Visual, Linguistic Data Mining Using Self-Organizing Maps

Dumidu Wijayasekara, *Student Member, IEEE*, Milos Manic, *Senior Member, IEEE*

Abstract— **Data mining methods are becoming vital as the amount and complexity of available data is rapidly growing. Visual data mining methods aim at including a human observer in the loop and leveraging human perception for knowledge extraction. However, for large datasets, the rough knowledge gained via visualization is often times not sufficient. Thus, in such cases data summarization can provide a further insight into the problem at hand. Linguistic descriptors such as linguistic summaries and linguistic rules can be used in data summarization to further increase the understandability of datasets. This paper presents a Visual Linguistic Summarization tool (VLS-SOM) that combines the visual data mining capability of the Self-Organizing Map (SOM) with the understandability of linguistic descriptors. This paper also presents new quality measures for ranking of predictive rules. The presented data mining tool enables users to 1) interactively derive summaries and rules about interesting behaviors of the data visualized though the SOM, 2) visualize linguistic descriptors and visually assess the importance of generated summaries and rules. The data mining tool was tested on two benchmark problems. The tool was helpful in identifying important features of the datasets. The visualization enabled the identification of the most important summaries. For classification, the visualization proved useful in identifying multiple rules that classify the dataset.**

Keywords; Linguistic summarization, predictive rule generation, data summarization, data visualization, Self-Organizing Maps

I. Introduction

DATA MINING and knowledge extraction from raw data is becoming more and more important and useful as the amount and complexity of data are rapidly increasing [1], [2]. Thus, efficient and easy to use data mining techniques are required to extract knowledge from these datasets. Data mining algorithms incorporate methodologies from different fields in order to effectively extract knowledge from large datasets, one of which is visual data mining.

Visual data mining is defined as the process of exploration, interaction and reasoning with the abstract data using natural human perception [3]. By including humans in the data mining process, the flexibility, creativity and general knowledge of the human is combined with the computation and storage capabilities of computers [1], [4]. Thus, the effectiveness and efficiency of the data mining process is increased [1], [5].

For further knowledge extraction it is required to understand underlying inter-dependencies and characteristics of data. Again size and complexity of datasets hinder the knowledge extraction process and traditional manual knowledge extraction is becoming obsolete [6]. Among these knowledge extraction techniques, data summarization is used as a data pre-exploration method to gather knowledge from previously unseen data. Data summarization is defined as the process of extracting the most important information from a dataset to produce a concise, more understandable version for a particular application [7]-[10]. Conventional data summarization is called statistical or numerical data summarization and uses precise terms such as mean, median and standard deviation. However, it has been shown that these terse, precise numbers are counter intuitive to natural human language and therefore, more difficult to understand [7], [8], [11]. Thus, humans tend to prefer and better understand linguistically expressed properties. Linguistic descriptors of data are also preferable when a higher level of understanding of data is required and when dealing with non-numeric data [7], [8]. Linguistic descriptors of data can be expressed in two forms: linguistic summaries and predictive rules.

Linguistic Summarization (LS) of data was introduced by Yager in 1982 [12] and since then has been applied used in many areas [10], [13]-[17]. More recently these Yager type summaries were extended to Zadeh type fuzzy "*if - then*" summaries [6]-[9], [18]. LS derives descriptive summaries that describe the most common characteristics of a dataset. Therefore, LS is used in data pre-exploration to discover these characteristics and inter-dependent dimensions.

Linguistic Predictive Rule (PR) generation is closely related to LS. PR generates rules that predict classes in the dataset [11], [19]-[21]. PR can also be expressed as Zadeh type fuzzy associative rules [18]. Using PR it is possible to understand the dimensions and their values that contribute to classification of data. In previous work clustering and classification capability of Self-Organizing Maps (SOM) have been used to generate fuzzy type associative PR [22], [23], as well as non-fuzzy type PR [24], [25]. SOM have been also used in conjunction with various other algorithms to increase the effectiveness of the generated PR [26]-[29]. Malone et al. and Hung used the cluster boundary of SOM to generate non fuzzy type classification rules [30], [31].

This paper presents 1) a linguistic data mining tool, VLS-SOM that automatically generates linguistic descriptors, 2) a

Dumidu Wijayasekara is with the Computer Science Dept, University of Idaho, Idaho Falls, ID 83402 USA (phone: 208-533-8149; e-mail: wija2589@vandals.uidaho.edu).

Milos Manic is with the Computer Science Dept, University of Idaho, Idaho Falls, ID 83402 USA (e-mail: misko@ieee.org).

novel SOM based linguistic descriptor visualization method and 3) tools that enable visual assessment of linguistic descriptors according to the application and 4) novel quality measures for predictive rule generation. These quality measures are based on quality measures proposed in [8], [9]. The implementation facilitates the generation of custom LS and PR by means of interactive control of the linguistic data mining process. Quality measures that were proposed by Wu et al. in [8], [9], and extended for the use in SOM in [32] are used for linguistic summarization of data. The presented tool was tested on two benchmark datasets and was shown to be helpful in identifying interesting patterns that exists in the data and in identifying dimensions that contribute to classification of data.

VLS-SOM utilizes the 3D visual data mining tool called the CAVE-SOM which was presented in [3]. CAVE-SOM utilizes the dimensionality reduction, generalization, and approximation capabilities of Self-Organizing Maps (SOM) and the immersive virtual environment known as the Cave Automated Virtual Environment (CAVE).

