \tag{2.1}$$

where *X* is called the *antecedent* and *Y* is the *consequence*.

Two concepts are essential in defining the interestingness of an association rule (Han J. and Kamber M. 2006), support and confidence. The support of rule $X \Rightarrow Y$ is defined to be the percentage of transactions that consist of $X \bigcup Y$ to the total number of transactions, as shown in Eq. (2.2).

$$support \; (X \Rightarrow Y) = P(X \bigcup Y) \tag{2.2}$$

The confidence of rule $X \Rightarrow Y$ is the percentage of transactions that consist of *X* and *Y* to the number of transactions that only contain *X* (Han J. and Kamber M. 2006). The definition is in the form of conditional probability shown in Eq. (2.3).

$$confidence \; (X \Rightarrow Y) = P(Y \mid X) = \frac{P(X \bigcup Y)}{P(X)} \tag{2.3}$$

Rules that are satisfied with large support and confidence values are considered to be *interesting*. The objective of ARM is the generalization of all interesting rules from the transaction database satisfying both a minimum support threshold (*minsup*) and a minimum confidence threshold (*minconf*).

In general, the process of ARM can be divided into two steps:

1. Find all the frequent itemsets. Frequent itemsets are those itemsets that satisfy the *minsup* threshold.

2. Generate the desired association rules. This step generates all strong rules from the frequent itemsets that satisfy the *minconf* threshold.

In the next section, a widely used association rule mining algorithm is introduced.

2.4.2 The Apriori Algorithm

Apriori is a classic ARM algorithm. It was proposed by Agralwal and Srikant (1994) for mining frequent itemsets and associations for a transactional dataset. Apriori is a seminal algorithm, and it applies a level-wise search mechanism to find all the frequent itemsets. It starts by identifying the frequent 1-itemset by scanning the dataset and counting the support of each item. Next, the frequent 1-itemsets are used to find the frequent 2-itemsets, which are used to find frequent 3-itemsets. This process continues until no frequent itemsets can be found. The search for the itemsets of each frequent level requires a full scan of the dataset. To improve the efficiency of the level-wise frequent itemset search, an important property is introduced to reduce the searching space:

- **Definition 2.6 (Apriori property)**: All nonempty subsets of a frequent itemset must also be frequent.

The Apriori property is based on the observation that a super itemset of a non-frequent itemset is still non-frequent. For example, assume itemset X is not frequent, $sup(X) < minsup$. If item Y is added to itemset X, then the resulting itemset, $X \bigcup Y$, cannot occur more frequently than X; thus, $X \bigcup Y$ is not frequent, either.

Based on the Apriori property, the Apriori algorithm is presented in Figure 2–1. Let k-itemset denote an itemset containing k items and F_k and C_k be the collections of frequent k-itemsets and candidate k-itemsets, respectively. The Apriori algorithm first

passes the dataset to count the occurrence of each item and determine 1-frequent itemsets, F_1. The subsequent passes contain two steps. In the first step, the frequent k-itemsets, F_k, found in the k-th pass are used to generate the candidate (k+1)-itemsets, C_{k+1}, using the Apriori-gen function, as shown in Figure 2–2. C_{k+1} is a superset of F_k and all the subsets of $c \in C_{k+1}$ are frequent. In the following step, the Apriori algorithm scans the data again to count the support of each candidate in $C_{k+1;}$ and, the ones with support of less than *minsup* are deleted. This process continues until F_k is empty.

Algorithm: Apriori (*D*, *minsup*)

Input: (1) The dataset *D* containing all the transaction records (2) *minsup*

Output: $\bigcup_k F_k$

01: Let F_1={frequent 1-itemsets};

02: **for (k=2;** $F_{k-1} \neq \varnothing$; k++)

03: C_k=Apriori-gen(F_{k-1});

04: Scan *D* to determine the support to each candidate $c \in C_k$

05: F_k={ $c \in C_k$ | c.support ≥ *minsup*} ;

06: **end**

07: **return** $\bigcup_k F_k$;

Figure 2–1 Pseudo code of Apriori algorithm (Wu et al., 2007)

The Apriori-gen function takes an argument of the frequent k-itemset F_k and returns a superset of the set of all candidate (k+1)-itemsets, C_{k+1}. To generate C_{k+1}, F_k is

joined with F_k itself. It is assumed that all items in an itemset are sorted in the lexicographic order. For two frequent k-itemsets f_1 and f_2 belonging to F_k, a candidate (k+1)-itemset is generated by merging them only if the first (k-1) items in f_1 and f_2 are the same and the last items are different, which is shown in line 03 in Figure 2–2. $f_1.item_k < f_2.item_k$ denotes that $f_2.item_k$ is placed in a later position in a lexicographic order than $f_1.item_k$, which ensures that no duplication is made. The resulting candidate (k+1)-itemsets by joining f_1 and f_2 is < $f_1.item_1$, $f_1.item_2$, …, $f_1.item_k$, $f_2.item_k$ >.

Function: Apriori-gen (F_k)

Input: Frequent k-itemsets: F_k

Output: Candidate k-itemsets: C_{k+1}

01: **foreach** itemset $f_1 \in F_k$

02: **foreach** itemset $f_2 \in F_k$

03: **if** ($f_1.item_1 = f_2.item_1$ **and** $f_1.item_2 = f_2.item_2$ **and**

$f_1.item_{k-1} = f_2.item_{k-1}$ **and** $f_1.item_k < f_2.item_k$) **then**

04: { $c = merge(f_1, f_2)$ **and** Add c into C_{k+1}; }

05: **end** //end if

06: **end** //end foreach

07: **end** // end foreach

08 **return** C_{k+1};

Figure 2–2 Pseudo code of Apriori generation function (Wu et al., 2007)

After finding all frequent itemsets, the remaining task is the generation of the desired association rules. To generate interesting rules, all nonempty subsets of every frequent itemset, *f*, are enumerated. For each subset of *f*, *a*=subset(*f*), a rule is generated with the form of *a*=>*f*-*a*, if its confidence is larger than *minconf* (Han and Kamber, 2006).

Chapter Three: **Using Spatial Clustering and Artificial Neural Network for Reservoir Characterization**

This chapter starts by presenting a new density-based spatial clustering method for grouping core analysis data considering the spatial correlation. Second, a new FR-Neural framework is proposed to characterize reservoir properties from well log data. Finally, the proposed methods are tested on synthetic and real datasets.

3.1 Introduction

In the petroleum industry, reservoir properties, such as porosity, permeability and saturation, have significant impacts on reservoir simulation, enhanced oil recovery design, field operations and geological studies (Aminian and Ameri, 2005). For example, porosity describes the volume fraction of the pore space and is related to the hydrocarbon reserves contained in a reservoir. Reservoir characterization is a process of quantitatively assigning reservoir and fluid properties, such as porosity, permeability and fluid saturation, while recognizing geologic uncertainties in spatial variability (Mohaghegh et al., 1996). This thesis focuses on characterizing reservoir properties. In practice, reservoir characterization is a very complex geological problem and being able to obtain reliable and accurate reservoir properties is crucial (Al-Bulushi et al., 2009; Lim, 2005; Mohaghegh, 2000).

Core analysis data provides accurate reservoir properties by analyzing core samples taken from a reservoir and, thus, is widely used as benchmark or validation data in reservoir characterization. Each record in core analysis contains the spatial attributes, i.e., longitude, latitude and elevation, where the sample was taken from, and the

nonspatial attributes, i.e., the reservoir properties. Due to the high cost, core analysis data is limited and can only provide reservoir properties for certain discrete points. For the areas without core samples, the reservoir properties are calculated by building correlations to the core analysis data in the surrounding areas. The essential assumption of this process is that a strong spatial correlation of reservoir properties exists so that the reservoir properties in an area can be correlated to the core samples nearby. Therefore, clustering core analysis data considering the spatial correlation is helpful in selecting the representative core analysis data. However, most existing spatial clustering methods consider spatial and nonspatial attributes independently. These methods are not suitable in clustering core analysis data where the spatial correlation plays important roles.

Spatial entropy is the extension of Shannon Entropy with the spatial configuration. It measures the distribution of a nonspatial attribute in the spatial domain (Claramunt, 2005; Leibovici, 2009). This thesis proposes to apply the spatial entropy to measure the nonspatial similarity and spatial correlation. A new spatial entropy-based clustering method, called SEClu, is introduced, which discovers clusters in core analysis data that are not only dense spatially but that also have a high spatial correlation across the space.

In addition, ANN has been introduced into reservoir characterization for building the correlations from well logs and core analysis data (Al-Bulushi et al. 2009; Aminian and Ameri, 2005; Mohagheg, 2000). Well log and core analysis data with the same depth interval are paired together to train the ANN. After proper training, a well trained ANN is capable of predicting reservoir properties based only on well log data for depth intervals where core analysis data is not available. As a data-driven method, ANN learns the

complex relationships between well log data and reservoir properties from the training data and, thus, training data selection is critical for the ANN's performance.

How to select a proper set of well log variables for ANN is not a trivial problem. A random selection or empirical selection based on limited experience for reservoir characterization may eliminate useful information, which will decrease the accuracy of the ANN. A complete well log contains approximately 20 well log variables, recording geological information from resistivity, spontaneous potential, sonic, or thermal sensors (Brock, 1986; Ellis, and Singer, 2007; Wong et al., 1995). Different well log variables have various levels of correlations to a target reservoir property. For example, Resistivity Logs are believed to have a closer relation with saturation. Additionally, well log variables generated from similar well logging sensors are highly correlated (Aminian and Ameri, 2005). When the number of dependent or irrelevant neural inputs rises, ANN tends especially to converge to local minima, which decreases the predication accuracy of ANN in reservoir characterization (Lin Y.H. et al. 1996). However, limited work has been conducted for selecting representative well log data for ANN-based reservoir characterization.

Fuzzy ranking (FR) is a global prioritizing technique that can automatically and quickly identify a subset of independent significant inputs for use in nonlinear systems (Lin Y.H. et al. 1996). It can identify the representative data for neural inputs mechanically without prior knowledge. By identifying a subset of representative variables from the well log, FR has the potential to improve the modeling and predication performance of ANN-based reservoir characterization. This chapter presents a novel fuzzy ranking-artificial neural network (FR-Neural) framework for reservoir

characterization from well logs. By removing irrelevant and highly correlated well log data, this framework reduces the risk of local minima and over-fitting, and implicitly increases the predication accuracy of the ANN.

The remainder of this chapter is organized as follows: Section 3.2 introduces a new density-based spatial clustering method for grouping core analysis data by considering spatial correlation. Section 3.3 presents the FR-Neural reservoir characterization framework. Section 3.4 evaluates the proposed methods using synthetic and real datasets.

3.2 Spatial Clustering of Core Analysis Data

In order to identify meaningful clusters from core analysis data, spatial clustering methods need to consider spatial attributes, nonspatial attributes and inherent spatial correlations during the clustering process. Especially, this chapter focuses on identifying arbitrary shaped clusters with spatial correlation. Identifying arbitrary shaped clusters does not require clusters forming specific geometrical shapes, which generalizes the method to more spatial clustering problems. Furthermore, spatial correlation always indicates the dependency between spatial and nonspatial attributes, with the chance that some cause and effect lead to it. Therefore, identifying clusters with spatial correlation would be interesting. In practice, spatial correlation can be in a large scale in which the nonspatial attributes may change gradually for a large extent. Under such condition, the values of nonspatial attributes might be similar in a small local area but differ significantly for the whole spatial correlated area. Identifying such areas as clusters is

implicitly interesting to reveal the effect leading to the spatial correlation. Therefore, in this research, nonspatial similarity is required in a small area but not in the whole cluster.

In the following, the spatial entropy is introduced and a discussion is given to justify that it is an unbiased measure in spatial clustering with respect to local nonspatial similarity and spatial correlation.

3.2.1 Spatial Entropy

Spatial entropy is an information measure of nonspatial attributes that also takes into account the influence of spatial spaces. Various forms of spatial entropy have been developed for how to quantify the extent of the role played by space (Li and Claramunt, 2006; Leibovici, 2009; Claramunt, 2005). The one from (Claramunt, 2005) is adopted here because it is simple and can handle both discrete and continuous nonspatial attributes.

Given a dataset D with a nonspatial attribute *prop* in spatial spaces $\{S_1,\ldots,S_m\}$, $\{D_1,\ldots,D_i,\ldots,D_n\}$ is a partition of D based on *prop*, i.e., $D_i \subseteq D$, $\bigcup D_i = D$ and $D_i \bigcap D_j = \varnothing, i \neq j$. p_i is the fraction of the number of objects in category D_i over the whole dataset D; i.e. $p_i=|D_i|/|D|$ and $\sum p_i = 1$. The intra-distance of D_i, denoted by d_i^{int} is the average distance between objects in D_i (as shown in Eq.(3.1)). The extra-distance of D_i, denoted by d_i^{ext} is the average distance of objects in D_i to other partition classes of D (as shown in Eq.(3.2)).

$$\text{Intra-Distance}: d_i^{int} = \begin{cases} \dfrac{1}{|D_i| \times |D_i - 1|} \sum_{j=1, j \in D_i}^{|D_i|} \sum_{k=1, k \neq j, k \in D_i}^{|D_i|} dist(j,k), & if\ |D_i| > 1 \\ \lambda, & \text{otherwise} \end{cases} \quad (3.1)$$

$$\text{Extra-Distance: } d_i^{ext} = \begin{cases} \frac{1}{|D_i| \times |D - D_i|} \sum_{j=1, j \in D_i}^{|D_i|} \sum_{k=1, k \neq j, k \notin D_i}^{|D - D_i|} dist(j,k), & if\ D \neq D_i \\ \beta, & \text{otherwise} \end{cases} \quad (3.2)$$

In Eq.(3.1), when D_i is empty or contains only one object, it is assumed that the intra-distance is very small and a small constant λ is assigned to d_i^{int} to avoid the influence of null values on the computation. In Eq.(3.2), when D_i includes all of the objects in D, i.e. all objects have similar values of *prop*, it is assumed that the extra-distance d_i^{ext} is very large, and a large constant β is assigned. *dist*(*j*,*k*) is the distance between objects *j* and *k* in the spatial space.

- **Definition 3.1 (spatial entropy) (Li and Claramunt, 2006; Claramunt. 2005):** The ***spatial entropy*** of dataset *D* based on its partition $\{D_1,\ldots,D_i,\ldots,D_n\}$ is defined as:

$$H_s(p_1,\ldots,p_i,\ldots,p_n) = -\sum_{i=1}^{n} \frac{d_i^{int}}{d_i^{ext}} p_i \log_2(p_i) \quad (3.3)$$

In this definition, a spatial configuration d_i^{int}/d_i^{ext} is added as a weight factor in the Shannon Entropy. The weight factor decreases when either the intra-distance decreases or the extra-distance increases, which enables spatial entropy to measure the spatial distribution. In addition, given *D* and its partition, the spatial entropy is similar to Shannon Entropy in that it reaches the maximum value when $p_1=\ldots p_i\ldots=p_n$.

3.2.2 Using Spatial Entropy in Spatial Clustering

This section will demonstrate that spatial entropy is a monotonic decreasing function for local nonspatial attribute similarity and spatial correlation.

3.2.2.1 Spatial Entropy vs. Local Nonspatial Similarity

The nonspatial attribute *prop* of the spatial dataset *D* can be viewed as a random variable with its probability density function approximated using a histogram. If the nonspatial attribute *prop* is random, it follows an even distribution. As the local nonspatial similarity increases, *prop* tends to be more concentrated.

It has been shown that the Shannon entropy of an even distribution reaches the maximum value and tends to decrease as the concentration of the distribution increases. Spatial Entropy H_s is a special form of Shannon entropy and has a spatial configuration weight factor d_i^{int}/d_i^{ext} . Even though each object's nonspatial attribute is correlated within the spatial spaces, the probability distribution of *prop* is independent from d_i^{int}/d_i^{ext} . Hence the weight factor d_i^{int}/d_i^{ext} does not influence the property of spatial entropy H_s, which is a measure of randomness. Therefore, when *prop* follows an even distribution, its spatial entropy value reaches the maximum; otherwise, the spatial entropy H_s decreases as the concentration of the probability distribution increases.

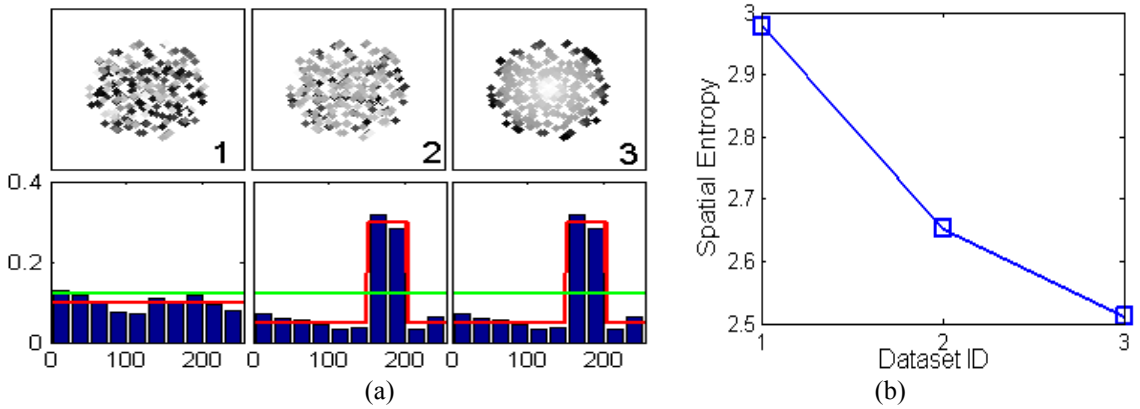

Figure 3–1 (a) Scatter plots and histograms for three grey value point datasets (b) Spatial entropy for the three datasets shown in (a).

In Figure 3–1 (a), datasets 1 and 2 have the same spatial attributes, but the points in dataset 2 have more similar nonspatial attributes than those in dataset 1. From the histograms it is evident that more than 60% of points in dataset 2 have grey values between [150,200] while values in dataset 1 are random. Figure 3–1 (b) shows that the spatial entropy value decreases from dataset 1 to dataset 2.

3.2.2.2 Spatial Entropy vs. Spatial Correlation

Spatial entropy measures spatial correlation by quantifying spatial diversity. As in (Tobler, 1970), the First Law of Geography states that spatial correlation generally exists. Furthermore, two supporting rules can be derived from it (Claramunt, 2005):

Rule 1: When different objects are close, diversity increases.

Rule 2: When similar objects are close, diversity decreases.

These two rules imply that spatial diversity increases when either the distance between different objects decreases or the distance between similar entities increases. In Eq.(3.1), the intra-distance d_i^{int} is defined as the average distance between similar objects and the extra-distance d_i^{ext} is defined as the average distance between diverse objects. They are integrated together with the form of d_i^{int}/d_i^{ext}, which keeps the spatial entropy decreasing when similar objects are close and diverse objects are far from each other. For spatial objects where similar nonspatial attribute values are close and where spatial objects where different nonspatial attributes are far from each other, d_i^{int}/d_i^{ext} decreases. Therefore, spatial entropy decreases when spatial correlation increases.

In Figure 3–1 (a), datasets 2 and 3 have the same spatial and nonspatial attributes, but different distributions for the nonspatial attribute. Here the spatial correlation

increases from 2 to 3 (high value points centered and low value in the periphery), and the spatial entropy value decreases accordingly, as shown in Figure 3–1 (b).

3.2.3 A Spatial Entropy-based Spatial Clustering Algorithm

In this section, a novel spatial clustering method, Spatial Entropy-based Clustering (SEClu), is introduced. Given a spatial dataset *SD* with a nonspatial attribute *prop*, a symmetric function *dist* measuring the distance in the spatial space, and parameters *Eps*, *MinNum* and *MaxSp*, the following definitions are introduced:

- **Definition 3.2 (neighbourhood):** The *neighbourhood* of spatial object *p*, denoted by $N_{Eps}(p)$, is defined as $N_{Eps}(p)=\{q \in SD \mid dist(p,q) \leq Eps\}$.

The neighbourhood definition is taken from (Ester et al., 1996). *dist* can be any form of a symmetric function, *Eps* is the threshold based on the dist function, and $N_{Eps}(p)$ returns all of the objects in *SD* whose distance from *p* is smaller than *Eps*.

SEClu extends DBSCAN by applying spatial entropy to control the local nonspatial similarity and spatial correlation of $N_{Eps}(p)$. The previous discussion demonstrates that the spatial entropy value decreases for the local nonspatial attribute similarity and that spatial correlation increases. Therefore, SEClu introduces the maximum threshold of spatial entropy, denoted by *MaxSp*.

In SEClu, a *core object* is an object whose neighbourhood is (1) dense, i.e., it has at least *MinNum* neighbours in spatial spaces and (2) has similar nonspatial attributes and high spatial correlation in its neighbourhood satisfying $H_s(N_{Eps}(p)) \leq MaxSp$. A *border object* is a neighbour object of a core object which is not a core object itself. Objects other than core objects or border objects are *noise*.

- **Definition 3.3 (directly density-spEntropy-reachable):** A spatial object p is directly density-spEntropy-reachable to an object q w.r.t *Eps*, *MinNum*, *MaxSp* if (1) $q \in N_{Eps}(p)$; (2) $N_{Eps}(p) \geq MinNum$; and (3) $H_s(N_{Eps}(p)) \leq MaxSp$.

In the above definition, the second condition examines the density of the neighbourhood of p. The third condition examines nonspatial attribute of the neighbourhood of p. A smaller value of spatial entropy H_s implies that objects in $N_{Eps}(p)$ have higher nonspatial similarity and spatial correlation. Directly density-spEntropy-reachable is symmetric for core objects as well as one core object and one border object. But it is asymmetric for two border objects.

- **Definition 3.4 (density-spEntropy reachable):** Spatial objects p and q are *density-spEntropy reachable* (*DSR-reachable*) w.r.t *Eps*, *MinNum*, *MaxSp*, denoted by $DSR(p,q)$, if there is a chain of objects $p_1,\ldots,p_n$, $p_1=q$, $p_n=p$ such that p_{i+1} is directly density-spEntropy reachable from p_i.
- **Definition 3.5 (density-spEntropy-based cluster)** A *density-spEntropy based cluster* C is a non-empty subset of *SD* satisfying: $\forall\, p, q \in SD$, if $p \in C$ and $DSR(p,q)$ holds, then $q \in C$.

It is obvious that for each pair of objects $(p,q) \in C$, when C is a density-spEntropy based cluster, $DSR(p,q)$ holds. Therefore, SEClu finds a cluster by identifying all objects that are density-spEntropy reachable.

The SEClu algorithm is shown in Figure 3–2. It starts by querying the neighbourhood of an arbitrary object o in the spatial space to see if it is dense enough $N_{Eps}(p) \geq MinNum$. If not, p is labeled as noise; otherwise, SEClu continues to check the nonspatial attribute. If the nonspatial attribute of p's neighbourhood has a random pattern,

i.e., it cannot satisfy $H_s(N_{Eps}(p)) \leq MaxSp$, then p is labelled as noise. Otherwise, a new cluster C is created and all objects $x \in N_{Eps}(p)$ are placed in C. The neighbourhood of each of p's neighbours is examined in the same way to see if it can be added to C. This process is repeated until all objects that are density-spEntropy reachable to p have been added to cluster C. If cluster C cannot be expanded further, SEClu chooses another unlabelled object and repeats this process until all objects have been assigned to a cluster or labeled as noise. The average complexity of SEClu is O ($n(\log n+k^2)$), where n is the number of the objects in SD and k is the average number of objects in $N_{Eps}(p_i)$.

3.2.3.1 Calculating Spatial Entropy Efficiently

In SEClu, the spatial entropy H_s is computed on the nonspatial attribute of $N_{Eps}(p)$. To be able to use H_s to measure the spatial correlation, a partition process of $N_{Eps}(p)$ is generated in the first step. Given a spatial dataset $D=N_{Eps}(p)$ with the nonspatial attribute *prop,* if *prop* is discrete it is binned into n slots with different values. If *prop* is continuous, it is binned into n contiguous slots ($\chi_1,\ldots,\chi_i,\ldots\chi_n$) with the interval of Δ. Then each object in D is assigned to a unique slot based on its *prop* value, and a partition of D, denoted by $\{D_1,...,D_i,\ldots,D_n\}$, is generated.

The number of subsets n should be selected carefully. If n is too large, each subset may contain a very small number of data and also result in a high computational cost. Sturges' rule (Sturges, 1926) is widely recommended for choosing a histogram interval since it provides a good approximation of the best n in capturing the distribution pattern. The Sturges' rule is adopted, and the subset number n is given by Eq. (3.4). Since SEClu is a density-based method, an effective way is to assign *MinNum* to N.

$$n = 1 + \log_2 N, N \text{ is the number of objects in } D \tag{3.4}$$

Algorithm: SEClu (*SD*, *Eps*, *MinNum*, *MaxSp*)

Input:(1) A spatial dataset *SD* (2) Searching distance *Eps*

(3) Minimum number of objects in a cluster *MinNum*

(4) Maximum spatial entropy threshold *MaxSp*

Output: Clustering results for each $p \in SD$

```
01  for each unclassified p∈SD do
02   if |N_Eps(p)|<MinNum or H_s(N_Eps(p))>MaxSp
03    mark p as noise;
04   else
05    create a new cluster C and add x∈N_Eps(p) in C;
06    add x∈N_Eps(p) into a queue Q;
07    while Q is not empty do
08     q = first object in Q and remove q from Q;
09     if |N_Eps(q)|>=MinNum and H_s(N_Eps(q))<=MaxSp
10      for each object t∈N_Eps(q) do
11       if t is unclassified,
12        add t to Q and add t into C;
13       if t is noise
14        add t into C;
```

Figure 3–2 Pseudo code of the SEClu algorithm

In practice, spatial entropy H_s is computed numerically. Since prior knowledge of the distribution of *prop* is always unknown, p_i is estimated from the frequency $p_i=|D_i|/|D|$. d_i^{int} and d_i^{ext} can be computed from Eqs. (3.1) and (3.2), respectively. Also, spatial entropy H_s needs to be normalized in order to make a fair comparison. It has been demonstrated that $d_i^{int} \le 2d_i^{ext}$ (Li and Claramunt, 2006). Also, for a discrete random variable its Shannon entropy value satisfies $0 \le -\sum_{i=1}^{n} p_i \log_2(p_i) \le \log_2 n$. Then

$$0 \le H_s = -\sum_{i=1}^{n} {d_i^{int}}/{d_i^{ext}}\ p_i \log_2(p_i) \le -2\sum_{i=1}^{n} p_i \log_2(p_i) \le 2\log_2 n$$

In the following, all spatial entropy values are normalized by $H_s/2\log_2 n \in [0,1]$. Figure 3–3 shows the pseudocode for the spatial entropy calculation.

Function: Spatial Entropy $H_s(D)$

Input: A spatial dataset *D* (can be a subset of *SD* in Figure 3–2)

Output: Spatial entropy value of *D*

```
01  Bin D into {D1,…,Di,…,Dn}based on D.prop;
02  for each Di do
03    Compute pi, di^int and di^ext (from Eqs. (3.1) and (3.2));
04  compute Hs (from Eq.(3.3));
05  return Hs;
```

Figure 3–3 Pseudo code of the spatial entropy computational function

3.2.3.2 Spatial Entropy Parameter *Maxsp*

In SEClu, the parameters *Eps* and *MinNum* can be determined using the heuristic method in (Ester et al., 1996). Besides, given *Eps* and *MinNum*, *MaxSp* can be determined by the following rationale. Meeting the requirement of $N_{Eps}(p) \geq MinNum$, p should form a core object if it satisfies $H_s(N_{Eps}(p)) \leq MaxSp$. Thus *MaxSp* can be determined by seeking a threshold that makes the cluster the "thinnest", i.e. the least density-spEntropy reachable cluster in the spatial dataset. In order to serve as a good spatial correlation measurement parameters, the candidate *MaxSp* specifies the highest spatial entropy value which does not identify spatial correlated clusters as noise. As an example, Figure 3–4 suggests an way to determine *MaxSp*. When *Eps* and *MinNum* are fixed, the core object number is changing with respect to the *MaxSp* value. The *MaxSp* threshold is determined with the highest gradient, which appears as the first jump point in Figure 3–4. Here all objects with the spatial entropy value higher than the threshold (above the line) are noise and the others are core objects.

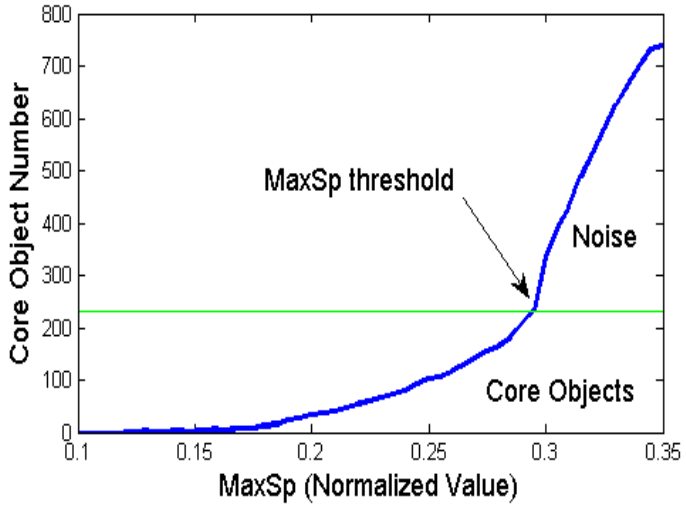

Figure 3–4 Core objects number vs. *MaxSp*

3.3 FR-Neural Reservoir Characterization

Although core analysis data are accurate, they can only provide reservoir properties for certain discrete points. In order to acquire reservoir properties in a large scale, ANN has been introduced to build correlations from other field data to core analysis data. In this section, a new fuzzy ranking-artificial neural network (FR-Neural) framework is introduced to characterize reservoir properties by building correlations between well log and core analysis data. As shown in Figure 3–5, the FR-Neural framework includes two steps: the fuzzy ranking and the pattern recognition.

The fuzzy ranking step is to select proper well log variables for the neural network. By selecting the representative neural inputs, fuzzy ranking helps implicitly improve the pattern recognition performance of the ANN in the following step. In the pattern recognition step, a MLP neural network is trained to learn the desired complex relations between selected well log variables and core analysis data. After verification, it further predicts the reservoir property values for intervals where core analysis data is non-existent.

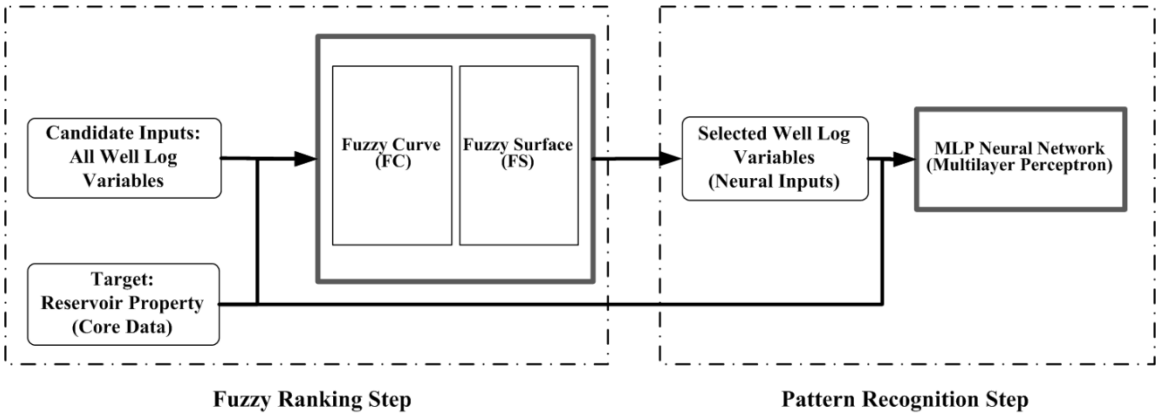

Figure 3–5 The proposed FR-Neural reservoir characterization framework

3.3.1 Fuzzy Ranking Step

The fuzzy ranking step includes Fuzzy Curve (FC) and Fuzzy Surface (FS). FC ranks all potential well log variables according to their relevance to the target reservoir properties and eliminates irrelevant variables. FS identifies the dependency among well log variables selected by FC and eliminates the redundant highly correlated variables.

3.3.1.1 Fuzzy Curve (FC)

FC is based on the assumption that the most important input variable plays the most important role in approximating the output (Lin et al., 1998). It simulates the relationship between each potential input and the output by building a fuzzy curve function, and meaningful inputs are selected based on the closeness between the simulated and real relationship.

In the reservoir characterization case, all well log variables serving as the candidate inputs and the target reservoir property as the output. The well log variables are denoted by $X=\{x_i \mid i=1,2,\ldots,n\}$ and the target reservoir property is denoted by y. Each well log variable has m data points, (x_{ik}, $k=1,2,\ldots,m$) and the corresponding reservoir property values (y_k, $k=1,2,\ldots,m$). With no prior knowledge about the relationship between x_i and y, the objective of FC is to select a subset of variables from X, $SX=\{x_s \mid s=1,2,\ldots l,$ and $l \leqslant n\}$ so that a nonlinear relation exists between SX and y:

$$y \approx f(SX),\ SX = \{x_s \mid s = 1,2,...,l,\ \texttt{and}\ l \leq n\}$$

The unselected variables tend to have a random correlation with y.

Since different well log variables have different value ranges, the first step of FC is to normalize all the candidate well log variables, as shown in Eq.(3.5).

$$x_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{3.5}$$

Then FC builds fuzzy membership functions for every data point in the x_i-y space. In this study, the fuzzy membership functions are in Gaussian form, as shown in Eq.(3.6), where b is a constant that controls the span of the Gaussian function.

$$\mu_{ik}(x_i) = \exp(-(\frac{x_{ik} - x_i}{b})^2) \tag{3.6}$$

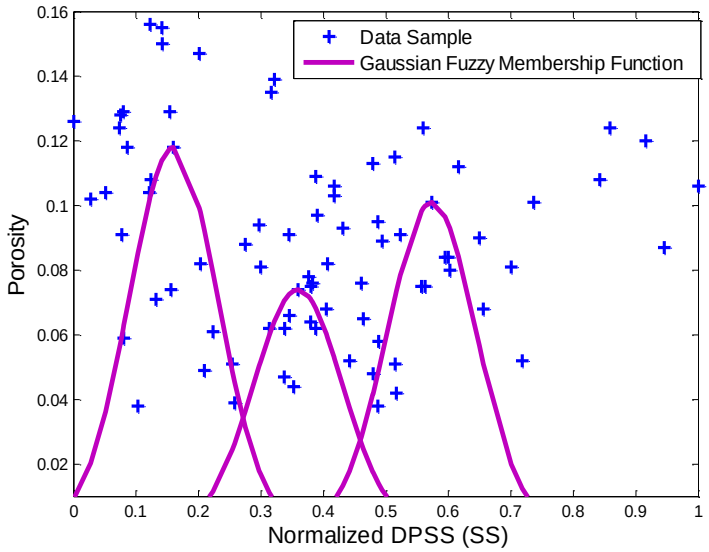

Figure 3–6 Gaussian fuzzy membership functions in the DPSS-Porosity space

The Gaussian fuzzy membership function gives a prediction of the target reservoir property y when the well log variable x_i changes slightly in a neighbourhood close to x_{ik}. For example, Figure 3–6 shows the scatter plot of data points between the normalized DPSS (Density Porosity Sandstone Scale), a well log variable from sonic sensor, and reservoir porosity, denoted by '+'. The Gaussian fuzzy membership functions $\mu_{ik}(x_i)$ are built for each data sample x_{ik} in normalized DPSS. Figure 3–7 shows the fuzzy

membership function curves $y_k \mu_{ik}(x_i)$ for three points, where y_k is the reservoir porosity value at the same depth interval as x_{ik}.

Next, FC integrates all the fuzzy membership functions forming a fuzzy curve $c_i(x_i)$. Specifically, $c_i(x_i)$ defuzzifies all the Gaussian fuzzy membership functions $\mu_{ik}(x_i)$ by normalized summation, as Eq.(3.7), which is an approximation of the relationship between x_i and y.

$$c_i(x_i) = \frac{\sum_{k=1}^{m} y_k \times \mu_{ik}(x_i)}{\sum_{k=1}^{m} \mu_{ik}(x_i)} \tag{3.7}$$

Continuing the previous example in Figure 3–6, all Gaussian functions are weighted-averaged using Eq.(3.7), where the weight is the target y_k corresponding to x_{ik}. The solid line in Figure 3–7 shows the fuzzy curve of DPSS to the reservoir porosity.

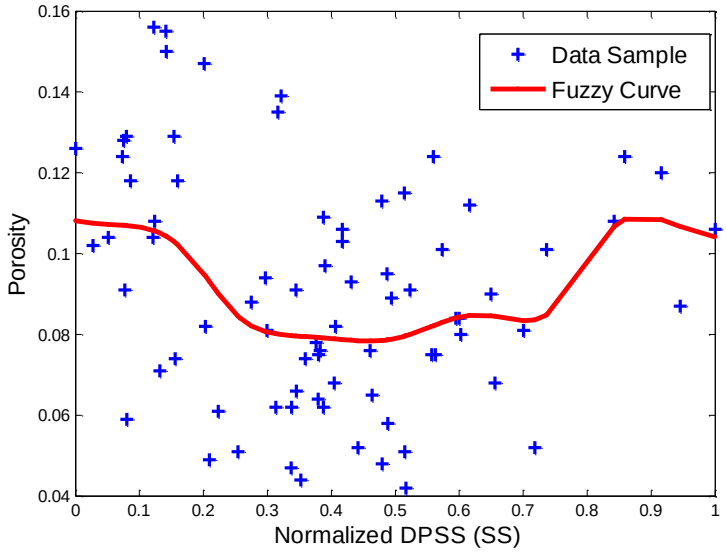

Figure 3–7 Fuzzy curve of DPSS to porosity

FC is a weighted local average of y_k along each x_i axis, where the size of a local neighbourhood is controlled by b in Eq.(3.6). When b is large, $\mu_{ik}(x_i)$ is equal to approximately 1 for all x_{ik} so that $c_i(x_i) \approx \sum_{k=1}^{m} y_k / m$ equals the average of y_k at every x_{ik}. When b is small, $\mu_{ik}(x_i) \approx 1$ only for $x_i = x_{ik}$, while 0 elsewhere, so $c_i(x_i) \approx y_k$ only for $x_i = x_{ik}$ and $c_i(x_i) \approx 0$ elsewhere. Hence $c_i(x_i)$ is an approximation of y based on x_i. Here, b controlling the size of the local neighbourhood is critical to the approximation of y. When b is too large, $c_i(x_i)$ is not sensitive to a local change. When b is too small, $c_i(x_i)$ will lose the average information. b is chosen to be 0.08 in this reservoir characterization study.

$$MSEc_i = \frac{\sum_{k=1}^{m} (c_i(x_{i,k}) - y_k)^2}{m \times \mathrm{var}(y)} \tag{3.8}$$

The more information x_i contains, the closer the approximation c_i is to output y. $MSEc_i$ is the normalized mean square error to measure the distance from fuzzy curve c_i to y, as shown in Eq.(3.8) where $var(y)$ is the variance of y used to scale the mean square error.

Thus the last step of FC is to sort the well log variables in ascending order in terms of $MSEc_i$. The most important well log variable is the one with the smallest value of $MSEc_i$. Specifically, if x_i is a random noise and has no relation with the output, then the fuzzy curve c_i to x_i tends to be flat which also results in a high mean square error.

However, well log variables are highly correlated, which not only increases the risk of a local minimum but also brings additional computational cost to neural networks. Hence elimination of highly correlated variables is essential for the pattern recognition step. To remove the correlated well log variables, the fuzzy surface is to be used.

3.3.1.2 Fuzzy Surface (FS)

FS identifies and eliminates the highly correlated variables from the results selected from the FC step. FS is based on the assumption that two independent variables do a better job of approximating the output than two correlated variables (Lin et al., 1998).

For two selected variables x_i and x_j from the previous FC step, FS is defined in Eq.(3.9), where μ_i and μ_j are the Gaussian fuzzy membership functions of x_i and x_j defined by Eq.(3.6). s_{ij} computes the weighted average of output y_k based on the information from the combination of x_i and x_j.

$$s_{ij}(x_i, x_j) = \frac{\sum_{k=1}^{m} y_k \times \mu_{ik}(x_i) \times \mu_{jk}(x_j)}{\sum_{k=1}^{m} \mu_{ik}(x_i) \times \mu_{jk}(x_j)} \tag{3.9}$$

The more information contained between x_i and x_j, the better the approximation $s_{ij}(x_i,x_j)$ will be to the output y. The normalized mean square error function, Eq.(3.10), is used to compute the distance from $s_{ij}(x_i,x_j)$ to y, where $var(y)$ is the variance of y. $MSEs_{ij}$ from two independent variables x_i and x_k will be smaller than that from two correlated variables x_i and x_j.

$$MSEs_{ij} = \frac{\sum_{k=1}^{m} (s_{ij}(x_{ik}, x_{jk}) - y_k)^2}{m \times \mathrm{var}(y)} \tag{3.10}$$

Below, the process of applying fuzzy ranking in the neural inputs selection for reservoir characterization is summarized. Assume that the original well log contains n candidate variables, $X=\{x_i \mid i=1,2,\ldots n\}$, and the target reservoir property is y. fuzzy ranking works as follows:

Figure 3–8 shows the pseudo code of fuzzy ranking in the FR-Neural reservoir characterization framework. It takes three input arguments: $X=\{x_i \mid i=1,2,\ldots n\}$ containing n well log variables, y the target reservoir property, and α% the elimination threshold in the FC process. Fuzzy ranking starts by FC ranking n candidate well log variables in the candidate list (*CL*). In line 5, the variable with the smallest $MSEc_i$ is selected as the most important variable, denoted by $\overline{X}$ and added to the selected list *SX*. Meanwhile, α% variables in *CL* with highest $MSEc_i$ are eliminated. After FC, $\overline{X}$ is paired with each of the remaining (1-α%)n-1 candidate variables remaining in the *CL* and FS is applied to rank them. In line 11, the variable with the highest $MSEs_{ij}$ is eliminated from *CL* as dependent redundancy while the one with the smallest $MSEs_{ij}$ is selected, added to *SX* and assigned to $\overline{X}$. Using $\overline{X}$, the FS process is repeated until the number of variables in the *CL* is smaller than 2. In the end, the variables in *SX* form the final selected results, and are returned back and used as neural inputs in the subsequent pattern recognition step.