The rest of the paper is organized as follows: Section II provides background review of the SOM algorithm. Section III introduces the SOM based linguistic data mining and Section IV describes the implemented data mining system, VLS-SOM. Section V presents the experimental results, and Section VI concludes the paper.

II. Self-Organizing Maps

The Self-Organizing Map (SOM) algorithm was developed by Kohonen [33]. The SOM consists of a topological grid of neurons typically arranged in 1D or 2D lattice [34]. The fixed grid defines the spatial neighborhood of each neuron.

Each neuron maintains a synaptic weight vector $\vec{w}=\{w_1,\ldots,w_N\}$, where N is the dimensionality of the input space. A dataset D containing M data points can be expressed as:

$$D=\{d_1,d_2,\ldots..d_M\} \qquad (1)$$

where d_m represents a single data point and each data point d_m has N dimensions and is expressed as:

$$d_m=\{v_{m,1},v_{m,2},\ldots..,v_{m,N}\} \qquad (2)$$

where $v_{m,n}$ is the n^{th} dimension of the m^{th} data point.

The structure of a 2D SOM is depicted in Fig. 1(a). All neurons are first randomly initialized and then iteratively adapted based on the training set of input data. The training process can be described in several steps as follows [34]:

Step 1 - Initialization: Randomly initialize all synaptic weight vectors in the input domain.

Step 2 - Sampling: Select a random input pattern $\vec{d}_m$ from the training dataset.

Step 3 – Competitive Learning: Find the Best Matching Unit (BMU) for the current input pattern $\vec{d}_m$. The BMU is found by minimizing the Euclidean distance between the input pattern $\vec{d}_m$ and the synaptic weight vectors $\vec{w}$:

$$BMU(\vec{d}_m)=\arg\min_k \left\|\vec{d}_m-\vec{w}_k\right\|,\ k=1,2,\ldots,K \qquad (3)$$

Here, $BMU(\vec{d}_m)$ is the best matching unit for input pattern $\vec{d}_m$, operator$\|\ \|$ denotes the Euclidian distance norm, and K is the number of all the neurons in the SOM.

Step 4 – Cooperative Updating: Update the synaptic weight vectors of all neurons in SOM using the cooperative update rule:

$$\vec{w}_k(i+1)=\vec{w}_k(i)+\eta(i)\,h_{k,BMU(\vec{d}_m)}(i)\,(\vec{d}_m-\vec{w}_k(i)) \quad (4)$$

Here, i denotes the iteration, $\eta(i)$ is the learning rate and $h_{k,BMU(\vec{d}_m)}(i)$ is the value of the neighborhood function for the neuron k centered at $BMU(\vec{d}_m)$.

Step 5 – Convergence Test: Until a specified convergence criterion is met go to **Step 2**.

The learning process is controlled by the dynamic learning rate η and the neighborhood function h. The size of the neighborhood function and the learning rate is reduced exponentially to enforce a convergent behavior

The learning process described in **Steps 2-5** is repeated

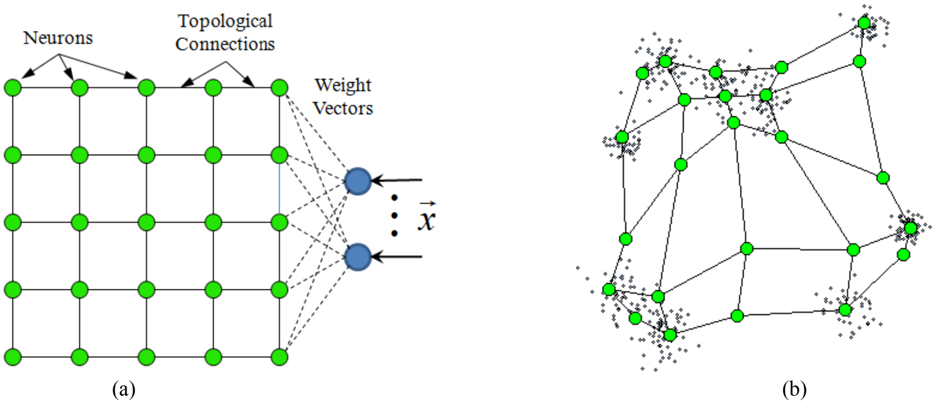


Fig. 1 Self-Organizing Map displayed in the output space (a) and in the input space adapted to 2D distribution of input points (b).

until a specific convergence criterion is met. An illustrative example of a 2D SOM in the input space adapted to a 2D distribution of data is shown in Fig. 1(b).

After convergence, the number of times each neuron k was selected as a best matching unit (BMU) was stored as $N_{BMU,k}$, such that:

$$\sum_{k=1}^{K} N_{BMU,k} = M \qquad (5)$$

where K is the number of neurons and M is the total number of data points.

Further, for labeled data, the number of times each neuron k was selected as a best matching unit (BMU) for each class was stored as $N_{BMU,k,c}$, such that:

$$\sum_{k=1}^{K}\sum_{c=1}^{C} N_{BMU,k,c} = M \qquad (6)$$

where c is the class label and C is the number of classes in the dataset.

III. SOM Based Linguistic Data Mining

This section discusses linguistic summarization process as applied to SOM.

Different types of linguistic uncertainties can be used to represent the uncertainty of the dataset. Yager type linguistic uncertainty uses terms such as "*more, less than half, all of*" etc. to describe datasets [12]. Zadeh type fuzzy uncertainty divides each dimension into membership functions and uses these membership functions to derive "*if, then*" type associative descriptors [18]. This paper focuses on Zadeh type "*if, then*" linguistic summaries and predictive rules for data mining.

An "*if, then*" type fuzzy linguistic summary can be expressed as:

$$IF \;\; ant_1 \;\; IS \;\; S_1 \;\; AND \;\; ant_2 \;\; IS \;\; S_2 \;\; THEN \;\; cons \;\; IS \;\; S_3 \qquad (7)$$

where ant_1 and ant_2 are the antecedents and S_1 and S_2 are the fuzzy sets of the two antecedents, respectively, and *cons* is the consequent and S_3 is the fuzzy set of the consequent. In linguistic summarization each antecedent and consequent of fuzzy rules are different dimensions of the dataset, thus the fuzzy linguistic summary in (7) can be re-written in terms of dimensions as:

$$IF \;\; v_a \;\; IS \;\; S_1 \;\; AND \;\; v_b \;\; IS \;\; S_2 \;\; THEN \;\; v_c \;\; IS \;\; S_3 \qquad (8)$$

where $a \neq b \neq c$ and v_a , v_b and v_c are dimensions of the dataset.

However, for predictive rules, the consequent is the class predicted by the rule. Therefore the linguistic summary in (8) can be converted into a predictive rule as:

$$IF \;\; v_a \;\; IS \;\; S_1 \;\; AND \;\; v_b \;\; IS \;\; S_2 \;\; THEN \;\; class \;\; IS \;\; c \qquad (9)$$

where c is the predicted class label.

For the sake of simplicity and ease of understanding, in this paper each dimension of the input dataset was decomposed into three fuzzy sets shown in Fig. 2. However, depending on the requirements of the application, the number and shape of the fuzzy sets can be changed for each dimension. An input value x is mapped to a degree of belonging to each fuzzy set called the membership degree, which is denoted as $\mu_S(x)$. Using this membership degree it is possible to derive a goodness measure for a linguistic summary or a predictive rule.

In order to generate linguistic descriptors that are more pertinent to the dataset all possible descriptors are generated, and various quality measures are used to rank these generated descriptors. In this paper three quality measures presented in [8], [9] and modified in [32] for the use in SOM, are used for ranking linguistic summaries. For linguistic rule generation, new quality measures are proposed, that are based on the quality measures used for linguistic summarization.

A. Linguistic Summarization

Linguistic summarization derives summaries of the dataset which are descriptive, and provide an understanding of the distribution of each dimension. Thus, the consequent of a linguistic summary is a dimension of the dataset.

As mentioned earlier, in this paper, three quality measures are used for ranking linguistic summaries.

The degree of truth was first presented by Yager in [12], and was also used by Wu et al. in [8] and [9]. Degree of truth is calculated by deriving the minimum membership degree for each antecedent and consequent, for all the data points. The degree of truth, T_{SOM} modified to be used in SOM can be expressed as [32]:

$$T_{SOM} = \frac{\sum_{k=1}^{K} \min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{S3}(w_{k,c})) \times N_{BMU,k}}{\sum_{k=1}^{K} \min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b})) \times N_{BMU,k}} \qquad (10)$$

According to the equation, if none of the data points satisfy the antecedents, then the denominator goes to zero thus invalidating the summary. As the number of data points that does not satisfy the consequent increase the numerator goes to zero thereby reducing the degree of truth for the

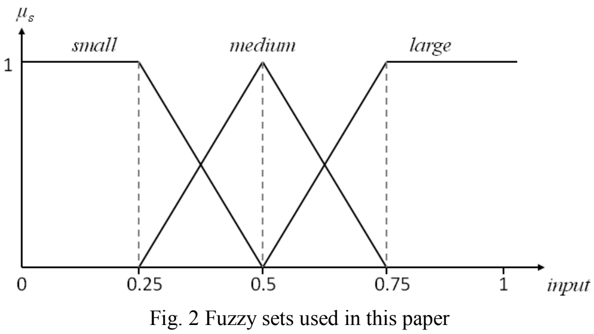


Fig. 2 Fuzzy sets used in this paper

summary.

Therefore, the larger the degree of truth the more data points satisfy both the antecedents and the consequents. However the degree of truth does not elaborate on the number of data points each summary is associated with.

Degree of sufficient coverage was presented by Wu et al. [8], [9], and was also modified in [32] for the use in SOM. This quality measure uses the percentage of data points that supports a certain rule. Thus degree of sufficient coverage C_{SOM} is expressed as:

$$C_{SOM} = f_c\left(\frac{\sum_{k=1}^{K} t_k}{M}\right) \qquad (11)$$

where,

$$t_k = \begin{cases} 1, (\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{S3}(w_{k,c})) \times N_{BMU,k} > 0 \\ 0, \ otherwise \end{cases} \qquad (12)$$

The function f_c is used to specify the necessary coverage for the dataset. In this paper the sigmoid function shown in Fig. 3 was used. The values r_{min} and r_{max} can be set by the user according to prior knowledge of the data distribution.

Degree of reliability proposed in [9] combines the degree of truth and degree of sufficient coverage. A summary is more expressive of the dataset if it has a high truth and a high coverage. Thus degree of reliability R_{SOM} is expressed as:

$$R_{SOM} = \min(T_{SOM}, C_{SOM}) \qquad (13)$$

where T_{SOM} is the degree of truth and C_{SOM} is the degree of sufficient coverage.

B. Predictive Rule Generation

Linguistic rules provide a predictive classification of the dataset using linguistic terms. Thus the consequent of a predictive rule is the class it predicts. These rules are used to understand clusters within the dataset and the dimensions that contribute to the separation of the cluster.