Fuzzy Ranking (*X*, *y*, *α*%**)**

Input: (1) Candidate well log variables $X=\{x_i \mid i=1,2,\ldots n\}$

(2) Target reservoir property *y*

(3) Elimination threshold *α*% in the Fuzzy Curve step

Output: Selected well log variables $SX=\{x_s \mid s=1,2,\ldots l,$ and $l \leq n\}$

01: Initialize candidate list *CL*=*X* and selected list *SX*=*null*;

02: **for** each variable $x_{i \in} CL$ /* **Fuzzy Curve Step** */

03: Compute the fuzzy curve $c_i(x_i)$ by Eq. (3.7);

04: Compute $MSEc_i$ by Eq. (3.8);

05: $\overline{X}$ =the variable in *CL* with the smallest $MSEc_i$, and add $\overline{X}$ into *SX*;

06: Remove $\overline{X}$ and *α*% variables with highest $MSEc_i$ from *CL*;

07: **while** (SizeOf (*CL*) > 1) **do** /* **Fuzzy Surface Step** */

08: **for** each variable $x_{i \in} CL$

09: Compute the fuzzy surface $s_{ij}(x_i, \overline{X})$ by Eq. (3.9);

10: Compute $MSEs_{ij}$ by Eq. (3.10);

11: $\overline{X}$ =the variable in *CL* with the smallest $MSEs_{ij}$, and add $\overline{X}$ into *SX*;

12: Remove $\overline{X}$ and the variable with the highest $MSEs_{ij}$ from *CL*;

13: **if** ($CL \neq null$)

14: Add *CL* into *SX*;

15: **return** *SX*;

Figure 3–8 Pseudo code of fuzzy ranking for well log variable selection

3.3.2 Pattern Recognition Step

The pattern recognition step implements the MLP neural network to simulate the complex relationship between the selected well log variables and the target reservoir property. After the fuzzy ranking step, representative training data for reservoir characterization are selected as neural inputs, which implicitly help in improving the pattern recognition performance of MLP.

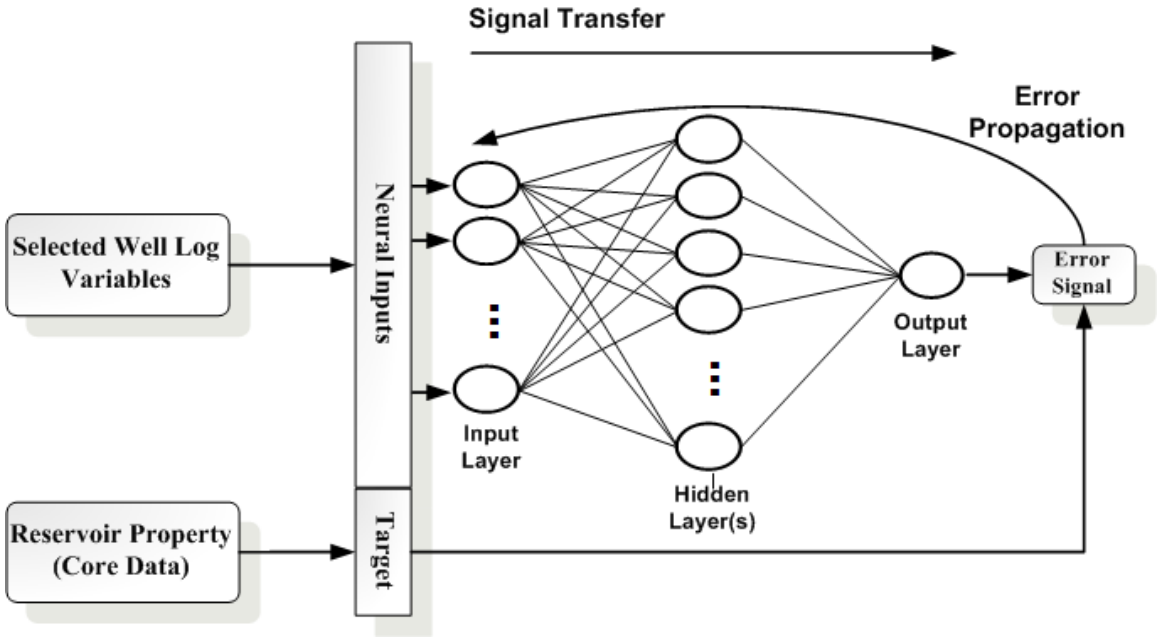

Figure 3–9 The Multilayer Perceptron (MLP) model in the pattern recognition step for reservoir characterization

MLP in this work, as shown in Figure 3–9, is designed with one input layer, one hidden layer and one output layer. Neurons in the input layer are automatically initialized by the selected well log variables from the fuzzy ranking step, and the output layer has only one neuron in terms of the target reservoir property. Previous studies have indicated that a network with one hidden layer can approximate any continuous function given

sufficient hidden neurons (Haykin, 1999). Neurons in each layer are fully connected with the ones in the adjacent layers with weighted links. Each neuron applies an activation function that processes the weighted sum results and transforms to the others. In this way, input signals transfer through the whole network and generate the output in the end. As a supervising learning mechanism, for each pair of neural inputs there is a desired target value. For example, the neural input can be the values of the selected well log variables at a certain depth interval, and the target value can be the core analysis data at the same depth interval. As shown in Figure 3–9, the difference between the network output and the desired target will generate an error signal, which transfers back to the network modifying the link weights via back-propagation (BP) algorithms. After being trained by a number of training data, MLP can not only generate the reservoir property very closely to the desired value for training samples but can predict the desired target for unseen neural inputs.

3.4 Experiments

In this section, the proposed SEClu spatial clustering algorithm and FR-Neural reservoir characterization framework are evaluated using synthetic and real datasets. All experiments are performed on a 2.8GHz PC with 3G memory.

3.4.1 SEClu on Synthetic Data

This experiment demonstrates SEClu in identifying clusters with spatial correlation. Figure 3–10 shows two sample datasets. Each dataset includes *x, y* coordinates as the spatial attribute and a grey value as the nonspatial attribute. The datasets in Figure 3–10 have different shapes and follow different spatial correlation functions. Figure 3–10 (a)

includes three round shaped groups. The grey values of points in group 1 decrease as an exponential function with distance from the center. The grey values in group 3 increase linearly with distance to the center. The grey values of group 2 are random. Since both groups 1 and 3 have strong spatial correlation they should be identified as clusters. Group 2 does not form a cluster with spatial correlations and should be labelled as noise. Figure 3–10 (b) shows three irregularly shaped groups. The S-shaped and new moon shaped groups have clusters with spatial correlation while the objects in the V-shaped group have random nonspatial attributes and should be identified as noise.

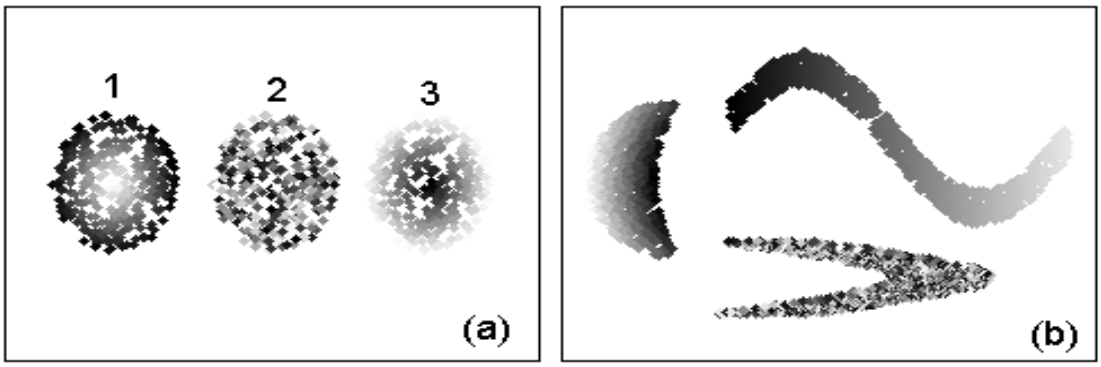

Figure 3–10 Two synthetic spatial datasets

For SEClu, the parameters are set (*Eps*=0.017, *MinNum*=18, *MaxSp*=0.29) for two datasets in Figure 3–10. Figure 3–11 shows the clustering results, where the black points are noise and the color points are clusters. From the figure, it is evident that: first, SEClu discovers clusters with spatial correlations successfully. In (a), data groups with either a high value center correlation or a low value center correlation are identified as clusters, while the random one is labelled as noise. Second, since SEClu is a density-based

clustering method, it can discover clusters with irregular shapes. For example, SEClu identifies the S-shaped and the new moon shaped spatial correlated clusters.

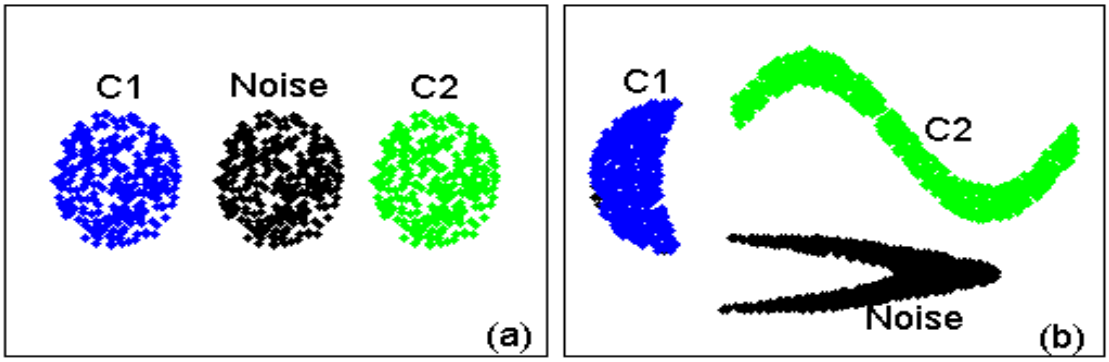

Figure 3–11 SEClu clustering results

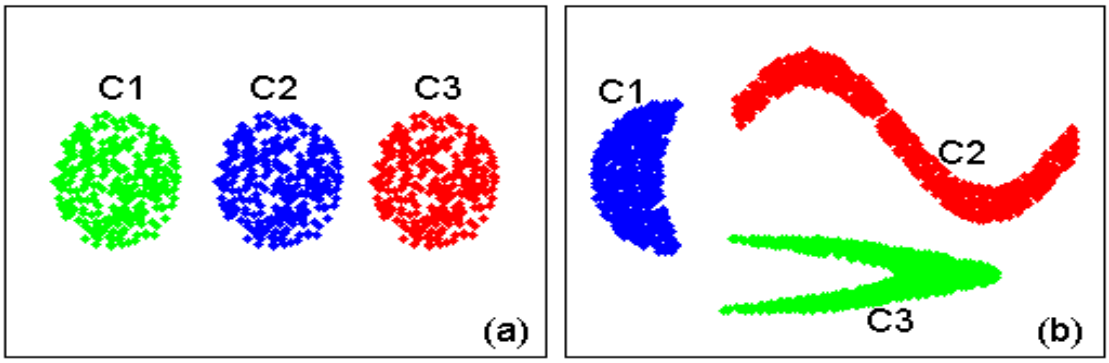

Figure 3–12 GDBSCAN clustering results

SEClu is compared with GDBSCAN (Sander et al., 1998). The parameters to GDBSCAN are set as (*Eps*=0.017, *MinPts*=18), which is similar to the SEClu configuration. Figure 3–12 shows the clustering result of GDBSCAN. Considering the nonspatial attribute as independent from spatial attributes, GDBSCAN incorrectly labels data with random distributed grey values as clusters. Compared with the results of GDBSCAN, SEClu finds more meaningful clusters. A detailed accuracy comparison

between SEClu and GDBSCAN is shown in Table 3–1. From the table, the accuracy of SEClu to the three datasets is 100% while GDBSCAN is 67%, which demonstrates SEClu performs better than GDBSCAN in identifying clusters with spatial correlations.

Table 3–1 Accuracy comparison between SEClu and GDBSCAN

Dataset	**SEClu**			**GDBSCAN**		
(No. of data objects)	**Correct**	**Error**	**Accuracy**	**Correct**	**Error**	**Accuracy**
Dataset(a) (900 pts)	900	0	**100%**	600	300	**66.7%**
Dataset(b) (1500 pts)	1500	0	**100%**	1000	500	**66.7%**

3.4.2 SEClu on Real Data

The second experiment is performed on the real core porosity data taken from south Alberta, Canada. Porosity is one of the most important reservoir properties, which measures the fraction of the void volume over the total volume of rocks. As shown in Figure 3–13, within the area from 28-R1-W5 to 26-R27-W4 using DLS (Dominion Land Survey, 2012), there are 120 cored wells containing information about reservoir porosity. Each cored well is represented as a point on the map with the spatial and nonspatial attributes. The spatial attributes refer to the locations of the well, i.e., longitude and latitude. The nonspatial attribute refers to the average porosity value from core analysis data. The clustering result from SEClu is compared with that from GDBSCAN. For SEClu, the parameters are set to be (*Eps*=4000, *MinNum*=5, *MaxSp*=0.32). For a fair comparison, the parameters to GDBSCAN are set to be (*Eps*=4000, *MinNum*=5), and the

clustering results from GDBSCAN and SEClu are listed in Figure 3–14 and Figure 3–15, respectively.

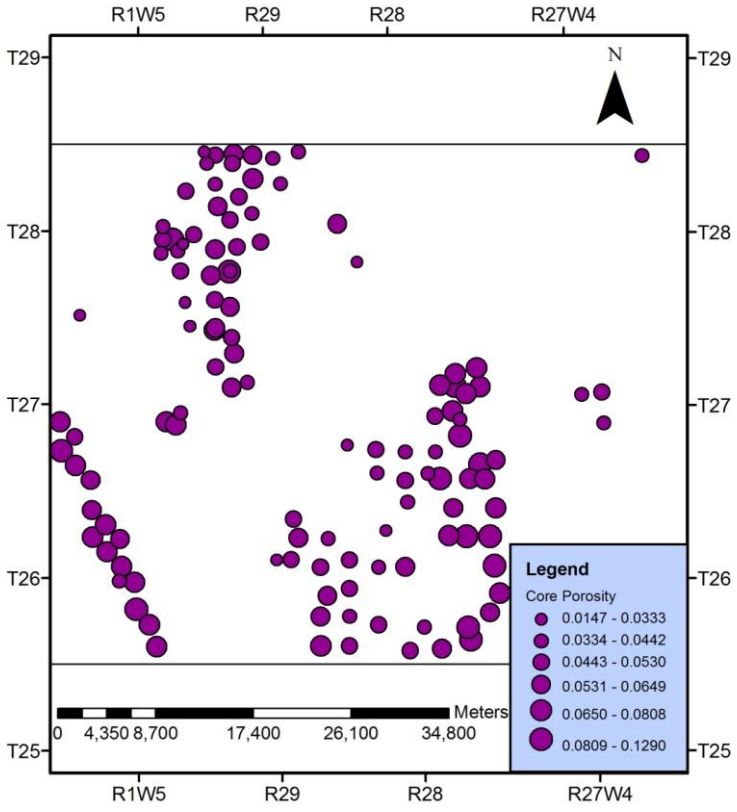

Figure 3–13 Cored wells within the area from 28-R1-W5 to 26-R27-W4

In Figure 3–14, GDBSCAN finds 3 density-reachable clusters in which within a radius of 4000 meters at least 5 core points exist (*Eps*=4000, *MinNum*=5). In comparison, as shown in Figure 3–15, SEClu identifies 8 density-spEntropy-reachable clusters. Clusters identified from SEClu satisfy not only the density requirement, as GDBSCAN, but also the nonspatial constraint that the spatial entropy of nonspatial attributes in the same cluster is smaller than a given threshold. Hence SEClu further separate the clusters discovered in GDBSCAN into small clusters. For example, clusters 1, 2, 3 and 4 from

SEClu in Figure 3–15 are subclasses of cluster 2 from GDBSCAN in Figure 3–14. Since SEClu considers the nonspatial attributes and spatial correlation during the clustering, the porosity values of data points in the same cluster tend to be similar. For example, the clusters 2 and cluster 4 have distinct core porosity values. The cluster 2 has core porosity values from 0.060 to 0.095, while the cluster 4 has a smaller core porosity value from 0.027 to 0.042. Even though they are adjacent to each other, SEClu identified them as two separate clusters while GDBSCAN groups them into one cluster.

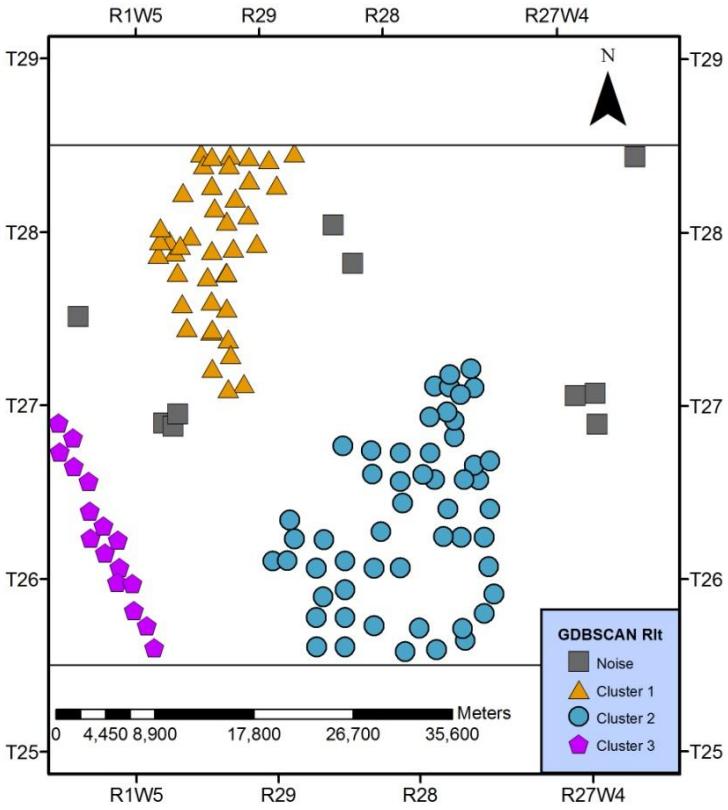

Figure 3–14 Clustering results from GDBSCAN over core analysis data

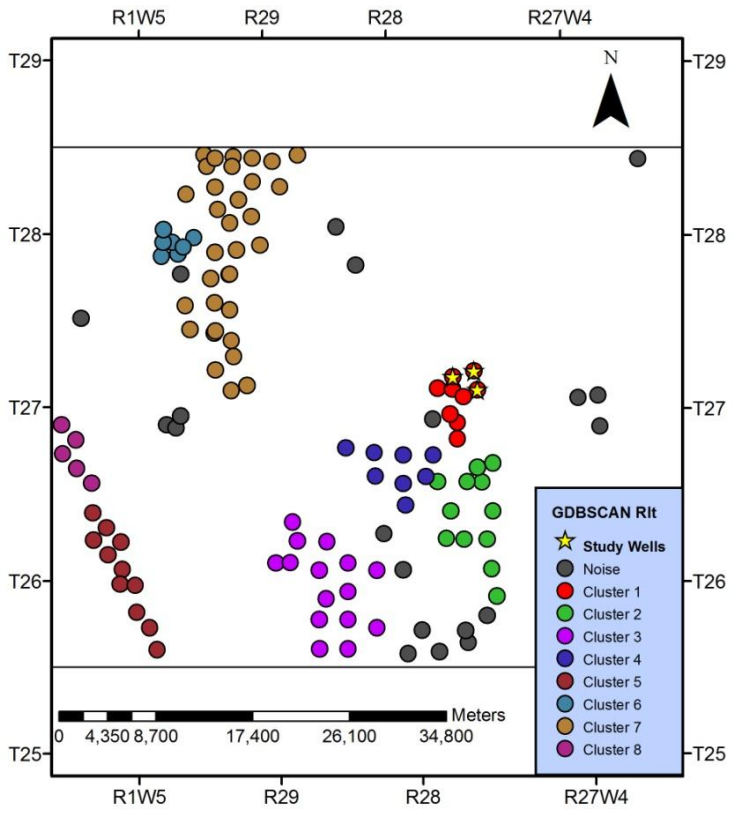

Figure 3–15 Clustering results from SEClu over core analysis data

3.4.3 FR-Neural on Real Data

The proposed FR-Neural framework for reservoir characterization is demonstrated using real industrial data. From the core analysis data clustering, three wells located in the same cluster and symbolized as pentagrams in Figure 3–15 are selected for the case study. The three wells are denoted as W-1, W-2 and W-3, respectively, and the corresponding digital well log data and core analysis data are reviewed in Table 3–2. Each well has 20~23 well log variables serving as candidate neural inputs to the MLP and porosity values from core analysis data serve as target data. For instance, W-1 has 21 well log variables with each of them rangeing from 278.4 meters to 2329.8 meters below surface. In addition, W-1 has

79 data samples of porosity data ranging from 2,240.0~2,265.4 meters below surface. Below W-1 is used as an example to demonstrate the proposed FR-Neural framework. W-1 has 21 well log variables organized into seven categories, including Resistivity, Sonic, Thermal, Neutron, Density, Caliper and Spontaneous Potential logs, based on the well logging sensors used. Note that not all candidate well log variables have the same degree of relation to the target reservoir porosity; therefore, a subset of well log variables correlated to the porosity needs to be selected.