Many quality measures have been proposed in the past for generating predictive rules [19]-[22], [31], [35]. However, in this paper new quality measures are proposed that can be used to generate linguistic rules from SOM. The proposed quality measures are based on the quality measures used for generating summaries, and therefore has the same advantages.

The membership degree of the consequent is calculated differently for rule generation. In this paper the membership degree of the consequent class c for neuron k, $\mu_{k,c}$, is calculated as:

$$\mu_{k,c} = \frac{N_{BMU,k,c}}{\max_j (N_{BMU,j,c}), \ j = 1,2...,K} \times \frac{N_{BMU,k,c}}{\sum_{y=1}^{C} N_{BMU,k,y}} \qquad (14)$$

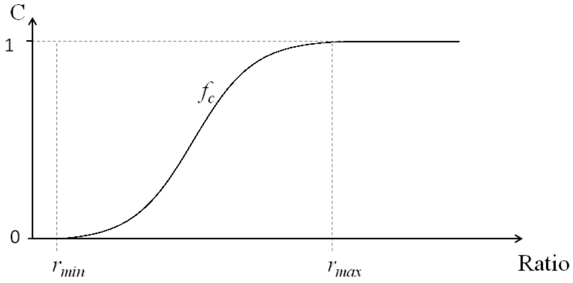


Fig. 3. Sigmoid function used to calculate the Degree of Coverage

The first term of the equation normalizes the number of times neuron k was selected as the best matching unit for class c. Therefore, as the number of data points associated with a neuron is reduced the membership degree of the neurons is also reduced. The second term calculates the ratio between the number of times neuron k was selected as the best matching unit for class c and the total number of times the neuron was selected as the best matching unit. Thus, the membership degree is further reduced if the neuron is associated with multiple classes.

Using $\mu_{k,c}$, the quality measures used for linguistic summarization were modified for ranking of predictive rules. The degree of truth for the predictive rule in (9) can be expressed as:

$$T_{PRED} = \frac{\sum_{k=1}^{K} \min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{k,c}) \times N_{BMU,k}}{\sum_{k=1}^{K} \min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b})) \times N_{BMU,k}} \qquad (15)$$

Similarly degree of sufficient coverage for predictive rules is expressed as:

$$C_{PRED} = f_c\left(\frac{\sum_{k=1}^{K} t_k}{M}\right) \qquad (16)$$

where,

$$t_k = \begin{cases} 1, (\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{k,c}) \times N_{BMU,k} > 0 \\ 0, \ otherwise \end{cases} \qquad (17)$$

Using degree of truth and degree of sufficient coverage, degree of reliability for predictive rules is expressed as:

$$R_{PRED} = \min(T_{PRED} C_{PRED}) \qquad (18)$$

As the degree of reliability is the most expressive quality measure [9], it was used as the final ranking quality measure for linguistic summaries and predictive rules.

IV. VLS-SOM Implementation

VLS-SOM takes advantage of the data compression, generalization and visualization capabilities of the SOM and the higher level understandability of linguistic descriptors. The SOM is used to visualize large multi dimensional datasets in lower, human perceivable dimensions. Using SOM users are able to identify clusters and patterns in the input dataset.

VLS-SOM utilizes CAVE-SOM presented in [3] to visually represent data. The CAVE-SOM is a 3-dimensional SOM implemented in the immersive virtual environment known as CAVE. The CAVE-SOM utilizes the color, size and transparency of neurons to provide vital information about the dataset to the user. VLS-SOM builds upon the visualization provided by CAVE-SOM, by enabling the user to visualize linguistic descriptors by means of color and size of neurons.

VLS-SOM further implements functionalities that enable the user to control the summarization process and generate summaries and rules depending on the application. VLS-SOM implements two main toolsets for data mining: linguistic summarization tools and rule generation tools. The tool also utilizes different visualization techniques to convey information about generated rules and summaries to the user visually. Fig 4. illustrates the visual linguistic data mining process of VLS-SOM.

The data mining process is usually expressed as a three step process: *1) overview first, 2) zoom and filter and 3) details-on-demand* [1]. VLS-SOM follows these three steps by first visualizing the whole data set, then generating linguistic predictors that summarize the dataset, and finally allowing the user to generate linguistic predictors on specific dimensions and clusters.

A. Linguistic Summary and Predictive Rule Generation

VLS-SOM allows the user to generate summaries or rules according to the requirements of the application. Using the GUI users are able to set the desired antecedents and consequent of the generated summary or rule (see Fig. 5(e)). This allows the generation and visualization of summaries and rules that span only a selected dimension. Furthermore for predictive rule generation users are able to generate rules for specific classes in the dataset.

The interactive GUI also enables the user to select the quality measure linguistic descriptors are ranked upon, thereby generating descriptors suited for different uses.

B. Visualization

After generating linguistic summaries or predictive rules, users are able to visualize them by means of color or size of the neurons (see Fig. 5(b) and 5(c)). These visualizations enable the user to understand the distribution and the truth of the summaries and visually asses the importance of these trends.

In order to visualize linguistic summaries and predictive rules, the presented quality measures in Section III must be calculated for each neuron separately.

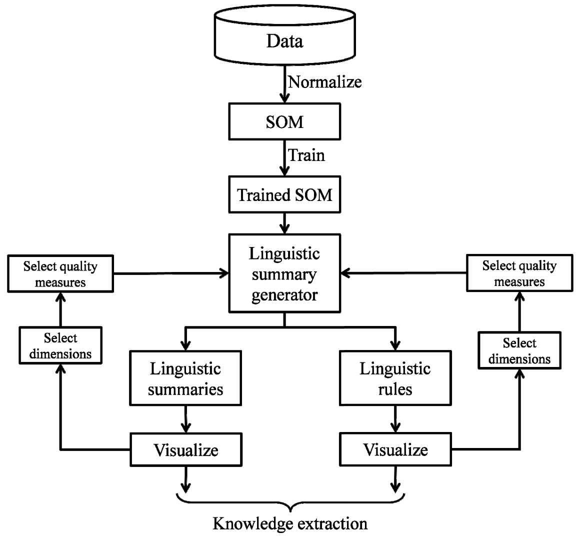


Fig. 4. Visual linguistic data mining process used by VLS-SOM

Thus degree of truth of a linguistic summary for a single neuron *k* can be expressed as:

$$T_{k,SOM} = \frac{\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{S3}(w_{k,c})) \times N_{BMU,k}}{\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b})) \times N_{BMU,k}} \quad (19)$$

Similarly the degree of truth of a predictive rule for neuron *k* can be expressed as:

$$T_{k,PRED} = \frac{\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b}), \mu_{k,c}) \times N_{BMU,k}}{\min(\mu_{S1}(w_{k,a}), \mu_{S2}(w_{k,b})) \times N_{BMU,k}} \quad (20)$$

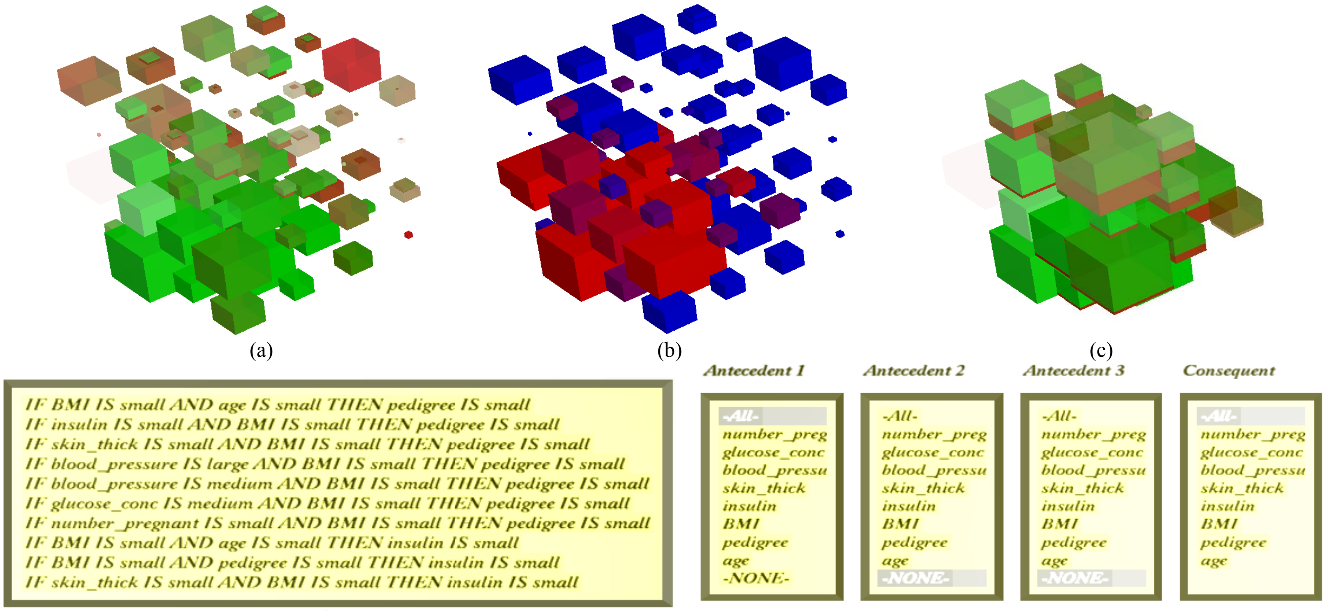
Degree of sufficient coverage is calculated for each neuron by means of the fraction each neuron contributes to the overall degree of sufficient coverage. Thus, degree of sufficient coverage for neuron *k* is expressed as:

$$C_k = f_c\left(\frac{t_k}{\sum_{y=1}^{K} t_y}\right) \quad (21)$$

where *t* is calculated using equations (12) and (17) for linguistic summarization and predictive rule generation, respectively.

These per neuron quality measures are then used to calculate the degree of reliability for each neuron using equations (13) and (18). The calculated quality measures for each neuron can then be mapped to the size or the color of each neuron. The implemented GUI allows users to select different rules and visualize them using either color or size of the neurons.

For predictive rules VLS-SOM provides a visualization that shows false positives and false negatives by using color of the neurons (see Fig. 6(d)). This enables the user to visually asses the performance of generated rules.