Table 3–2 Overall data description

Well ID	Candidate Inputs: All Well Log Variables		Target Data: Porosity (Core Data)	
	NO. of Variables	Depth (Unit: meter)	NO. of Data Samples	Depth (Unit: meter)
W-1	21	278.4~2329.8	79	2240.0~2265.4
W-2	23	291.8~2199.2	76	2109.0~2135.6
W-3	20	2126.0~2333.2	76	2232.0~2254.3

3.4.3.1 Well Log Variables Selection

To identify the representative data from a well log, 21 well log variables and core porosity records for W-1, with the same depth intervals, are paired together. This forms a data space (x_i, y), i=1,2,…,21, where x_i denotes the 21 well log variables and y denotes the target reservoir porosity.

The first step is to identify a subset of well log variables containing direct information for characterizing porosity using FC. FC ranks all 21 candidate inputs along with ascending $MSEc_i$ and the result is listed in Table 3–3. In Table 3–3, CAL2 has the

smallest $MSEc_i$ of 0.78725 and thus ranks as the most direct well log variable for the target reservoir porosity. In contrast, HCAL has the largest $MSEc_i$ and thus ranks as the least important well log variable. Visual observations from fuzzy curves can also be indicative of the different levels of connection of the candidate inputs to the target reservoir porosity. From the previous discussion, a flatter curve means that the corresponding variable contains less or more random information for the target output. Figure 3–16 shows fuzzy curves for partial well log variables. Among them, HCAL, in Figure 3–16 (b), has the flattest fuzzy curve, which demonstrates in another way that it contains the least information compared to other variables. If the elimination threshold for FC is set to 30%, the last seven variables including HCAL, AF20, AF30, HDRA, AT9C, PEFZ and GDEV are removed from the candidate inputs.

Table 3–3 Fuzzy Curve ranking result

Candidate Well Log	$MSEc_i$	Rank	Candidate Well Log	$MSEc_i$	Rank	Candidate Well Log	$MSEc_i$	Rank
CAL2	0.787250	1	GR	0.842015	8	GDEV	0.875072	15
DPSS	0.796621	2	CNTC	0.842649	9	PEFZ	0.882191	16
DPDL	0.796866	3	SP	0.848234	10	AT9C	0.886251	17
RHOZ	0.796887	4	CFTC	0.858982	11	HDRA	0.904673	18
DPLS	0.796922	5	NPSS	0.870224	12	AF30	0.908325	19
AF90	0.804379	6	NPDL	0.871355	13	AF20	0.941702	20
AT90	0.841790	7	NPLS	0.872583	14	HCAL	0.951503	21

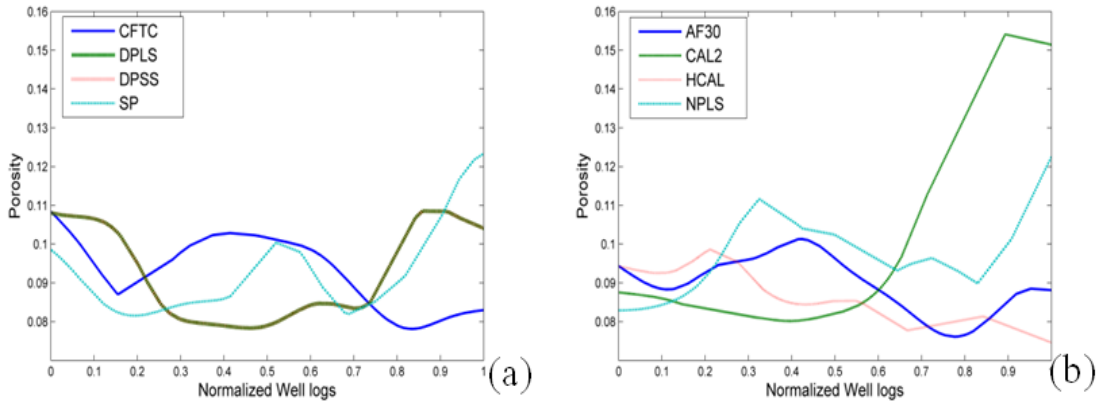

Figure 3–16 Fuzzy curves for partial well log variables: (a)AF30, CAL2, HCAL, NPLS (b)CFTC, DPLS, DPSS, SP.

FC selects the candidate well log variables by analyzing their information content relative to the reservoir porosity. In the next step, FS is implemented to remove the highly dependent variables from the candidate well log variables.

In the FC step, CAL2 is selected as the most direct well log variable for the property of reservoir porosity. Hence in the FS analysis, CAL2 is used as the reference variable in which $MSEs_{ij}$ is calculated between CAL2 and each of the remaining 13 candidate inputs. Table 3–4 shows the first iteration result with an ascending $MSEs_{ij}$ from FS. From Table 4, (CAL2, GR) has the minimum $MSEs_{ij}$ so that GR is identified as the second important variable. GR records the signal via the Gamma Ray well logging tool and has minimum dependence with CAL2 from the Caliper Log; therefore, the paired well log variables (CAL2, GR) are selected. In comparison, AF90 is discarded because the pair (CAL2, AF90) shows the highest value of *MSE*. After the first iteration, 11 candidate inputs are left. In the second iteration, GR takes the place of CAL2 and is used as the reference variable to evaluate the remaining 11 inputs.

Table 3–4. The first iteration FS ranking result

Candidate Well Log Variables	$MSEs_{ij}$	Rank	Candidate Well Log Variables	$MSEs_{ij}$	Rank
(CAL2, GR)	0.455055	1	(CAL2, CNTC)	0.497867	8
(CAL2, CFTC)	0.466716	2	(CAL2, NPSS)	0.521086	9
(CAL2, SP)	0.467368	3	(CAL2, NPDL)	0.523705	10
(CAL2, DPSS)	0.481365	4	(CAL2, NPLS)	0.525688	11
(CAL2, DPDL)	0.481389	5	(CAL2, AT90)	0.533946	12
(CAL2, RHOZ)	0.481453	6	(CAL2, AF90)	0.560741	13
(CAL2, DPLS)	0.481508	7			

Table 3–5. Fuzzy Surface (FS) ranking results and final selection result

Iter #	Ref Variable	Ranked Sequence by ascending $MSEs_{ij}$	Selected Variables	Eliminated Variables
2	GR	AT90,DPSS,DPDL,DPLS,RHOZ, CNTC,SP,CFTC, NPSS,NPDL,NPLS	CAL2,GR,AT90	AF90,NPLS
3	AT90	CFTC,CNTC,RHOZ, DPSS,DPDL,DPLS, NPSS,SP,NPDL	CAL2,GR,AT90, CFTC	AF90,NPLS,NPDL
4	CFTC	SP,DPDL,DPSS,DPLS, RHOZ,NPSS,CNTC	CAL2,GR,AT90, CFTC,SP	AF90,NPLS,NPDL, CNTC
5	SP	RHOZ,DPSS (SS),DPLS, DPDL,NPSS	CAL2,GR,AT90, CFTC,SP,RHOZ	AF90,NPLS,NPDL, CNTC,NPSS
6	RHOZ	DPDL,DPSS,DPLS	CAL2,GR,AT90, CFTC,SP,RHOZ, DPDL	AF90,NPLS,NPDL, CNTC,NPSS,DPLS
7	DPDL	DPSS	CAL2,GR,AT90,CFTC, SP,RHOZ, DPDL,DPSS	AF90,NPLS,NPDL, CNTC,NPSS,DPLS

Table 3–5 shows the results for the rest of the FS ranking iterations as well as the final selection result. FS stops after 7 iterations when only one variable is left in the candidate list. Finally, eight variables are selected, comprising 35% of the 21 the original well log variables. These form the final selection result.

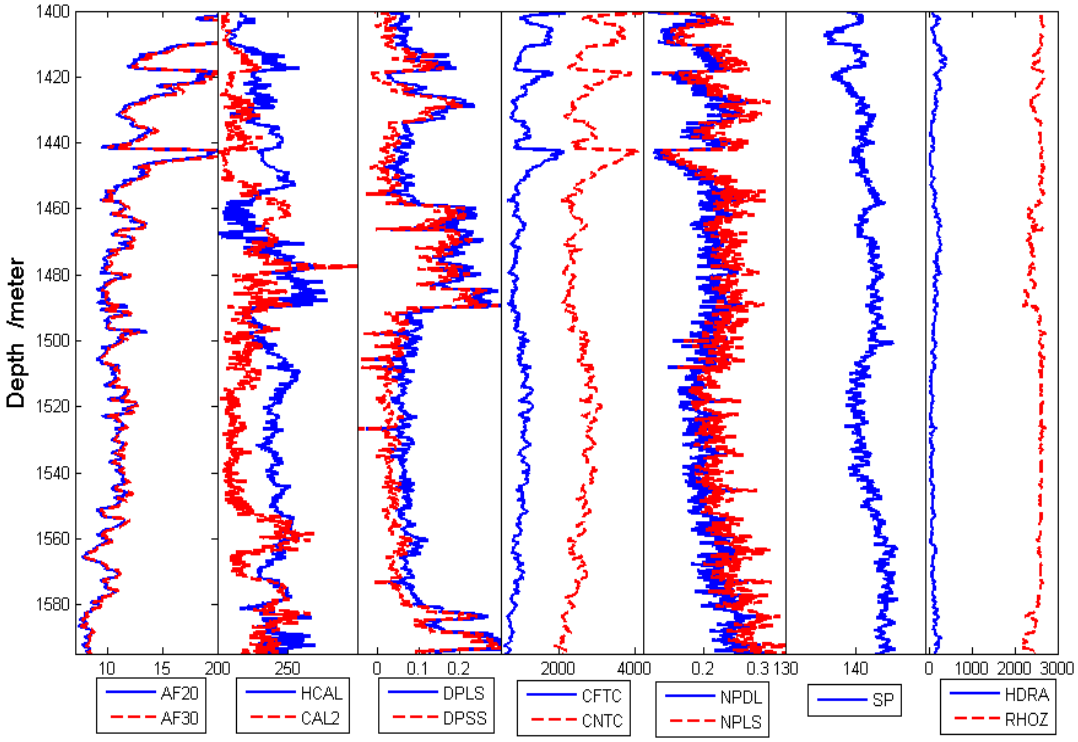

Figure 3–17 Well log variables versus depth.

FS achieves the removal of the highly correlated well log variables. Figure 3–17 shows well log variables versus depth from 1,400 to 1,600 meters below the surface for W-1, and the well log variables from the same category are paired together. Visual observation shows that the well log variables from the same category are highly correlated. For example, high correlation can be found for the pairs of DPLS / DPSS and CFTC / CNTC. After FS, DPLS has been removed from DPLS / DPSS and CNTC has

been removed from CFTC / CNTC. In addition, the eight selected well log variables cover all seven categories in Figure 3–17, which indicates that the seven categories all play roles in characterizing reservoir porosity. Also, the variables in Caliper Log, Gamma Ray and Sonic Log are always empirically selected when using statistical correlations for porosity characterization, and these variables are also ranked in the final result, which shows the success of fuzzy ranking in feature well log variable selection.

3.4.3.2 Reservoir Porosity Characterization

Eight well log variables from the previous fuzzy ranking step serve as neural inputs to MLP in the pattern recognition step, and porosity values in the core analysis data serve as the target data for MLP's supervised learning process. Data are randomly separated into 60%, 20% and 20% as training data, validation data and test data, respectively. The input layer is automatically initialized by the eight feature well log variables. Ten hidden neurons construct the hidden layer to MLP and the output layer includes only one neuron in terms of the target reservoir porosity. Transfer functions for the input, hidden and output layers are 'tan-sigmoid', 'tan-sigmoid' and 'log-sigmoid', respectively. The designed MLP is trained repeatedly for 10 times using the Levenberg-Marquart algorithm (Chen et al., 2003; Marquardt, 1964), and the final neural network is the one with the highest value of the correlation coefficient R^2 on the test data.

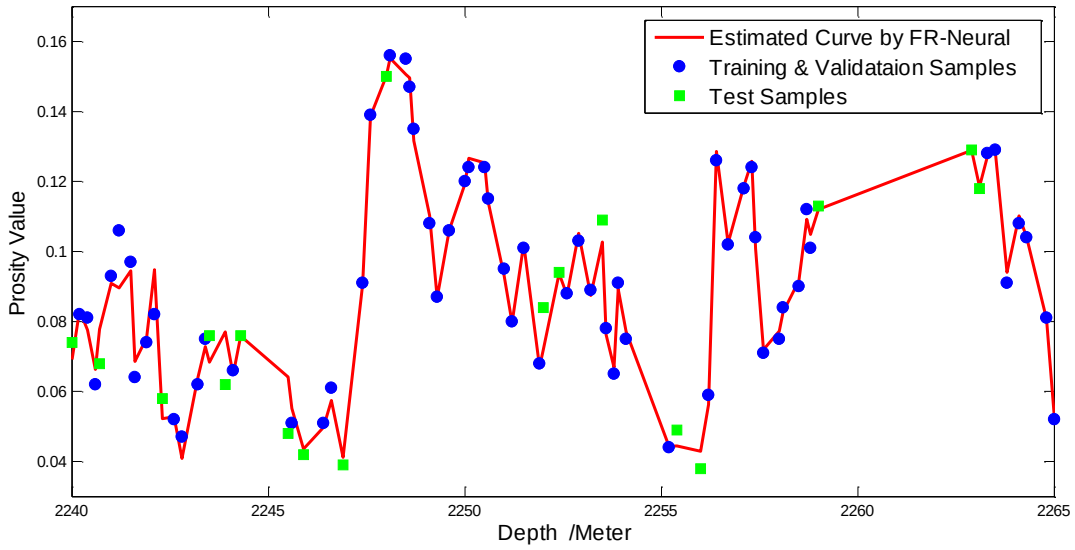

Figure 3–18 Estimated and core porosity values from MLP with depth

Figure 3–18 shows the final results for the porosity characterization from MLP using the well log variables selected by fuzzy ranking, where the round points denote the training and validation samples and the square points denote the test samples. In Figure 3–18, the curve is the estimated reservoir porosity from MLP. Almost all training, validation and test data match well with the estimated porosity curve. Figure 3–19 evaluates the results by the linear regression analysis, which shows that the Correlation Coefficient R^2 for the test data is 0.9504, which demonstrates that the proposed FR-Neural framework is capable of predicting accurate porosity values given unseen well log data.

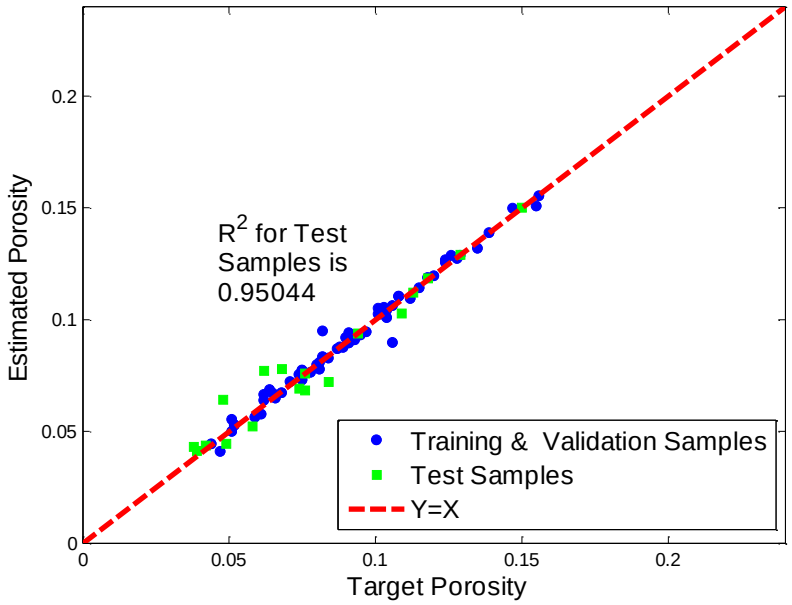

Figure 3–19 Cross plot of estimated and core porosity values

3.4.3.3 Performance Comparison

To evaluate the efficiency of representative well log data selection using fuzzy ranking, the reservoir characterization results from MLP using the fuzzy ranking results are compared with results using three other control neural inputs, namely: random selection, empirical selection and without selection. The first control group contains eight randomly selected well log variables, and the MLP network structure is 8-10-1, the same as for the one using fuzzy ranking. The second group uses the empirical selection result from (Helle et al., 2001) where three well log variables selected, respectively, from sonic, density and resistivity categories. Additionally, the MLP network structure is 3-7-1, which is consistent with the work in (Helle et al., 2001). For the third one, without selection all 21 well log variables from W-1 are imported as neural inputs, and the network structure is 21-14-1, which includes four extra hidden neurons due to the augmentation of the input

data. In order to make a consistent comparison, this experiment keeps the other configurations for MLP neural networks, including the training algorithm, transfer function and training parameters, the same. Four MLP neural networks with different inputs are repeatedly trained for 10 times and the results are shown in Figure 3–20.

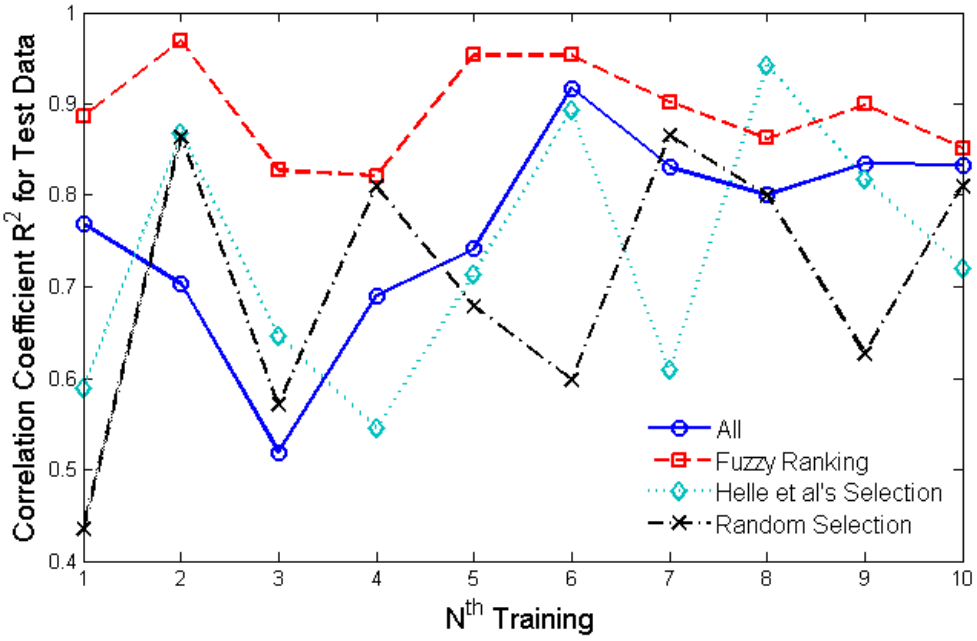

Figure 3–20 Comparison of R^2 on test samples from MLPs using different neural inputs

Figure 3–20 compares the predication accuracy of four MLPs. Table 3–6 summarizes the comparison result. Several observations can be made: First, the overall predication accuracy of MLP using the fuzzy ranking results surpasses the other three groups. As shown in Table 3–6, the average R^2 on the test samples for MLPs, using: fuzzy ranking, all well log variables, Helle et al's selection, and random selection, are 0.8746, 0.7639, 0.7344 and 0.7058, respectively. Second, the predication results of MLP using the fuzzy ranking are more stable since it has a lower chance to generate a very

poor predication result, e.g., the worst R^2 for the MLP using fuzzy ranking is 0.8037. Therefore, the elimination of well log variables in the fuzzy ranking step helps MLP in increasing the system stability and predication accuracy.

Table 3–6. Results comparison among MLPs using neural inputs from four different methods

Neural Input Selection Method	Fuzzy Ranking	Random Selection	Helle et al's Selection	All Well Log Variables
Best R^2	**0.9504**	0.8762	0.9437	0.9108
Avg R^2	**0.8746**	0.7058	0.7344	0.7639
Worst R^2	**0.8037**	0.4350	0.5451	0.5188

Table 3–7. Summarized results for three study wells

UWI	Selected/Original Well Log Variables	MLP Structure	Avg Train Time (sec)	Best R^2 for Test Samples
W-1	8/21	8-10-1	4.1462	0.9504
W-2	9/23	9-11-1	5.1892	0.9190
W-3	7/20	7-9-1	4.3908	0.9216

Table 3–7 summarizes the porosity characterization results for three case study wells using the proposed FR-Neural framework. The number of the selected well log variables by the fuzzy ranking step counts for approximately only 40% of the original well log variables, which also decreases the computation of MLP significantly. The MLP repeats the training processes 10 times for each well and R^2 for the test samples is calculated for each group. The overall R^2 on these test samples is above 0.9, which shows the prediction accuracy of the proposed FR-Neural reservoir characterization framework.

3.5 Summary

This chapter discusses reservoir characterization problems with focuses on core analysis and well log data. Section 3.1 introduces the background of reservoir characterization and reviews core analysis and well log data. Section 3.2 indicates that spatial entropy is a decreasing function with nonspatial similarity and spatial correlation. Further, a new spatial entropy-based spatial clustering algorithm, named SEClu, is proposed to group core analysis data. In Section 3.3, a new FR-Neural framework is presented to characterize reservoir properties using core analysis and well log data. This framework includes two steps: fuzzy ranking and pattern recognition. The Fuzzy ranking step selects the representative well log data with the objective to characterize a specific reservoir property. In the pattern recognition step, a MLP neural network learns the correlations between the selected well log and core analysis data, and predicts reservoir properties based only on well logs for depth intervals where core analysis data is absent. In Section 3.4, the proposed SEClu algorithm and the FR-Neural framework are evaluated. Experiments on both synthetic and real datasets show SEClu identifies meaningful clutters with spatial correlation patterns compared to GDBSCAN. In addition, the proposed FR-Neural framework is tested on a porosity characterization problem using datasets taken from three wells in Alberta, Canada. Results show that the FR-Neural framework predicts more accurate reservoir properties comparing to previous ANN-based reservoir characterization methods. Especially, the prediction accuracy of FR-Neural framework in reservoir characterization on three study wells reach to overall 85%.

Chapter Four: **Horizontal Well Placement Guideline Acquisition**

The following chapter first defines a variety of horizontal well placement attributes and formalizes the horizontal well placement guidance acquisition problem. Second, a new association rule mining algorithm, called SE-Apriori, is presented to efficiently solve the HWPGA problem. Third, a GIS system, named PetroData-GIS, containing the SE-Apriori tool is developed. Finally, the proposed method is evaluated using a real dataset taken from a SAGD project in Alberta.

4.1 Introduction

Horizontal well placement has a significant impact on enhanced oil recovery (EOR) recovery processes, such as the SAGD. Poor horizontal well placement negatively impacts the oil production rate, thermal efficiency and ultimate recovery rate (Chen et al., 1997). As discussed in Section 2.3, previous horizontal well placement planning methods relied only on simulated or predicted data, which have difficulty in providing satisfactory results, especially when the reservoir geological and geomechanical complexity increases. To our best knowledge, limited work has been conducted to investigate horizontal well placement plans using real field data.

The SAGD technology has been commercialized for over ten years, and a large amount of field data, including geological, drilling and production data, has been collected. These data contain implicit but interesting horizontal well placement patterns, such as under what placement conditions the oil production performance is satisfactory. By analyzing and presenting these patterns, it is very helpful for geologists and reservoir

engineers to understand the actual response of a reservoir to different well placement plans. The problem lies in how to efficiently sort through the related field data, identify the interesting horizontal well placement patterns and present them in a comprehensible way.

This chapter formalizes the HWPGA as a problem which *sorts through related SAGD field data and identifies interesting associations between horizontal well placement attributes and oil production performance*. Unlike previous methods providing horizontal well placement plans based on simulated data, HWPGA presents interesting well placement associations by analyzing real SAGD field data. The associations discovered in HWPGA are interesting since they reveal an implicit but strong influence from horizontal well placement attributes on oil production performance. In order to characterize the geological heterogeneity along the horizontal wells, a group of well placement attributes are defined. Meanwhile, the Steam-Oil-Ratio (SOR) is chosen to be the oil production performance indicator. SOR defines the ratio of the amount of steam required to produce a unit of oil. The smaller the SOR value, the better the oil production performance. Meanwhile, the discovered associations are presented in a rule format, which provides a straightforward way for users to understand.

Association Rule Mining (ARM) is introduced to solve a HWPGA problem by efficiently analyzing and presenting strong associations between horizontal well placement attributes and SOR. ARM requires the dataset to be in a transactional format in which each record contained is composed only of binary attributes. Therefore, quantitative and categorical attributes in the HWPGA dataset must be first transformed into binary attributes. Second, the computational cost of ARM is quite high since it is

required passing the entire database multiple times to prune a huge amount of candidate itemsets. In this work, two constraints are defined from the HWPGA problem, which narrow down the candidate itemsets search space. Based on the new constraints, a new ARM algorithm, named SE-Apriori, is introduced. Third, a Geographic Information System (GIS) containing the SE-Apriori tool is developed, which helps efficiently manage the field data from the petroleum industry and visualize the association rule mining results.

Subsequently, Section 4.2 defines 40 new horizontal well placement attributes. Next, Section 4.3 presents a new association rule mining algorithm by considering two constraints in the HWPGA problem. Section 4.4 then introduces the PetroData-GIS system. Finally, Section 4.5 evaluates the SE-Apriori using a real industrial dataset.

4.2 Horizontal Well Placement Characterization

In SAGD projects, horizontal well pairs are drilled with the “pad pattern”, in which multiple well pairs are drilled parallel into the reservoir with a pair space of 100 meters and a horizontal well length of 1,000 meters. Within each well pair, the distance between the upper and bottom wells is 5 meters on average. Such a pattern allows a large volume of steam to be delivered into the reservoir evenly. Even though the pad pattern is fixed, the vertical location of each horizontal well is not, which can influence the oil production performance. Thus, this work focuses on the vertical location of horizontal wells.

The vertical location of a horizontal well inside a heterogeneous reservoir can best be described as the relative distance between the horizontal well and different geological surfaces. Horizontal wells are drilled into a reservoir between different geological

surfaces. A geological surface is a summary of geological heterogeneity from rock properties like porosity, permeability and fluid saturation. For example, the Oil-Water-Contact (OWC) is defined as a geological surface with 80% water saturation; So50 is defined as a geological surface with 50% water saturation. Since the oil production performance of each well pair is significantly related to the reservoir's geology, incorporating geological surfaces in the well placement study is necessary. The distance between a horizontal well and a geological surface is retrieved by calculating the difference in elevation between them.

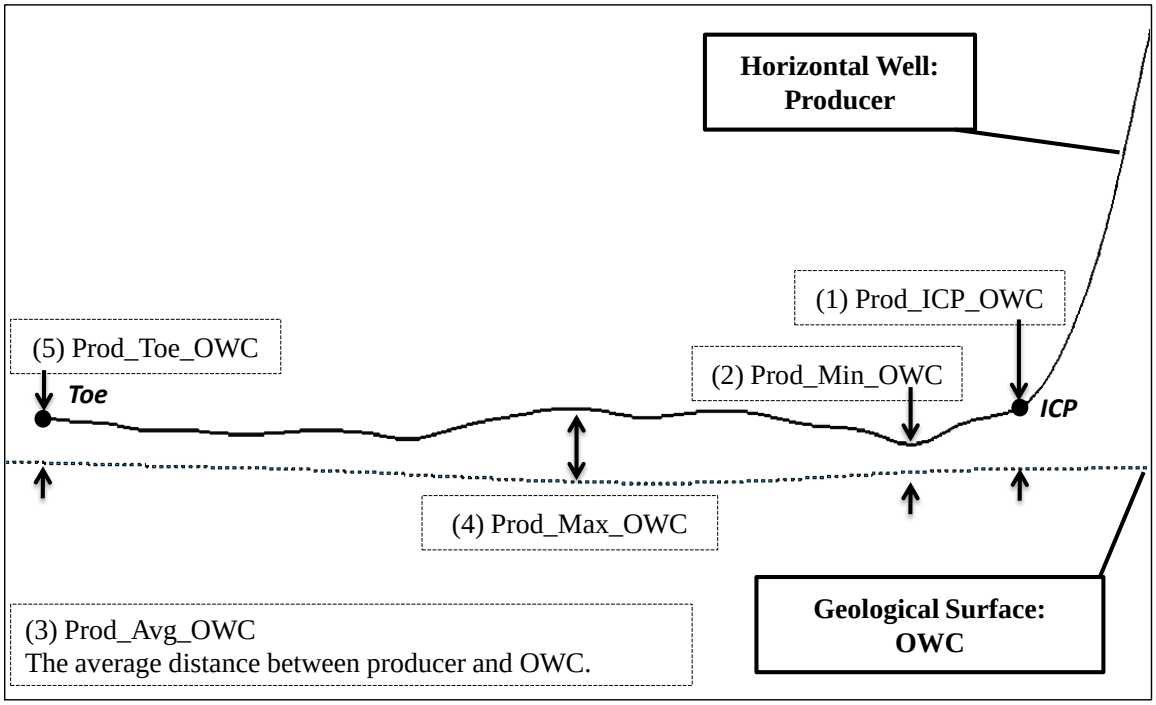

Figure 4–1 Five Well placement attributes from a horizontal producer and the Oil-Water-Contact (OWC) geological surface

In practice, both the horizontal wells and geological surfaces are not flat, as shown in Figure 4–1. In order to characterize the vertical movement between horizontal

wells and geological surfaces, it is necessary to sample some points when quantifying the relative distance between them. For each horizontal well, five attributes are retrieved to represent its relative distance to a geological surface, including: 1) minimum distance, 2) maximum distance, 3) average distance, 4) distance at the ICP point (ICP refers to the intermedium casing point which is the starting point of the horizontal well section), and 5) distance at the Toe point (Toe point refers the end point of a horizontal well). Taking the OWC as an example, Figure 4–1 shows the five attributes including Prod_ICP_OWC, Prod_Min_OWC, Prod_Avg_OWC, Prod_Max_OWC and Prod_Toe_OWC. Given that each SAGD well pair has two horizontal wells, i.e. injector and producer, there are 10 attributes defined to characterize the horizontal well placement in reference to a geological surface. For n geological surfaces of interest, the number of well placement attributes will be $10 \times n$.

4.3 Association Rule Mining in HWPGA with Constraints

The objective of association rule mining (ARM) in HWPGA is to analyze the SAGD field data and present interesting associations between horizontal well placement attributes and the oil production performance. ARM requires that the dataset to be in the transactional format in which each contained record contained is composed of binary attributes. Therefore, it is necessary to transfer the quantitative attributes in the HWPGA dataset to binary attributes. In the following, the partition and transformation processes are first presented. Second, two constraints in HWPGA and a formal problem description are given. Finally, a new ARM algorithm, named SE-Apriori algorithm, is introduced to efficiently mine the HWPGA dataset.

4.3.1 Data Transformation

There are essential differences between the transactional dataset and the dataset used in HWPGA (referred to as the HWPGA dataset). In the transactional dataset, each record is composed of binary attributes. For example, in the "market basket" transactional dataset, each item, like milk, bread or butter, can be modeled as a binary attribute. In each record, if the value of the attribute is "1," it means that the corresponding item is in the basket, and if it equals "0," it denotes that the corresponding item is not in the basket. Thus association rule mining in the "market basket" dataset can be viewed as finding frequent itemsets in the dataset where all the attributes are binary. However, attributes in a HWPGA dataset are either quantitative or categorical. As an illustration, Table 4–1 shows an example with two well placement attributes, Prod_Avg_OWC and Prod_Avg_So50, and the oil production performance indicator, SOR. Prod_Avg_OWC and Prod_Avg_So50 are quantitative attributes, where Prod_Avg_OWC is ranged between 5 meters and 9 meters. SOR is a categorical attribute, i.e. Poor or Good.

Table 4–1 Example of quantitative and categorical attributes in HWPGA dataset

Well Pair ID	Prod_Avg_OWC	Prod_Avg_So50	SOR
Well Pair 1	8.7 meters	3.8 meters	Poor
Well Pair 2	6.4 meters	3.2 meters	Poor
Well Pair 3	9.0 meters	3.2 meters	Poor
Well Pair 4	9.2 meters	2.1 meters	Good
Well Pair 5	8.1 meters	3.8 meters	Poor

In order to identify association rules from the HWPGA dataset, the quantitative and categorical attributes in HWPGA need to be transferred to binary attributes. A partition approach is used to solve this problem. For a categorical attribute, each

categorical value is transferred to a binary attribute. For each quantitative attribute, it is partitioned into consecutive intervals and each interval is further transferred to a binary attribute. For example, Table 4–1 shows the binary attributes transferred from the sample HWPGA dataset in Table 4–1. The quantitative attribute Prod_Avg_OWC is partitioned into {Prod_Avg_OWC: 5 m~8 m} and {Prod_Avg_OWC: 8 m~10 m}. A variety of discretization methods can be applied to partition the quantitative attributes (Han and Kamber, 2006; Zhang et al., 2003). The categorical attribute SOR is partitioned into {SOR: Poor} and {SOR: Good}. After the data transformation, the HWPGA dataset is ready to be solved with the ARM algorithms.

Table 4–2 The quantitative and categorical attributes in Table 4–1 are partitioned and transferred into binary attributes.

Well Pair ID	Prod_Avg_OWC (meters)		Prod_Avg_So50 (meters)		SOR :	
	(6~8)	(8~10)	(2~3)	(3~4)	Poor	Good
Well Pair 1	1	0	0	1	1	0
Well Pair 2	1	0	0	1	1	0
Well Pair 3	0	1	0	1	1	0
Well Pair 4	0	1	1	0	0	1
Well Pair 5	0	1	0	1	1	0

4.3.2 Constraints in Association Rule Mining of HWPGA Dataset

An association rule discovered in the HWPGA problem should represent the influence of horizontal well placement attributes on the oil production performance. This suggests that the well placement attributes should be the antecedents of a rule, while the oil production performance indicator is the consequence, as defined in Eq. (4.1). However, association

rules from classic ARM methods, such as Apriori, do not have the antecedent and consequence constraints. For example, with a frequent 2-itemset {Prod_Avg_OWC, SOR}, the generated rules could be {Prod_Avg_OWC}=>{SOR} or {SOR}=>{Prod_Avg_OWC}. Based on Eq. (4.1), the second rule is invalid.

$$\{\text{Well Placement Attributes}\} \Rightarrow \{\text{Oil Production Performance}\} \quad (4.1)$$

Even though the association rules resulting from Apriori can be filtered by a post-processing step, it is much more efficient to incorporate the constraints into the association rule mining process. This work considers the HWPGA constraints in the association rule mining process, and proposes a new Apriori algorithm, named SE-Apriori. Next, a formal problem description and the SE-Apriori algorithm are introduced.

A HWPGA dataset D contains N well pair records. Each record R contains s well placement attributes $\{w_1,\ldots,w_i,\ldots,w_s\}$ and one oil production performance indicator P. Denote $W=\{w_1,\ldots,w_i,\ldots,w_s\}$, and each record R is in the form of $<W,P>$. After the data transformation, each quantitative attribute w_i is transformed into m binary attributes $\{w_{i1},\ldots,w_{ij},\ldots,w_{im}\}$, and P is transformed into n binary attributes $\{p_1,\ldots,p_k,\ldots p_n\}$. Here, each w_{ij} is called a *child* of w_i, and p_k is a child of P. Meanwhile, w_i and P are called parent attributes. Denote all binary children attributes from P by $\tilde{P}$ and all children attributes from W by $\tilde{W}$, $\tilde{P}=\{p_1,\ldots,p_k,\ldots p_n\}$ and $\tilde{W}=\{w_{11},\ldots,w_{1j},\ldots,w_{1m};\ldots;w_{i1},\ldots,w_{ij},\ldots,w_{im};\ldots;w_{s1},\ldots,w_{sj},\ldots,w_{sm}\}$. Thus, after the data transformation each record is in the form of $<\tilde{W},\tilde{P}>$.

An association rule discovered in a HWPGA dataset is with the form $\tilde{w} \Rightarrow \tilde{p}$, where $\tilde{w} \subset \tilde{W}$ and $\tilde{p} \subset \tilde{P}$. The rule $\tilde{w} \Rightarrow \tilde{p}$ holds in dataset *D* with the ***support*** equal to the percentage of records that consist of $\tilde{w} \bigcup \tilde{p}$ to the total number of records. The rule $\tilde{w} \Rightarrow \tilde{p}$ has the ***confidence*** equalling the percentage of records consisting of $\tilde{w} \bigcup \tilde{p}$ to the number of records that consist of $\tilde{w}$. Given a HWPGA dataset *D*, the objective of ARM is to find all association rules that have support and confidence not smaller than the predefined minimum support (*minsup*) and minimum confidence (*minconf*), respectively.

The ARM includes two steps: (1) finding all frequent itemsets and (2) generating association rules from frequent itemsets. The first step consumes a very high computational cost since it requires passing through the whole dataset multiple times to prune candidate itemsets. In addition, the second step may generate a huge amount of association rules and many of them may not be meaningful. In order to solve these two problems, two constraints, namely *selective constraint* and *exclusive constraint*, are defined to improve the efficiency of the association rule mining process in HWPGA.

- **Definition 4.1 (selective constraint):** Each frequent *k*-itemset discovered in a HWPGA dataset, $k \geq 2$, must contain both $\tilde{w}$ and $\tilde{p}$, where $\tilde{w} \subset \tilde{W}$ and $\tilde{p} \subset \tilde{P}$.

Each frequent *k*-itemset, $k \geq 2$, must contain w_{ij} and p_k at the same time. In order to generate association rules with the form $\tilde{w} \Rightarrow \tilde{p}$, the antecedent $\tilde{w}$ and consequence $\tilde{p}$ must exist in the frequent itemset at the same time. For the frequent itemsets containing only a subset of w_{ij}, such as {Prod_Avg_OWC: 6~8 m, Prod_Avg_So50: 3~4 m}, it is impossible to generate valid association rules with the SOR appearing in the rule

consequence. Similarly, it is also impossible to generate valid association rules from frequent itemsets containing only a subset of p_k. Thus, to avoid unnecessary computation, it is necessary to remove the candidate itemsets containing only a subset of w_{ij} or p_k.

(a) Dataset After Transformation

Well Pair ID	Prod_Avg_OWC	Prod_Avg_So50	Sor
Well Pair 1	6~8 m	3~4 m	Poor
Well Pair 2	6~8 m	3~4 m	Poor
Well Pair 3	8~10 m	3~4 m	Poor
Well Pair 4	8~10 m	2~3 m	Good
Well Pair 5	8~10m	3~4 m	Poor

(b) Frequent 1-Itemsets

Frequent 1-Itemsets (*minsup*=40%)	Support
{Prod_Avg_OWC: 6~8 m}	40%
{Prod_Avg_OWC:8~10 m}	60%
{Prod_Avg_So50: 3~4 m}	80%
{Sor: Poor}	80%

(c) Candidate2-Itemsets by ARMHC

Candidate 2-Itemsets	
{Prod_Avg_OWC: 6~8 m},{Prod_Avg_OWC: 8~10 m}	Invalid
{Prod_Avg_OWC: 6~8 m}, {Prod_Avg_So50: 3~4 m}	Invalid
{Prod_Avg_OWC:8~10 m}, {Prod_Avg_So50: 3~4 m}	Invalid
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m}	**Valid**
{Sor: Poor}, {Prod_Avg_OWC:8~10 m}	**Valid**
{Sor: Poor}, {Prod_Avg_So50: 3~4 m}	**Valid**

(d) Frequent 2-Itemsets

Frequent 2-Itemsets (*minsup*=40%)	Support
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m}	40%
{Sor: Poor}, {Prod_Avg_OWC:8~10 m}	40%
{Sor: Poor}, {Prod_Avg_So50: 3~4 m}	80%

(e) Candidate 3-Itemsets

Candidate 3-Itemsets	
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m},{Prod_Avg_OWC:8~10 m}	Invalid
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m}, {Prod_Avg_So50: 3~4 m}	**Valid**
{Sor: Poor}, {Prod_Avg_OWC:8~10 m}, Prod_Avg_So50: 3~4 m}	**Valid**

(f) Frequent 3-Itemsets

Frequent 3-Itemsets (*minsup*=40%)	Support
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m}, {Prod_Avg_So50: 3~4 m}	40%
{Sor: Poor}, {Prod_Avg_OWC:8~10 m},{Prod_Avg_So50: 3~4 m}	40%

Figure 4–2 An example of frequent itemsets generation with the selective and exclusive constraints.