(a)

(b)

(c)

IF BMI IS small AND age IS small THEN pedigree IS small
IF insulin IS small AND BMI IS small THEN pedigree IS small
IF skin_thick IS small AND BMI IS small THEN pedigree IS small
IF blood_pressure IS large AND BMI IS small THEN pedigree IS small
IF blood_pressure IS medium AND BMI IS small THEN pedigree IS small
IF glucose_conc IS medium AND BMI IS small THEN pedigree IS small
IF number_pregnant IS small AND BMI IS small THEN pedigree IS small
IF BMI IS small AND age IS small THEN insulin IS small
IF BMI IS small AND pedigree IS small THEN insulin IS small
IF skin_thick IS small AND BMI IS small THEN insulin IS small

(d)

Antecedent 1	Antecedent 2	Antecedent 3	Consequent
-All-	-All-	-All-	-All-
number_preg	number_preg	number_preg	number_preg
glucose_conc	glucose_conc	glucose_conc	glucose_conc
blood_pressu	blood_pressu	blood_pressu	blood_pressu
skin_thick	skin_thick	skin_thick	skin_thick
insulin	insulin	insulin	insulin
BMI	BMI	BMI	BMI
pedigree	pedigree	pedigree	pedigree
age	age	age	age
-NONE-	-NONE-	-NONE-	

(e)

Fig. 5. Linguistic data mining tool applied to the Pima Indian Diabetes dataset. (a) The trained SOM, (b) summary visualization using color, (c) the same summary visualized in size, (d) generated summaries, (e) antecedent, consequent selection tool.

V. Experimental Results

In this section the proposed visual, linguistic data mining tool, VLS-SOM is tested against 2 benchmark datasets. The emphasis of this analysis was not on the comparison of different quality measures but on the usefulness of the descriptors generated.

The visualization of the presented tool is based on CAVE-SOM which was first presented in [3]. CAVE-SOM represents neurons as cubes in 3D space. The size of the neuron represents the number of data points associated with each neuron and the transparency represents the distance between neurons. Furthermore, the color of the neurons represent the class of the neuron. Neurons that are associated with multiple classes are visualized as histograms that show the number of points associated with each class by means of size.

A. Benchmark Problem I – Pima Indians dataset

The first benchmark problem used was the Pima Indians Diabetes dataset [36]. The Pima Indians Diabetes dataset contains data about 768 female patients, with 268 patients tested positive for diabetes. The dataset contains 8 dimensions and arranged into two classes: patients with diabetes and patients without diabetes.

Using linguistic summarization it is possible to generate summaries that explain the data in a human understandable way. Fig. 5(a) shows the trained SOM, where red neurons represent the patients tested positive for diabetes. Fig. 5(d) shows the generated summaries for the dataset. Using these summaries it was possible to identify important characteristics of the dataset. After identifying interesting patterns it was possible to drill-up and drill-down using the antecedent consequent selection toolbox (Fig. 5(e)). This toolbox allowed the selection of the number of antecedents of the summaries as well as the actual dimension of the antecedent of the consequent.

The visualization tool enabled the visualization of each summary by means of color or size of neurons. Fig 5(b) shows a summary visualized using the color of the neurons. Red shows neurons that follow the summary the best, while blue neurons show neurons that follow the summary the least. Using this visualization it was possible to obtain an understanding about the amount of data the summary covers. Similarly Fig. 5(c) shows the same summary visualized as the size of neurons. Again this visualization shows the distribution of the summary within the dataset while at the same time showing the class distribution of the data the summary covers.

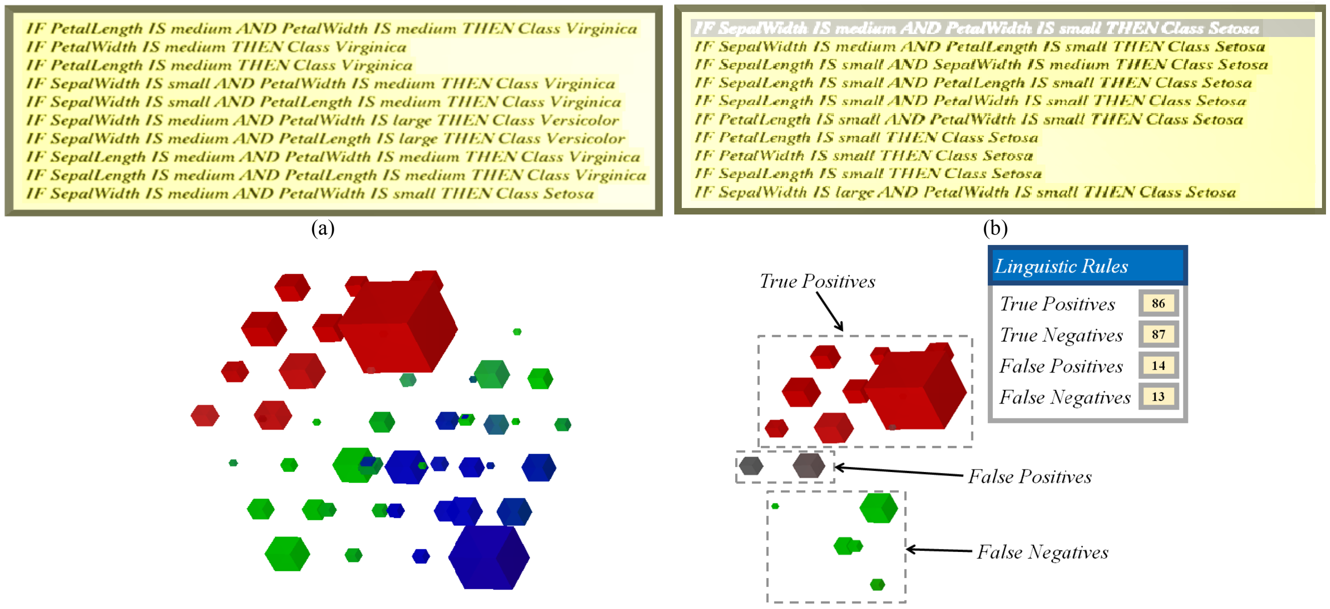
B. Benchmark Problem II – Iris dataset

VLS-SOM was used to analyze the well-known iris dataset [36]. The iris dataset is a typical benchmark problem describing the separation among three species of Iris flowers – Setosa, Virginica, and Versicolor. Each data point is described using 4 features: the length and the width of the sepal and petal. The dataset consists of 150 patterns, divided in 50 patterns for each class.

By utilizing the predictive rule generation tool, rules that can classify the dataset were generated. Fig. 6(a) shows two-antecedent rules that were generated for the iris dataset, and it can be seen that most of the top rules are for classifying Virginica class. Similar to LS, by using the antecedent consequent selection tool box, users can select the number of

TABLE I. True / False Positive and True / False Negatives for Classification

	Classified as class *A*	**Classified as not class *A***
Actually belongs to class *A*	True positives	False positives
Actually does not belong to class *A*	False negatives	True negatives



IF PetalLength IS medium AND PetalWidth IS medium THEN Class Virginica
IF PetalWidth IS medium THEN Class Virginica
IF PetalLength IS medium THEN Class Virginica
IF SepalWidth IS small AND PetalWidth IS medium THEN Class Virginica
IF SepalWidth IS small AND PetalLength IS medium THEN Class Virginica
IF SepalWidth IS medium AND PetalWidth IS large THEN Class Versicolor
IF SepalWidth IS medium AND PetalLength IS large THEN Class Versicolor
IF SepalLength IS medium AND PetalWidth IS medium THEN Class Virginica
IF SepalLength IS medium AND PetalLength IS medium THEN Class Virginica
IF SepalWidth IS medium AND PetalWidth IS small THEN Class Setosa

(a)

IF SepalWidth IS medium AND PetalWidth IS small THEN Class Setosa
IF SepalWidth IS medium AND PetalLength IS small THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS medium THEN Class Setosa
IF SepalLength IS small AND PetalLength IS small THEN Class Setosa
IF SepalLength IS small AND PetalWidth IS small THEN Class Setosa
IF PetalLength IS small AND PetalWidth IS small THEN Class Setosa
IF PetalLength IS small THEN Class Setosa
IF PetalWidth IS small THEN Class Setosa
IF SepalLength IS small THEN Class Setosa
IF SepalWidth IS large AND PetalWidth IS small THEN Class Setosa

(b)

(c)

Linguistic Rules

True Positives	86
True Negatives	87
False Positives	14
False Negatives	13

(d)

Fig. 6. Linguistic data mining tool applied to the Iris Diabetes dataset. (a) Initial set of rules, (b) rules generated for the Setosa class, (c) the trained SOM, (d) the false positive/false negatives visualization of the selected rule in (b).

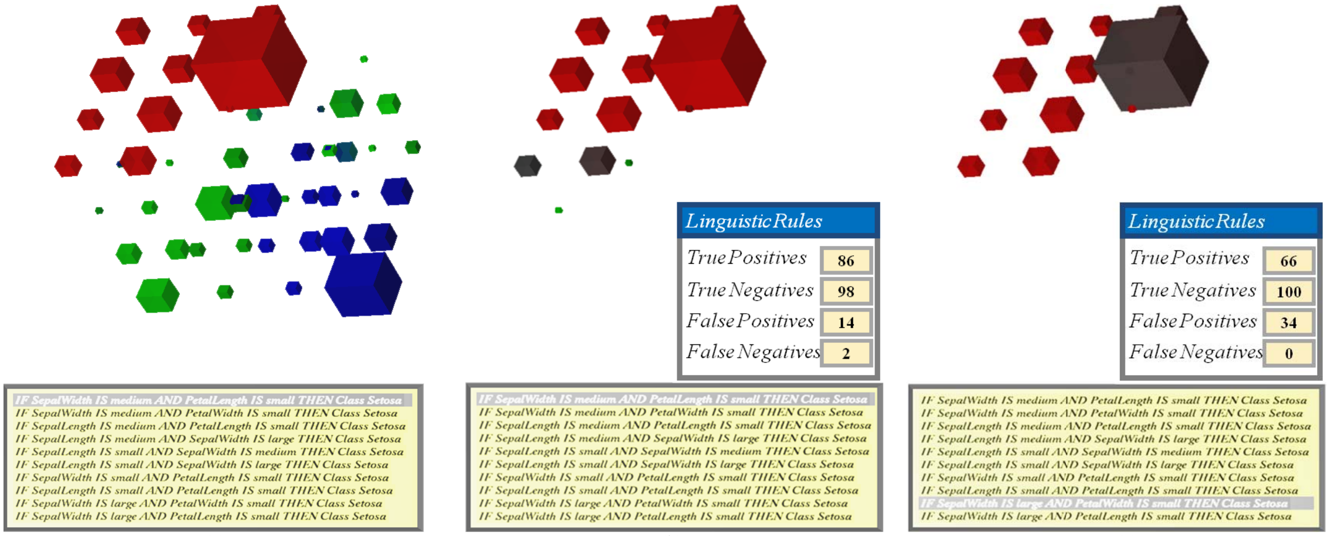
antecedents, antecedent dimensions and the consequent class for the rules. Fig. 6(b) show rules that were generated specifically for Setosa class.

The generated rules can then be visualized using the size or the color of the neurons. Similar to LS visualization, the size or the color of the neuron can emphasize the support of each neuron for the generated rule. For predictive rules a third visualization was implemented that shows the true and false positives and true and false negatives (see Table I). Fig. 6(d) shows the false-positive/false-negative view for the selected rule in Fig. 6(b). The size of the neurons show the number of data points associated with each neuron. True positives are shown in the color of the class that the selected rule is classifying (in Fig. 6(d) red). False positives are shown in grey. True negatives are not displayed and false negatives are shown in the colors each class is associated with (in Fig. 6(d) green). The visualization also shows the true/false positive and true/false negative percentages and allowed the selection optimal rules for the classification of the dataset.