Figure 4–2 shows an example of the frequent itemset generation process using the sample HWPGA dataset in Table 4–1. Here, Prod_Avg_OWC and Prod_Avg_So50 are two well placement attributes from *W*. SOR is the oil production performance indicator from *P*. After data transformation, the dataset is shown in Figure 4–2 (a). By scanning the dataset, it is easy to determine the frequent 1-itemsets shown in Figure 4–2 (b). Figure 4–2 (c) shows all candidate 2-itemsets obtained using the Apriori-generation function in Figure 2–2. However, given the selective constraint, the former three candidate itemsets are invalid due to the nonexistence of p_k. Thus 3 out of 6 candidate 2-itemsets are removed, which narrows the candidate itemsets searching space by 50%.

- **Definition 4.2 (exclusive constraint):** In each frequent *k*-itemset discovered in a HWPGA dataset, $k \geq 2$, children attributes derived from the same parent are exclusive to each other and only one can appear.

The exclusive constraint is applicable to both $<w_{i1},\ldots, w_{ij}, \ldots, w_{im}>$ derived from w_i and $<p_1,\ldots, p_k, \ldots p_n>$ derived from *P*. The definition of the exclusive constraint is based on the observation that each quantitative attribute value or categorical value can only be transformed into one binary attribute. Therefore, after data transformation, for each record *R* in *D*, there is no chance that more than one child binary attribute derived from the same parent will be equal to "1" at the same time. Therefore, for the candidate itemsets, such as $<p_2, w_{i2}, w_{i3}>$, their support will always be zero. Removing such candidate itemsets can narrow the search space and accelerate the frequent itemset generation process. Note that, from the exclusive constraint, it is easy to conclude that the highest order of the frequent itemises is $s+1$.

Continuing the previous example in Figure 4–2, three frequent 3-itemsets generated by the Apriori-generation function are shown in Figure 4–2 (e). Based on the exclusive constraint, the first itemset is invalid due to the cooccurrence of two child attributes derived from the same parent, {Prod_Agv_OWC:6~8} and {Prod_Agv_OWC:8~10}, and should be removed. Next, when the frequent 3-itemsets are found, the frequent itemsets generation process stops since the highest frequent itemsets have been reached. Hence, the introduction of the exclusive constraint decreases the candidate 3-itemsets searching space by 1/3 and ends the frequent itemsets generation process with an early stop.

In addition, Eq. (4.1) constrains the number of generated rules. After finding all the valid frequent itemsets, each frequent *k*-itemset *f* contains *k*-1 well placement attributes and one oil production performance indicator. To generate association rules satisfying Eq.(4.1), the oil production performance indicator p_k is selected out from *f* and an association rule is generated as (f - p_k) => p_k. This process automatically prunes the invalid association rules and leads to a small number of association rules.

Frequent 3-Itemset
{Sor: Poor}, {Prod_Avg_OWC: 6~8 m}, {Prod_Avg_So50: 3~4 m}

Generated Rules (*minconf*=60%)	**Support**	**Confidence**	
{Sor: Poor} AND {Prod_Avg_OWC: 6~8 m} => {Prod_Avg_So50: 3~4 m}	40%	100%	Invalid
{Sor: Poor} AND {Prod_Avg_So50: 3~4 m} => {Prod_Avg_OWC: 6~8 m}	40%	100%	Invalid
{Prod_Avg_OWC:8~10 m} AND {Prod_Avg_So50: 3~4 m} => {Sor: Poor}	**40%**	**100%**	**Valid**

Figure 4–3 Example of the association rules generation

4.3.3 SE-Apriori Algorithm

Given a HWPGA dataset, the association rule mining finds all valid association rules defined in Eq. (4.1) by satisfying the *minsup* and *minconf* thresholds. In practice, the the *minsup* and *minconf* parameters are determined by the users. In the following, a new Apriori-based association rule mining algorithm considering the selective and exclusive constraints, named SE-Apriori, is introduced.

The SE-Apriori includes two parts, SE-AprioriGen and SE-AprioriRule. Considering the selective and exclusive constraints, SE-AprioriGen finds all valid frequent itemsets from a HWPGA dataset with the support no less than the *minsup*. Furthermore, SE-AprioriRule generates association rules from the frequent itemsets with the confidence no less than the *minconf* threshold.

Figure 4–4 shows the pseudo code of the SE-AprioriGen algorithm. The algorithm requires passing the dataset D for at most s times to determine all the frequent itemsets. The first pass simply counts the cooccurrences of $<p_k, w_{ij}>$, where $1 \le k \le n$, $1 \le i \le s$ and $1 \le j \le m$ to determine the frequent 2-itemsets F_2. The subsequent pass consists of two steps. First, the frequent itemsets F_k are used to generate the candidate itemsets C_{k+1} using a new SE-Candidate function shown in Figure 4–5. Second, the SE-AprioriGen algorithm passes the datasets to calculate the support to each candidate itemset in C_{k+1} and the ones with support not less than *minsup* are inserted into F_{k+1}. Note that the Apriori Property is conserved in the SE-AprioriGen Algorithm since each step C_{k+1} is generated from a lower level itemset F_k. The SE-AprioriGen stops until F_k is empty or the s+1 level of the frequent itemsets has been reached.

Algorithm: SE-AprioriGen (*D*, *minsup*)

Input: (1) A HWPGA dataset *D* (2) minimum support *minsup*

Output: Frequent itemsets $\bigcup_k F_k$

01: Let F_2={frequent 2-itemsets};

02: Let *k*=2 **and** *maxlevel*=*s*+1; // the maximum level can't exceed *s*+1

03: **while** ($F_k \neq \varnothing$ **and** $k \leq maxlevel$)

/* a new candidate generation function considering constrains in HWPGA */

04: C_{k+1}=SE-Candidate (F_k);

05: Scan *D* to determine the support to each candidate $c \in C_{k+1}$

07: F_{k+1}={ $c \in C_{k+1} \mid c.support \geq minsup$} ;

08: *k*++;

09: **end** // end while

10: **return** $\bigcup_\nu F_\nu$;

Figure 4–4 Pseudo code of SE-**AprioriGen algorithm**

The SE-Candidate function, shown in Figure 4–5, takes an argument of the frequent *k*-itemsets F_k and returns all candidate (*k*+1)-itemsets satisfying the selective and exclusive constraints. It assumes that each frequent *k*-itemset is ordered in the form <*p*, w^1, w^2,…, w^{k-1}>, where $\{w^1, w^2, \ldots, w^{k-1}\}$ are ordered lexicographically and the sequence is conserved by the superscript. To find C_{k+1}, F_k is joined with itself. For each pair *k*-itemsets f_1 and f_2 belong to F_k, they can be merged into a candidate (*k*+1)-itemsets only if

their first k-1 items are the same and the last item is derived from different parent attributes. In line 3 of Figure 4–5, $f_1.w^{k-1} < f_2.w^{k-1}$ denotes $f_1.w^{k-1}$ stands in a former place of $f_2.w^{k-1}$ lexicographically. $f_1.w^{k-1}.parent \neq f_2.w^{k-1}.parent$ ensures that the exclusive constraint is satisfied. The resulting candidate (k+1)-itemsets by joining f_1 and f_2 is $< f_1.p,\ f_1.w^1, \ldots,\ f_1.w^{k-2}, f_1.w^{k-1}, f_2.w^{k-1} >$.

Function: SE-Candidate (F_k)

Input: Frequent k-itemsets: F_k, $k \geq 2$

Output: Candidate k-itemsets C_{k+1}

01: **foreach** itemset $f_1 \in F_k$

02: **foreach** itemset $f_2 \in F_k$

/* generate candidate itemsets considering selective and exclusive constraints */

03: **if** ($f_1.p = f_2.p$ **and** $f_1.w^1 = f_2.w^1$ **and** … **and** $f_1.w^{k-2} = f_2.w^{k-2}$ **and**

$f_1.w^{k-1} < f_2.w^{k-1}$ **and** $f_1.w^{k-1}.parent \neq f_2.w^{k-1}.parent$)

04: { $c = merge(f_1, f_2)$; and Add c into C_{k+1}; }

05: **end** // end if

06: **end** // end foreach

07: **end** // end foreach

08: **return** C_{k+1};

Figure 4–5 Pseudo code of the SE-**Candidate generation function**

After finding all valid frequent itemsets, the generation of association rules is straightforward. Figure 4–6 shows the pseudo code of the SE-AprioriRule algorithm. To generate rules complying with Eq. (4.1), each frequent itemset *f* is enumerated and a rule is built as (*f*-*f.p*) => *f.p*. This rule is inserted into the results only if its confidence satisfies the predefined *minconf* threshold.

Algorithm: SE-AprioriRule (*F*, *minconf*)

Input: (1) All frequent *k*-itemsets *F*, $k \geq 2$ (2) minimum confidence *minconf*

Output: Generated association rules

01: **foreach** frequent itemset $f \in F$

/* The rule antecedence can only be $f.w$ and the consequence can only be $f.p$ */

02: Let A={ $f.w^1$, $f.w^2$,…, $f.w^{k-1}$ } and B= $f.p$;

03: *conf*=support(f) / support (A);

04: **if** ($conf \geq minconf$)

05: Add to *R*: A=> B, sup=support(f) and conf=*conf*;

06: **end** // end if

07: **end** // end foreach

08: **return** *R*;

Figure 4–6 Pseudo code of the SE-**AprioriRule algorithm**

4.3.4 Complexity Analysis

The computational cost of SE-Apriori is analyzed and compared with Apriori. Using the problem definition in Section 4.3, the computational cost of SE-Apriori is deduced in the

following. Each record in a HWPGA dataset contains s quantitative well placement attributes $\{w_1,\ldots,w_i,\ldots,w_s\}$ and one categorical oil production performance indicator P. After data transformation, each w_i has m children binary attributes and P has n children binary attributes. Thus the total possible combination for each record is nm^s, and for all N records, this number is Nnm^s. The SE-Apriori needs to pass the dataset for at most s+1 to find all frequent itemsets. Thus the computational cost of SE-Apriori is given in Eq. (4.2).

$$O\left(N\left(\mathrm{s}+1\right)nm^s\right) \tag{4.2}$$

As for Apriori, the computational cost, as suggested by (Agrawal et al., 1993), is given in Eq. (4.3).

$$O\left(NM2^M\right) \tag{4.3}$$

where N is the number of records in D, M is the number of the total binary attributes, and m is the number of intervals in the partition process. Considering $sm+n=M$, $m>1$, $n>1$, the following deduction can be made from Eq.(4.2):

$$O\left(N\left(\mathrm{s}+1\right)nm^s\right) < O\left(NM2^{\log_2 nm^s}\right) = O\left(NM2^{(s\log_2 m+\log_2 n)}\right) < O\left(NM2^M\right)$$

Thus the computational cost of SE-Apriori is much lower than that of Apriori in solving a HWPGA problem.

Meanwhile, SE-Apriori generates results with a smaller number of rules in HWPGA compared to Apriori. With the Apriori algorithm, the number of possible rules suggested by (Agrawal et al., 1993) is $M2^{M-1}$. In SE-Apriori, the highest order of frequent itemset in HWPGA problem is s+1 due to the exclusive constraints. For each partition p_k of P, the maximum number of frequent itemsets containing it is m^s, thus the

total number of potential frequent itemsets is nm^s. Since each frequent itemset can only generate one association rule, the total possible number of rules is given in Eq. (4.4).

$$nm^s \tag{4.4}$$

Considering $sm+n=M$, $m>1$, $n>1$, the following deduction can be made from Eq.(4.4):

$$nm^s = 2^{\log_2 nm^s} = 2^{(s\log_2 m+\log_2 n)} < 2^{sm+n} = 2^M$$

Therefore, SE-Apriori generates results with a smaller number of rules in HWPGA compared to Apriori.

4.4 PetroData-GIS System Prototype

At present, various researches have been conducted on applying data mining to pick up meaningful patterns from the field data in order to increase oil production or decrease operational costs. In the meantime, a growing number of oil and gas companies have implemented Geographic Information Systems (GIS) to manage the large volume of field data. With the growing number of field data that have been geographically referenced, combining data mining and GIS shows high potential in efficient data management and visualizing the data mining results.

Integrating data mining into GIS, this thesis develops a system prototype, called PetroData-GIS. First, PetroData-GIS manages large amounts of field data from petroleum wells in a spatial database and visualizes the geospatial information on a 2D map. Second, the data mining methods are designed as the analysis tools in PetroData-GIS. By connecting to the database, they help in generating interesting patterns by sorting through large quantities of field data. Finally, the data mining results can be called back by PetroData-GIS and visualized on the map, which provides a user friendly interface. As an

example, the PetroData-GIS prototype is demonstrated in the a HWPGA problem by visualizing the association rules generated from the SE-Apriori.

In the following, the architecture of PetroData-GIS prototype and different components in the architecture are introduced.

4.4.1 PetroData-GIS Prototype Architecture

Figure 4–7 shows the architecture of the PetroData-GIS prototype. It has three main components: a spatial database, GIS functions and the graphical user interface (GUI). The GUI and the GIS functions are developed using C# programming language with integration of the ESRI ArcObjects. The PetroData-GIS prototype supports diverse data formats (*.mxd map file, *.lyr layer file, *.shp shape file, *.mdb geodatabase file) and visualizes them on a map. The following section describes the main components of the prototype in detail.

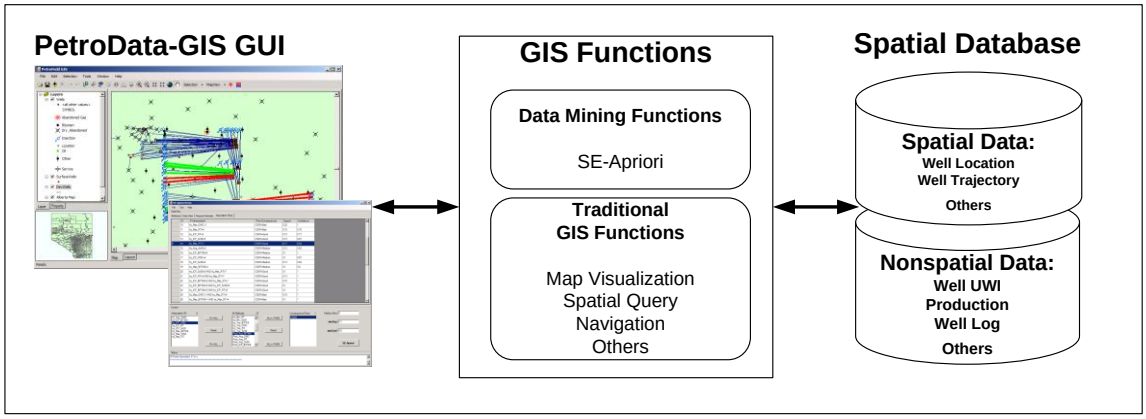

Figure 4–7 Architecture of the PetroData-GIS prototype

Spatial Database: The spatial database in PetroData-GIS prototype is implemented with Microsoft SQL Server 2005 and the ArcObject's GeoDatabase Library. The spatial database contains both the spatial and nonspatial data. The spatial

data stores the well locations and spatial objects of wells, i.e., points or polylines. Vertical wells are stored as spatial points while deviated wells are saved as polylines. The nonspatial data records the field data including the unique well identification (UWI), well ticket information, core analysis data, well logs and production data. The UWI is unique to each well and is used as the primary key in the database. The well ticket information records general information such as the well type, drilling and recovery dates. The nonspatial data is connected with the spatial data using the primary key of UWI.

GIS Functions: The GIS functions in the PetroData-GIS Prototype can be classified into two groups: the traditional GIS functions and data mining functions. The traditional GIS functions include map visualization, spatial query and simple spatial analysis tools. For example, the spatial query allows users to select wells on the map, which automatically retrieves the related field data from the database. All of these functions are implemented by calling the APIs from ESRI ArcObjects.

Apart from the traditional GIS functions, the current version of PetroData-GIS provides the SE-Apriori tool as the data mining function. The user can assign a group of attributes from selected wells to the SE-Apriori analysis. After transforming the data into the predefined format, SE-Apriori analyzes and presents the discovered association rules among the assigned attributes. In addition, the discovered association rules can be visualized on the map. For each discovered association rule X=>Y, the system will query the database to find the records satisfying both the antecedent X and the consequence Y, and the ones satisfying only the antecedent X. For example, given *minsup*=40% and *minconf*=60%, a sample rule generated from the HWPGA dataset in Table 4–1 is listed in Figure 4–8. By scanning the dataset, 2 records satisfying this rule and one record

satisfying only the antecedent can be identified. Figure 4–8 shows the two types of records related to this rule. With the GIS functions, the PetroData-GIS can highlight wells belonging to these two types separately on the map, which helps users in understanding the rule by linking the association with related wells.

Sample Rule	Sup	Conf
{Prod_Avg_OWC:8~10 m} And {Prod_Avg_So50: 3~4 m} => {SOR: Poor}	40%	66.6%

Two Types of Records Related to A Rule	Records
Type A: Satisfy both the antecedent X and the consequence Y	Well Pair 1 Well Pair 2
Type B: Satisfy only the antecedent X but not the consequence Y	Well Pair 5

Figure 4–8 Two types of records related to the sample rule

PetroData-GIS GUI: Figure 4–9 shows the main graphical user interface (GUI) of the PetroData-GIS system prototype. In the middle of the interface is the map display area in which the visualized petroleum well map is shown with the designated map scale and coordinates. On the left of the interface there is a layer table showing the map layers and an eagle eye window showing a global view of the current map. The top of the interface contains the menu and the tool bar from which the user can access different GIS functions.

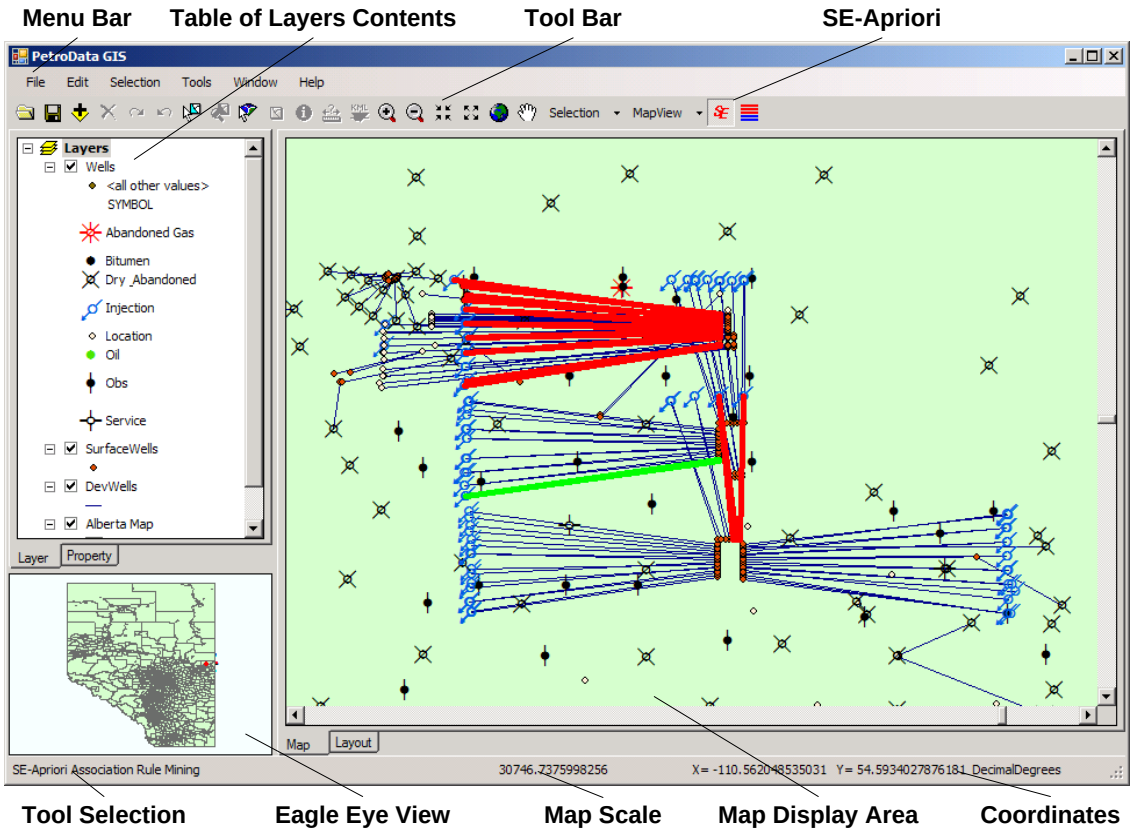

Figure 4–9 The main graphical user interface of PetroData-GIS prototype

In addition, through the interface the user can perform the association rule mining via the SE-Apriori tool. After assigning a group of interest attributes, the user can run the association rule mining by clicking the "SE" button in the tool bar. Figure 4–10 shows the GUI of the SE-Apriori tool. From top to bottom, there are the Menu Bar, Data & Result Viewer, Control Panel and Message Box. The user can constrain the antecedent and the consequence of a rule and specify the *minsup* and *minconf* parameters. SE-Apriori returns all the association rules from the predefined antecedent attributes to the consequence by satisfying the *minsup* and *minconf* thresholds. All of the resulting association rules will be listed in the Data & Result Viewer. By clicking each individual

rule, the wells related to this rule can be highlighted on the map. In the following section, a case study is discussed using the SE-Apriori.

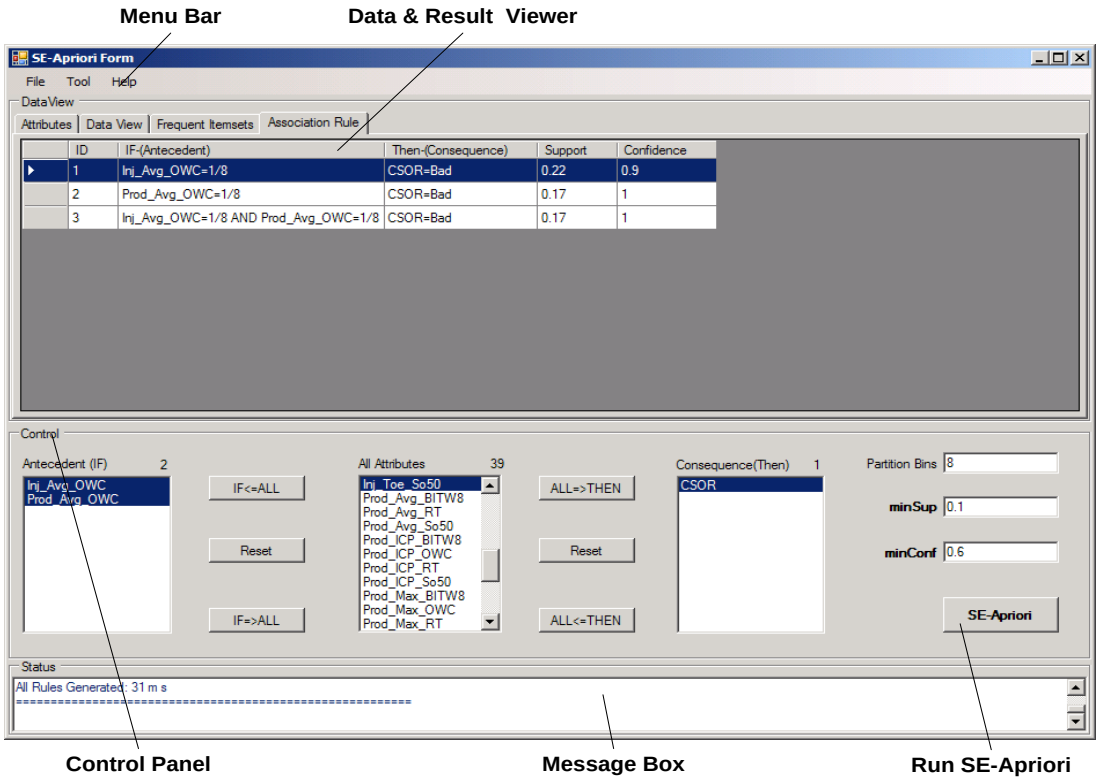

Figure 4–10 The graphical user interface of the SE-Apriori tool

4.5 A Case Study

This section demonstrates the SE-Apriori in a HWPGA problem using the real SAGD field data. It starts by describing the data collection and preprocessing. Also, the efficiency of the SE-Apriori algorithm is compared with Apriori. Furthermore, example association rules discovered using SE-Apriori are given.

4.5.1 Data Collection and Preprocessing

The proposed SE-Apriori algorithm is demonstrated using the real SAGD field data taken from a SAGD project located in northern Alberta, Canada. The study dataset contains 43 SAGD well pairs. Each well pair was drilled into four geological surfaces, i.e., RT, So50, BITW8 and OWC, as shown in Table 4–3. The horizontal well placement characterization is made by retrieving the relative distance between horizontal wells and four geological surfaces. For each well pair, 40 well placement attributes, as described in Table 4–4, are retrieved. In addition, the oil production performance for each well pair is labelled as 'Good', 'Fair' and 'Poor' based on their SOR value by the reservoir engineers. Hence each record in the study dataset contains 40 well placement attributes and 1 oil production performance indicator. In the data transformation process, each well placement attribute is evenly partitioned into eight intervals with each interval being transformed into a binary attribute. The SOR is transformed into three binary attributes, SOR: Good, SOR: Fair and SOR: Poor, based on its category. Meanwhile, due to data confidentiality, a labelling method is introduced to conceal the real data. For example, the label "Prod_Avg_OWC=2/8" denotes that the value of Prod_Avg_OWC has been partitioned into 8 intervals and it belongs to 2/8 of the intervals.

Table 4–3 Geological surfaces used in the thesis

Abbreviations	Description
RT	Reservoir Top
So50	50% Oil Saturation Surface.
BITW8	8% Bitumen Weight Surface.
OWC	80% Water Saturation Surface.

Table 4–4 Description of the 40 horizontal well placement attributes

Abbreviation	Group (count)	Description
Prod_ICP_RT	RT 10	distance between the producer **ICP** and RT
Prod_Min_RT		**minimum** distance between the producer and RT
Prod_Avg_RT		**average** distance between the producer and RT
Prod_Max_RT		**maximum** distance between the producer and RT
Prod_Toe_RT		distance between the producer **Toe** and RT
Inj_ICP_RT		distance between the injector **ICP** and RT
Inj_Min_RT		**minimum** distance between the injector and RT
Inj_Avg_RT		**average** distance between the injector and RT
Inj_Max_RT		**maximum** distance between the injector and RT
Inj_Toe_RT		distance between the injector **Toe** and RT
Prod_ICP_BITW8	BITW8 10	distance between the producer **ICP** and BITW8
Prod_Min_BITW8		**minimum** distance between the producer and BITW8
Prod_Avg_BITW8		**average** distance between the producer and BITW8
Prod_Max_BITW8		**maximum** distance between the producer and BITW8
Prod_Toe_BITW8		distance between the producer **Toe** and BITW8
Inj_ICP_BITW8		distance between the injector **ICP** and BITW8
Inj_Min_BITW8		**minimum** distance between the injector and BITW8
Inj_Avg_BITW8		**average** distance between the injector and BITW8
Inj_Max_BITW8		**maximum** distance between the injector and BITW8
Inj_Toe_BITW8		distance between the injector **Toe** and BITW8
Prod_ICP_So50	So50 10	distance between the producer **ICP** and SO50
Prod_Min_So50		**minimum** distance between the producer and SO50
Prod_Avg_So50		**average** distance between the producer and SO50
Prod_Max_So50		**maximum** distance between the producer and SO50
Prod_Toe_So50		distance between the producer **Toe** and SO50
Inj_ICP_So50		distance between the injector **ICP** and SO50
Inj_Min_So50		**minimum** distance between the injector and SO50
Inj_Avg_So50		**average** distance between the injector and SO50
Inj_Max_So50		**maximum** distance between the injector and SO50
Inj_Toe_So50		distance between the injector **Toe** and SO50
Prod_ICP_OWC	OWC 10	distance between the producer **ICP** and OWC
Prod_Min_OWC		**minimum** distance between the producer and OWC
Prod_Avg_OWC		**average** distance between the producer and OWC
Prod_Max_OWC		**maximum** distance between the producer and OWC
Prod_Toe_OWC		distance between the producer **Toe** and OWC
Inj_ICP_OWC		distance between the injector **ICP** and OWC
Inj_Min_OWC		**minimum** distance between the injector and OWC
Inj_Avg_OWC		**average** distance between the injector and OWC
Inj_Max_OWC		**maximum** distance between the injector and OWC
Inj_Toe_OWC		distance between the injector **Toe** and OWC

4.5.2 Association Rule Mining with SE-Apriori

After preprocessing, the study dataset is ready for the ARM. In the following, the efficiency of the SE-Apriori in HWPGA is compared with Apriori using the study dataset. The SE-Apriori is programmed with Microsoft C# based on .net framework 3.5. The Apriori is implemented in Weka (Weka, 2012), a third-party data mining software. All experiments are performed on a 2.8 GHz PC with 3 GB memory.

4.5.2.1 Computational Time

This experiment compares the computational cost between SE-Apriori and Apriori with varying *minsup* values. Most of the computational cost of association rule mining comes from finding the frequent itemsets.

Figure 4–11 shows the comparison of computational time between SE-Apriori and Apriori with varying *minsup* values from 0.1 to 0.18. From Figure 4–11, two observations can be made: First, SE-Apriori requires less computational time compared to Apriori. Table 3 lists the computational times of SE-Apriori and Apriori with different *minsup* values. For example, when *minsup* was set to 10%, SE-Apriori executed 3 seconds while Apriori run for 23 seconds. Second, when *minsup* was decreased, the computational time from SE-Apriori increased slower than Apriori. For example, when the *minsup* decreased from 12% to 10%, the computational time of Apriori increased by 20 sec, while that of the SE-Apriori increased only by 3 sec. The reason for this is when the *minsup* has a smaller value, a larger amount of candidate itemsets generate during the ARM process. Before checking the support of candidate itemsets against the whole dataset, SE-Apriori prunes them with the selective and exclusive constraints, which narrows the searching space of frequent itemsets, thus accelerating the execution.

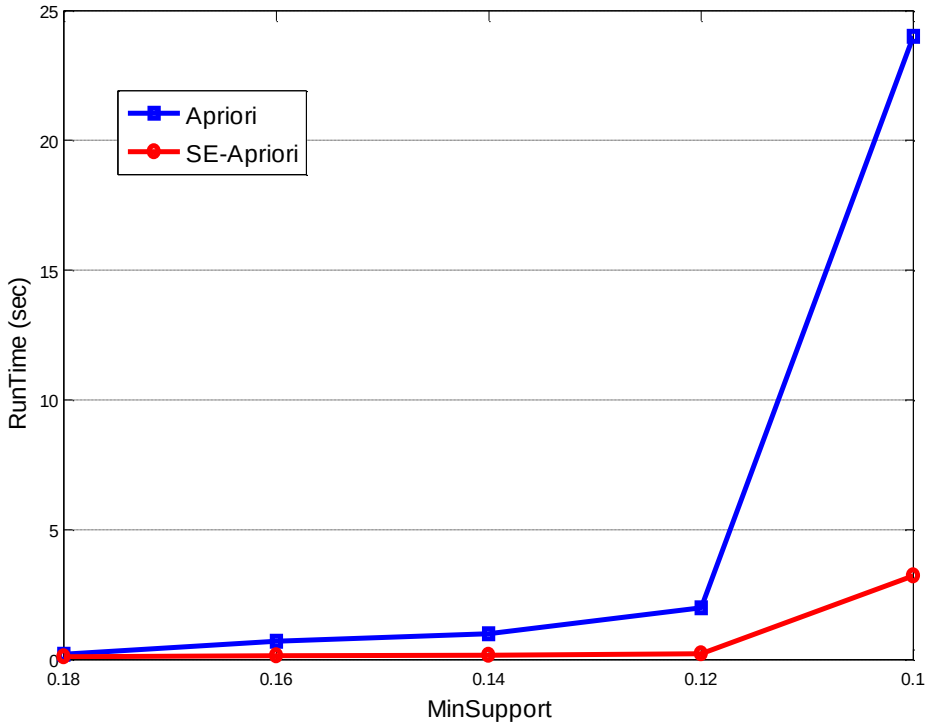

Figure 4–11 Comparison of computational time between SE-**Apriori and Apriori with varying *minsup* values**

4.5.2.2 Number of Generated Rules

The second experiment is to demonstrate that SE-Apriori generates a smaller number of association rules than Apriori. One shortcoming of association rule mining is that it generates a large number of rules, even though many of them do not indicate interesting associations. A concise result containing less association rules will save the effort to interpret them. Typically, the *minconf* threshold controls the number of rules by limiting the rules only to the ones with a high confidence value. With the *minconf* ranging from 50% to 90% and the *minsup* set to 12%, the number of generated rules from SE-Apriori and Apriori is compared.

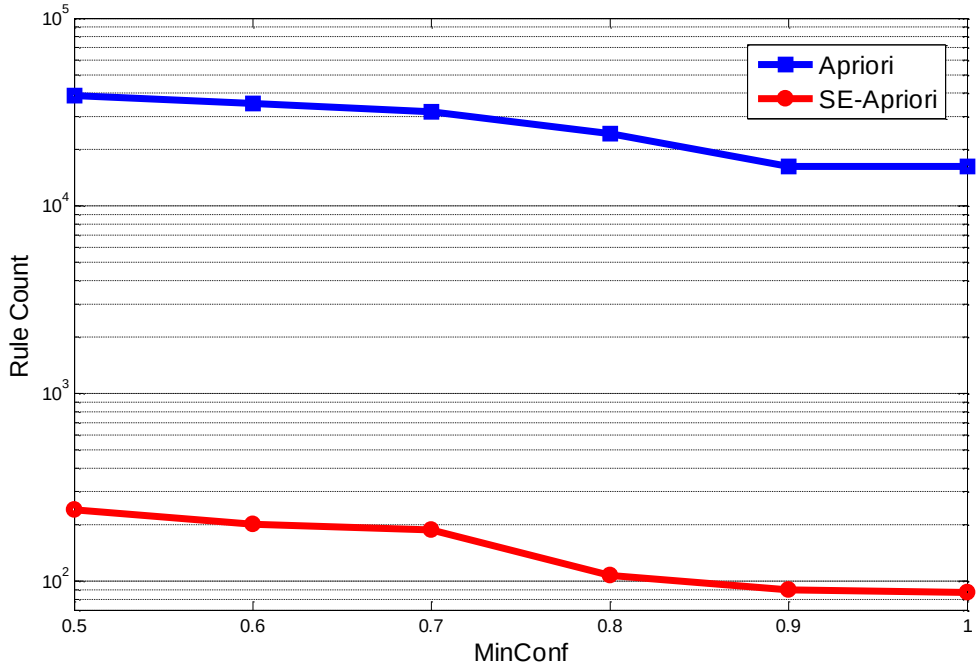

Figure 4–12 Comparison of the number of generated rules between SE-**Apriori and Apriori with varying *minconf* value (*minsup*=12%)**

Figure 4–12 shows the comparison of the number of association rules generated from SE-Apriori and Apriori when the minsup is set to 12%. From Figure 4–12, it is obvious that the number of generated rules from SE-Apriori is considerably less than Apriori. For example, with *minconf*=70% and *minsup*=12%, Apriori generates 31,861 rules while SE-Apriori only generates 187. There are two reasons why SE-Apriori generates a fewer number of rules. First, with selective and exclusive constrains, SE-Apriori prunes invalid frequent itemsets. Table 4–5 compares the number of frequent itemsets from Apriori and SE-Apriori with *minconf*=70%, *minsup*=12%. It is noted that Apriori finds 2,444 frequent itemsets and only 261 satisfying selective and exclusive constraints are kept by SE-Apriori. Second, each frequent itemset in Apriori may

generate many association rules while SE-Apriori generates at most one rule from a frequent itemset. In Apriori, for a frequent itemset X, each subset of X can generate the rule: subset (X)=>X-subset(X) as long as the confidence is larger than *minconf*. Thus, for a frequent 6-itemset, the number of possible rules is $C_6^1 + C_6^2 + C_6^3 + C_6^4 + C_6^5$. In contrast, SE-Apriori can generate at most one rule from a frequent itemset by satisfying Eq. (4.1). Table 4–5 shows that 2,444 frequent itemsets from Apriori generate 31,861 rules while 261 frequent itemsets from SE-Apriori only generate 187 rules.

Table 4–5 Numbers of frequent itemsets from Apriori and SE-Apriori with *minconf*=70%, *minsup*=12%

	Apriori (Count)	**SE-Apriori (Count)**
Frequent 1-Itemset	157	N/A
Frequent 2-Itemset	537	84
Frequent 3-Itemset	659	87
Frequent 4-Itemset	539	57
Frequent 5-Itemset	327	25
Frequent 6-Itemset	159	7
Frequent 7-Itemset	54	1
Frequent 8-Itemset	11	N/A
Frequent 9-Itemset	1	N/A
Total Frequent Itemsets	**2,444**	**261**
Total Generated Rules	**31,861**	**187**

4.5.3 Association Rule Results in HWPGA

The main objective of applying ARM in a HWPGA problem is to obtain interesting relationships between well placement attributes and SOR. Using the study dataset, two types of SE-Apriori analysis in HWPGA problems are presented. First, the sensitivity of

each well placement attribute influencing the SOR is analyzed. Second, sample high order association rules indicating the synergistic impact of multi-well placement attributes on SOR are presented.

The sensitivity analysis of each well placement attribute to SOR is based on the observation that the most sensitive well placement attribute generates the most widely applicable association rules. An association rule is widely applicable if there are a large number of records in the dataset satisfying this rule. A well placement attribute is sensitive to SOR only if it generates a group of rules which are applicable to a large portion of the records in the dataset. Specifically, if a well placement attribute is random to SOR, it either generates limited association rules or the value of the support to the generated rules tends to be zero. Thus, as in Eq. (4.5), a sensitivity index for a group of association rules is introduced, which is defined the summation of the support to each rule in the group.

$$SensitivityIndex = \sum_{Rule=1}^{t} N \times Support(Rule) \tag{4.5}$$

where N is the total number of records and t is the number of association rules.

Table 4–6 Values of *minsup* and *minconf* in the sensitivity analysis experiment

	Parameter Values	**Count**
minsup	8%, 10%, 12%	3
minconf	60%, 70%, 80%, 90%	4

To evaluate the sensitivity of each well placement attribute to SOR, 12 SE-Apriori tests with 3 *minsup* values and 4 *minconf* values, as listed in Table 4–6, are conducted. Association rule results from different SE-Apriori test are collected, and the

sensitivity index of each well placement attribute to the SOR is calculated, which is listed in Figure 4–13.

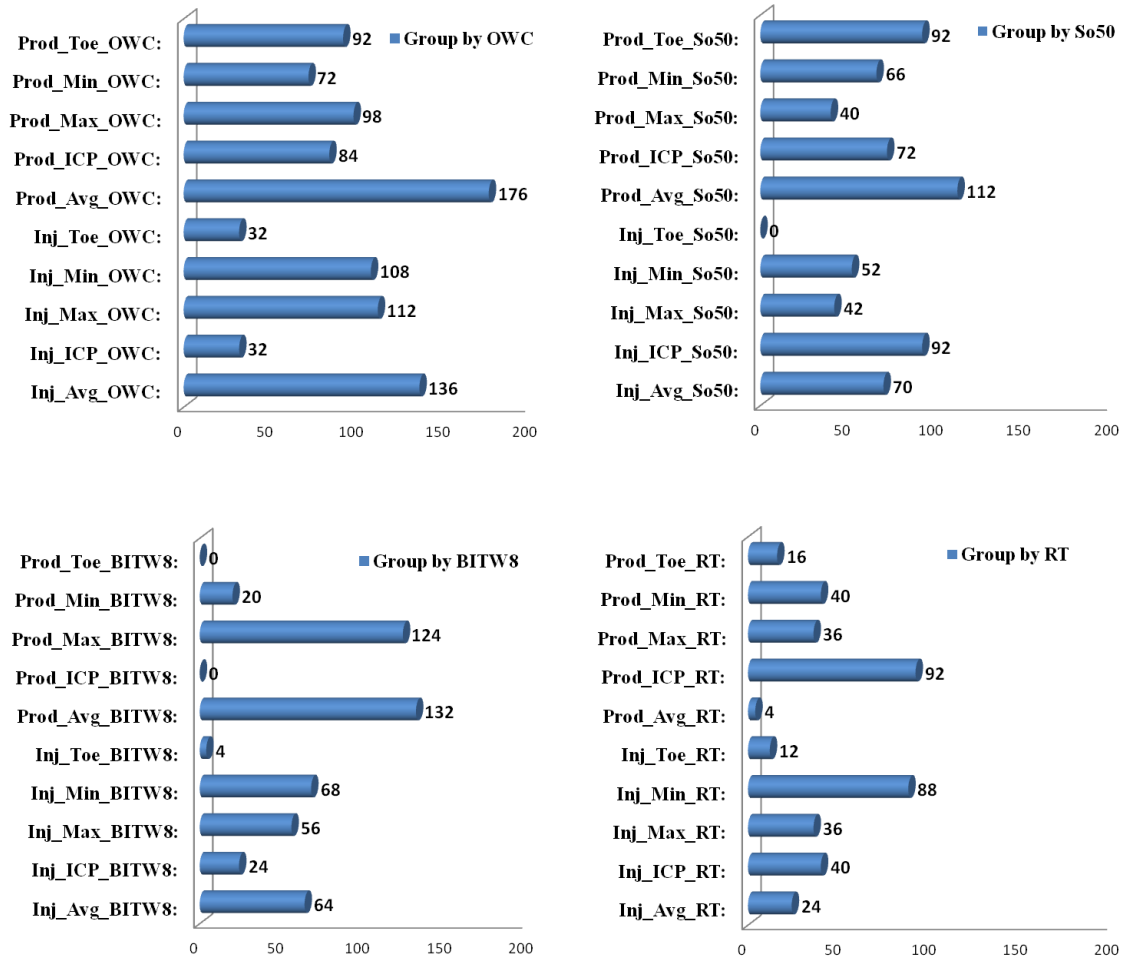

Figure 4–13 Sensitivity index to 40 well placement attributes grouped by geological surfaces

Figure 4–13 shows that the top 5 sensitive well placement attributes are Prod_Avg_OWC, Inj_Avg_OWC, Prod_Avg_BITW8, Prod_Max_BITW8 and Prod_Avg_So50. The five least sensitive well placement attributes are Prod_Toe_BITW8, Prod_ICP_BITW8, Inj_Toe_So50, Inj_Toe_BITW8 and Prod_Avg_RT. Meanwhile, the

well placement attributes in Figure 4–13 are grouped by geological surfaces. The summarized sensitivity indexes of well placement attributes belonging to the same geological surfaces are calculated. The summarized sensitivity indices to (OWC, So50, BITW8, RT) are 942, 638, 492 and 388, respectively. The geological surface OWC has the highest sensitivity index, which suggests that the distance between the horizontal wells and OWC acts as the most important rule influencing the SOR in the study dataset.

Table 4–7 Sample rules between multi-well placement attributes and SOR

Sample Rules	Sup	Conf
{Prod_Avg_OWC=2/8} AND {Prod_Max_OWC=2/8} AND {Prod_Min_OWC=2/8}AND {Prod_Toe_OWC=2/8 } => {SOR: Fair}	14%	71%
{Prod_Avg_OWC=1/8} AND {Prod_Max_OWC=1/8} AND {Prod_Min_OWC=1/8} AND {Prod_Toe_OWC=1/8 } => {SOR: Poor}	12%	83%

The second part of this analysis covers the synergistic impact of multi-well placement attributes on the SOR. One advantage of ARM is to present the associations among high dimensional attributes. In the study dataset, there are 40 well placement attributes and 1 SOR attribute. By analyzing the high order association rules, it may reveal the synergistic impact of different geological factors on the SOR. For example, with *minsup*=10% and *minconf*=70% SE-Apriori generates 33 rules with the order higher than 3 (equivalent to the antecedent of the rule which has more than 3 well placement attributes), and the highest order reaches 6. Two sample high order rules are shown in Table 4–7. These two rules have the same antecedent attributes but different values. In the first rule, Prod_Avg_OWC=2/8 means that Prod_Avg_OWC is partitioned into 8 intervals and it belongs to 2/8 level of the partition. The first rule reveals that there is a 71%

chance that the SAGD well pair has a fair SOR when the producer is placed 2/8 distance level to OWC. The second rule suggests that placing the wells closer to OWC, like from 2/8 to 1/8, there is a higher chance that the SOR tends to be ‘Poor’. Hence these two rules implicitly indicate that placing the SAGD well pair closer to OWC has a negative influence on oil production.

4.6 Summary

In this chapter, a customized method to analyze the horizontal well placement performance from real SAGD field data is given. It starts by formalizing the Horizontal well placement Guidance Acquisition (HWPGA) problem in Section 4.1. HWPGA is used to sort through related SAGD field data and identify interesting associations between horizontal well placement attributes and oil production performance. In addition, to capture the characteristics of horizontal well placement within a heterogeneous reservoir, a group of well placement attributes are defined in Section 4.2. In Section 4.3, association rule mining is introduced to solve the HWPGA problem. To improve efficiency, a new algorithm, named SE-Apriori, is modified from Apriori by considering two constraints in HWPGA problems. In Section 4.4, a GIS system containing the proposed SE-Apriori tool, named PetroData-GIS, is developed. Finally, in Section 4.5, the real dataset taken from a SAGD project in northern Alberta is used to demonstrate the SE-Apriori algorithm in solving the HWPGA problem. The experiments show that the SE-Apriori algorithm executes faster while generating a smaller number of association rules compared to Apriori. In the end, the SE-Apriori results from the study dataset are

evaluated by presenting the sensitivity analysis and high order association rule interpretation.

Chapter Five: **Conclusions and Future Work**

This chapter draws conclusions from this thesis and provides suggestions for future work.

5.1 Conclusions

In this thesis, the author applies data mining and GIS methods for two petroleum applications: reservoir characterization and horizontal well placement guidance acquisition.

The reservoir characterization problem focuses on two types of field data, i.e., core analysis and well log data. As a preprocessing step in reservoir characterization, a spatial clustering process is applied to group core analysis data with the spatial correlation. A new spatial clustering algorithm, named SEClu, is proposed to consider the nonspatial similarity and spatial correlation during the spatial clustering process. SEClu finds clusters whose members are *density-spEntropy-reachable* to each other. This requires data points in the same clusters not only to be in a dense neighbourhood but also to satisfy a small spatial entropy threshold. SEClu is able to identify clusters with arbitrary shapes. Meanwhile, it has been demonstrated in this thesis that spatial entropy is a deceasing function as nonspatial similarity and spatial correlation increase. A small threshold of spatial entropy restricts the nonspatial attributes of data points in the same cluster to be similar and spatially correlated. In the experiment, SEClu is evaluated with synthetic datasets and a real core analysis data clustering application. Experimental results show that compared to the traditional density-based spatial clustering algorithms SEClu performs better in finding meaningful clusters with spatial correlation.

In addition, in order to acquire accurate reservoir properties on a large scale, a new FR-Neural framework is proposed in reservoir characterization using core analysis and well log data. The proposed FR-Neural reservoir characterization framework includes two steps: fuzzy ranking and pattern recognition. The fuzzy ranking step selects the representative well log data for the target reservoir property characterization. In the pattern recognition step, a MLP neural network simulates the complex correlation from selected well log data to the target reservoir property. After proper training, the MLP predicts the target reservoir property based on new well log data. The FR-Neural framework is evaluated on a porosity characterization problem using data from three wells in southwestern Alberta, Canada. The reservoir characterization results from MLP using the fuzzy ranking selection is compared with results using other neural inputs. The comparison suggests that the proposed FR-Neural framework generates the best reservoir characterization results. Specifically, the correlation coefficient between the predicted porosity values from FR-Neural framework and the recorded values reaches up to 90%, which demonstrates the prediction accuracy of the proposed method.

The second problem discussed in this thesis surrounds horizontal well placement. Horizontal well placement is critical to the SAGD oil recovery process, and poor well placement negatively influences the oil production. This thesis formalizes a HWPGA problem which examines the horizontal well placement guidance by investigating real SAGD field data. It begins by defining a group of horizontal well placement attributes which characterize the locations of horizontal wells in a heterogeneous reservoir. Furthermore, a customized association rule mining algorithm, named SE-Apriori, is proposed, which solves the HWPGA problem by analyzing the interesting correlations

between horizontal well placement attributes and oil production performance. Two constraints, i.e., selective and exclusive constraints, are considered in the SE-Apriori algorithm, which narrows the frequent itemsets searching space and accelerate the process of association rule mining in a HWPGA problem. Given the minimum support and minimum confidence thresholds, SE-Apriori efficiently finds all satisfied association rules between the well placement attributes and oil production performance indicators. The proposed SE-Apriori algorithm is evaluated using a real dataset taken from a SAGD project in northern Alberta, Canada. Experimental results show that SE-Apriori can dramatically reduce the execution time while generating concise results regarding association rules. The generated association rules suggest that the oil production performance from the study dataset is very sensitive to the distance between horizontal wells and the OWC geological layer.

Furthermore, a GIS system prototype, named PetroData-GIS, is designed to efficiently manage the field data in the petroleum industry and visualize the data mining results. The PetroData-GIS prototype integrates the SE-Apriori tool into a GIS system. It helps manage a large volume of petroleum field data in a spatial database and visualizing the data with geographical information on a 2D map. The engineers can easily access the field data by clicking the symbols, such as wells, on the map. In the meantime, PetroData-GIS contains the SE-Apriori algorithm as a GIS function and helps to visualize the association rules from SE-Apriori. From PetroData-GIS, users can apply the SE-Apriori tool to analyze the association rules on selected attributes from the spatial database. Association rules from SE-Apriori can be visualized back in the PetroData-GIS prototype. Wells satisfying or not satisfying a specific rule are represented using different

symbols on the map, which helps the engineers in visually interpreting the association rule acquired from the system.

5.2 Future Work

Several extensions to this thesis are suggested and listed as follows:

1. The SEClu algorithm is developed to cluster spatial datasets by considering spatial attributes, nonspatial attributes and spatial correlation. Even though SEClu helps in identifying meaningful clusters with spatial correlation, it sacrifices computational efficiency. When the spatial dataset is large, the execution time of SEClu becomes unacceptable. Incorporating a spatial data index or a preprocessing method in the algorithm to reduce the execution time is suggested.
2. The current SEClu algorithm is only capable of clustering spatial points. In many cases, considering spatial correlation could also be very interesting. For example, clustering other spatial objects, such as polygons in land management systems. Hence extending SEClu to cluster other spatial objects would prove to be valuable research.
3. The proposed FR-Neural framework is suitable for characterizing most reservoir properties for different reservoir types. In this work, it is evaluated only for the porosity characterization problems for a gas reservoir. It would be interesting to examine this method for other reservoir properties, such as permeability, saturation and lithology, and for more complicated reservoirs, such as those with a bottom aquifer, top gas, dual porosity reservoir or natural fracture reservoirs.

4. The FR-Neural framework proposed in this thesis is designed to characterize reservoir properties from well log data. The derived reservoir properties are only available along with the well bore where the well logs are taken. For the reservoir regions further away from the well bores, reservoir properties are calculated via interpolation, which can be inaccurate. Several works have suggested that interpreting reservoir properties from seismic data is feasible. Therefore, it would be interesting to incorporate seismic data into the ANN-based reservoir characterization method in order to acquire accurate reservoir properties on a large scale.
5. For the HWPGA problem, horizontal well pairs in a SAGD project are treated independently in this work. In practice, due to pressure gradients in the reservoir, neighbour well pairs may communicate with each other after a period of production. Horizontal well placement may influence oil production by enhancing or weakening the communication effect. Therefore, incorporating spatial dependence into the well placement study may potentially help in delivering better horizontal well placement plans.
6. HWPGA is extendable to investigating the well placement performance for deviated or vertical wells. Despite the rapid development of horizontal wells in recent years, most traditional oil recovery technologies apply deviated or vertical wells and field data that have been accumulated over decades. Analyzing the reservoir response for different well placement plans may benefit traditional oil recovery by improving well planning and eventually increasing oil production.
7. Integrating data mining methods into GIS and providing solutions for the petroleum industry are new research topics. The PetroData-GIS prototype developed in this

thesis strives to integrate association rule mining tools into GIS and provide solutions to the HWPGA problems. In the meantime, there is high potential for combining other data mining methods with GIS to provide useful tools. This combination would assist engineers in exploring and identifying reservoirs with commercial value. It would also be helpful in the optimization of oil production, and in considering well dependency. Finally, it would allow for the visualization of data mining results combined with geographical information.

APPENDIX: PUBLICATION DURING THE PROGRAME

Journal Papers:

Wang B.J., Wang X. and Chen Z.X. Spatial Entropy-based Mutual Information in Hyperspectral Band Selection for Supervised Classification, *International Journal of Numerical Analysis and Modeling* (**Accepted**)

Wang B.J., Wang X. and Chen Z.X. Using Two-step Fuzzy Ranking and an Artificial Neural Network for Reservoir Characterization. *Computers & Geosciences* (**Submitted**)

Conference Papers:

Wang B.J. and Wang, X. (2011) "Spatial Entropy-based Clustering for Mining Data with Spatial Correlation." *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining,* ShenZhen, China, Springer LNCS, 6634, 196-208

Gu W., **Wang, B.J.** and Wang X. (2011) "An Integrated Approach to Multi-Criteria-based Health Care Facility Location Planning." *Proceedings of the 2nd Workshop on Data Mining for Healthcare Management*, ShenZhen, China, Springer LNCS, 420-430

Workshop Presentations:

Wang B.J. (2011) "Association Rule Mining of SAGD Database." *2011 Reservoir Simulation Technical Symposium Student Presentation*, Calgary

Wang B.J. (2010) "Using Two-step Fuzzy Ranking and Artificial Neural Network for Reservoir Characterization." *2010 Reservoir Simulation Technical Symposium Student Presentation*, Calgary

REFERENCE

Agrawal R., Imielinski T., and Swami A. (1993). "Mining association rules between sets of items in large databases." *The Proceeding of the ACM SIGMOD Conference on Management of Data*, Washington D.C., 207-216.

Agrawal R., and Srikant R. (1994). "Fast algorithms for mining association rules." *Proceedings of the 20th VLDB Conference*, Santiago, Chile., 487-499

Albahlani A.M., and Babadagli T. (2008). "A critical review of the status of SAGD: Where are we and what is next?" *SPE Western Regional and Pacific Section AAPG Joint Meeting*, California, USA., SPE 113283

Al-Bulushi N., King P.R., Blunt M.J., and Kraaijveld M. (2009). "Development of artificial neural network models for predicting water saturation and fluid distribution." *Journal of Petroleum Science and Engineering*, 68(3-4), 197-208.

Aminian K., and Ameri S. (2005). "Application of artificial neural networks for reservoir characterization with limited data." *Journal of Petroleum Science and Engineering*, 49(3-4), 212-222.

Aulia A., Ke, T.B., Maulut M.S., El-Khatib N., and Jasamai M. (2010). "Smart oilfield data mining for reservoir analysis." *International Journal of Engineering and Technology*, 10(6), 78-88.

Brock J. (1986). "Applied Open-hole Log Analysis." Gulf Pub. Co..

Chakrabarti S., Ester M., Fayyad U., Gehrke J., Han J., Morishita S., Piatetsky-Shapiro G. and Wang W. (2006) "Data mining curriculum: A proposal." ACM SIGKDD

Chang K. (2012). "Introduction to geographic information system." McGrawHill.

Chen M.Y.S., Fong J., and Leshchyshyn T. (1997). "Effects of well placement and critical operating conditions on the performance of dual well SAGD well pair in heavy oil reservoir." *Latin American and Caribbean Petroleum Engineering* Conference, Rio De Janeiro, SPE 39082

Chen T., Han D., Au F.T.K., and Tham L.G. (2003). "Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate." *Ieee Transactions on Neural Network*, 3(3), 1873-1878.

Claramunt C. (2005). "A spatial form of diversity." Spatial Information Theory, LNCS 3693, 218-231.

Coburn T.C., and Yarus J.M. (2000). "Geographic information systems in petroleum exploration and development.".

Creighton C., and Hanash S. (2003). "Mining gene expression databases for association rules." *Bioinformatics*, 19(1), 79-86.

Dominion Land Survey (DLS). "http://en.wikipedia.org/wiki/Dominion_Land_Survey" Last Accessed on Mar 10, 2012

El Ouahed A.K., Tiab D., and Mazouzi A. (2005). "Application of artificial intelligence to characterize naturally fractured zones in Hassi Messaoud Oil Field, Algeria." *Journal of Petroleum Science and Engineering*, 49(3-4), 122-141.

Ellis D.V., and Singer J.M. (2007). "Well logging for earth scientists." Springer.

Ester M., Kriegel H.P., Sander J., and Xu X.W. (1996). "A Density-based algorithm for discovering clusters in large spatial databases with noise." *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, USA, 226-231.

Frawley W., Piatetsky-Shapiro G., and Matheus C. (1992). "Knowledge discovery in database: An overview." *AI Magazine*, 13(3), 57-70.

Han J., and Kamber M. (2006). "Data mining concepts and techniques." Morgan Kaufmann.

Haykin S. (1999). "Neural networks: A comprehensive foundation." Prentice Hall.

Helle H.B., Bhatt A., and Ursin B. (2001). "Porosity and permeability predication from wireline logs using artificial neural networks: a North Sea case study." *Geophysical Prospecting*, 49, 431-444.

Jiao J., and Zhang Y. (2005). "Product portfolio identification based on association rule mining." *Computer-Aided Design*, 37, 149-172.

Kailing K., Kriegel H.P., Pryakhin A., and Schubert M. (2004). "Clustering multi-represented objects with noise." *Advances in Knowledge Discovery and Data Mining, Proceedings*, 3056, 394-403.

Leibovici D.G. (2009). "Defining spatial entropy from multivariate distribution of co-occurrences." LNCS, 392-404.

Li X., and Claramunt C. (2006). "A spatial entropy-based decision tree for classification of geographical information." *Transactions in GIS*, 10(3), 451-467.

Lim J.S. (2005). "Reservoir properties determination using fuzzy logic and neural networks from well data in offshore Korea." *Journal of Petroleum Science and Engineering*, 49(3-4), 182-192.

Lin Y.H., Cunningham G.A., and Coggeshall S.V. (1996). "Input variable identification - fuzzy curves and fuzzy surfaces." *Fuzzy Sets and Systems*, 82(1), 65-71.

Lin Y.H., Cunningham G.A., Coggeshall S.V., and Jones R.D. (1998). "Nonlinear system input structure identification: Two stage fuzzy curves and surfaces." *Ieee Transactions on Systems Man and Cybernetics Part A-Systems and Humans*, 28(5), 678-684.

Marquardt D.W. (1964). "An algorithm for least-squares estimation of non-linear parameters." *Journal of the Society for Industrial and Applied Mathematics*, 11, 431-441.

Marroquin I.D., Brault J.J., and Hart B.S. (2009). "A visual data mining methodology for seismic facies analysis: Part2-Application to 3D seismic data." *Geophysics*, 74(1), 13-23.

McLennan J.A., Ren W., Leuangthong O., and Deutsch C.V. (2006). "Optimization of SAGD well elevation." *Natural Resources Research*, 15(2), 119-127.

Mohaghegh S. (2000). "Virtual-intelligence applications in petroleum engineering: Part I - Artificial neural networks." *Journal of Petroleum Technology*, 52(9), 64-72.

Mohaghegh S. (2005). "Recent developments in application of artificial intelligence in petroleum engineering." *Journal of Petroleum Technology*, 57(4), 86-91.

Mohaghegh S., Arefi R., Ameri S., Aminiand K., and Nutter R. (1996). "Petroleum reservoir characterization with the aid of artificial neural networks." *Journal of Petroleum Science and Engineering*, 16(4), 263-274.

Mohaghegh S., Reeves S., and Hill D. (2000). "Development of an intelligent systems approach for reservoir candidate selection." *The Proceeding of 2000 SPE/CERI Gas Technology Symposium*, Calgary, Canada, SPE 59767.

Mohaghegh, S., Arefi, R., Ameri, S., Aminiand, K., and Nutter, R. (1996). "Petroleum reservoir characterization with the aid of artificial neural networks." *Journal of Petroleum Science and Engineering*, 16(4), 263-274.

Ng R.T., and Han J.W. (2002). "CLARANS: A method for clustering objects for spatial data mining." *Ieee Transactions on Knowledge and Data Engineering*, 14(5), 1003-1016.

Norrena K.P., and Deutsch C.V. (2002). "Automatic determination of well placement subject to geostatistical and economic constraints." *Proceedings of the 2002 SPE International Thermal Operations and Heavy Oil Symposium and International Horizontal Well Technology Conference*, Calgary, Canada, SPE 78996.

Sander J., Ester M., Kriegel H.P., and Xu X.W. (1998). "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications." *Data Mining and Knowledge Discovery*, 2(2), 169-194.

Shin H., and Polikar M. (2007). "Review of reservoir parameters to optimize SAGD and Fast-SAGD operating conditions." *Journal of Canadian Petroleum Technology*, 46(1), 35-41.

Sohn S.Y., and Kim Y. (2008). "Searching customer patterns of mobile service using clustering and quantitative association rule." *Expert System with Applications*, 34, 1070-1077.

Strecker U., and Uden R. (2010). "Data mining of 3D poststack seismic attribute volumes using Kohonen self-organizing maps." *The Leading Edge*, 21, 1032-1037.

Sturges H.A. (1926). "The choice of a class interval." *Journal of the American Statistical Association*, 21, 65-66.

Tobler W.R. (1970). "Computer movie simulating urban growth in Detroit region." *Economic Geography*, 46(2), 234-240.

Tutmez B., and Tercan A.E. (2007). "Assessment of uncertainty in geological sites based on data clustering and conditional probabilities" *Journal of Uncertain System*, 1(3), 207-221

Wang X., and Hamilton H.J. (2003). "DBRS: A density-based spatial clustering method with random sampling." *Advances in Knowledge Discovery and Data Mining*, 2637, 563-575.

Weka. "www.cs.waikato.ac.nz/ml/weka/" Last Accessed on Mar 10, 2012

Wong P.M., Gedeon T.D., and Taggart I.J. (1995). "An improved technique in porosity prediction - a neural-network approach." *Ieee Transactions on Geoscience and Remote Sensing*, 33(4), 971-980.

Wu X., Kumar V., Quinlan J.R., Ghosh J., Yang Q., Motoda H., Mclachlan G.J., Ng A., Liu B., Yu P.S., Zhou Z.H., Steinbach M., Hand D.J., and Steinberg D. (2007). "Top ten algorithms in data mining." *Knowledge and Information System*, 14(1), 1-37.

Zangl G., and Hannerer J. (2003). "Data mining applications in the petroleum industry." Round Oak Publishing.

Zhang S., Zhang C., and Yang Q. (2003). "Data preparation for data mining." *Applied Artificial Intelligence: An International Journal*, 17(5-6), 375-381.