Fig. 7 illustrates the use of predictive rule generation and visualization. Rules for classification of the Setosa class were generated (Fig. 7 (a)), however the best rule did not completely classify the Setosa class, i.e. there were some false positives (Fig. 7 (b)). However, by means of the visualization it was possible to identify the portion of the dataset that the rule fails to classify and find another rule that covers that portion. Fig. 7 (c) shows a rule that covers the portion of the dataset that the rule in Fig. 7 (a) failed to cover. Thus, by combining these two rules it was possible to generate a rule that classified the whole dataset.

VI. Conclusion

This paper presented a visual, linguistic data mining tool



IF SepalWidth IS medium AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS medium AND PetalWidth IS small THEN Class Setosa
IF SepalLength IS medium AND PetalLength IS small THEN Class Setosa
IF SepalLength IS medium AND SepalWidth IS large THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS medium THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS large THEN Class Setosa
IF SepalWidth IS small AND PetalLength IS small THEN Class Setosa
IF SepalLength IS small AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS large AND PetalWidth IS small THEN Class Setosa
IF SepalWidth IS large AND PetalLength IS small THEN Class Setosa

(a)

Linguistic Rules

True Positives	86
True Negatives	98
False Positives	14
False Negatives	2

IF SepalWidth IS medium AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS medium AND PetalWidth IS small THEN Class Setosa
IF SepalLength IS medium AND PetalLength IS small THEN Class Setosa
IF SepalLength IS medium AND SepalWidth IS large THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS medium THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS large THEN Class Setosa
IF SepalWidth IS small AND PetalLength IS small THEN Class Setosa
IF SepalLength IS small AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS large AND PetalWidth IS small THEN Class Setosa
IF SepalWidth IS large AND PetalLength IS small THEN Class Setosa

(b)

Linguistic Rules

True Positives	66
True Negatives	100
False Positives	34
False Negatives	0

IF SepalWidth IS medium AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS medium AND PetalWidth IS small THEN Class Setosa
IF SepalLength IS medium AND PetalLength IS small THEN Class Setosa
IF SepalLength IS medium AND SepalWidth IS large THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS medium THEN Class Setosa
IF SepalLength IS small AND SepalWidth IS large THEN Class Setosa
IF SepalWidth IS small AND PetalLength IS small THEN Class Setosa
IF SepalLength IS small AND PetalLength IS small THEN Class Setosa
IF SepalWidth IS large AND PetalWidth IS small THEN Class Setosa
IF SepalWidth IS large AND PetalLength IS small THEN Class Setosa

(c)

Fig. 7.Classification of the Setosa using predictive rule generation in VLS-SOM.

that utilizes the data visualization capability of the Self-Organizing Map and understandability of linguistic descriptors. The presented method implements interactive tools that allow the generation of custom descriptors based on the requirement of the application, as well as visualize the generated descriptors. The paper also presents novel quality measure for the ranking of predictive rules.

The implemented tool follows the three step data exploration process by, first: providing an overview by means of visualization of the data, second: zoom and filter by means of generating custom descriptors, and third: provide details on demand by generating descriptors for select portions or dimensions of the data.

The presented tool was tested on two benchmark datasets. The tool proved to be valuable in extracting knowledge from the datasets. The implemented tools enabled the generation of custom descriptors. The visualization methodologies enabled the identification of most important summaries as well as identifying multiple classification rules.

REFERENCES

[1] D. A. Keim, "Information Visualization and Visual Data Mining," in *IEEE Trans. on Visualization and Computer Graphics*, vol. 7, no. 1, pp. 100-107, January-March 2002.

[2] S. Sato, Y. Tateyama, "Super High Definition Three-Dimensional Display Environment Applied to Visual Data Mining," in *Proc. of the 13th International Conference on Network-Based Information Systems*, pp. 414-419, 2010.

[3] D. Wijayasekara, O. Linda, M. Manic, "CAVE-SOM: Immersive visual data mining using 3D Self-Organizing Maps," in *Proc. of The 2011 International Joint Conference on Neural Networks (IJCNN)*, pp.2471-2478, July 31-Aug. 5 2011.

[4] E. Mozzafari, A. Seffah, "From Visualization to Visual Mining: Application to Environmental Data," in *Proc. of the First International Conference on Advances in Computer-Human Interaction*, pp. 143-148, 2008.

[5] J. J. Valdes, “Evolutionary Computation Based Nonlinear Transformations to Low Dimensional Spaces for Sensor Data Fusion and Visual Data Mining,” in *Proc. of 2010 IEEE World Congress on Computational Intelligence – WCCI*, Barcelona, Spain, pp. 2242-2249, July, 2010.

[6] Ding-An Chiang , Louis R. Chow, Yi-Fan Wang, "Mining time series data by a fuzzy linguistic summary system," in *Fuzzy Sets and Systems*, vol. 112, pp. 419-432, 2000.

[7] A. Niewiadomski, "A Type-2 Fuzzy Approach to Linguistic Summarization of Data," in *IEEE Transactions On Fuzzy Systems*, vol. 16, no. 1, pp. 198-212, Feb. 2008.

[8] D. Wu, J. M. Mendel, J. Joo, "Linguistic Summarization Using IF-THEN Rules," in *Proc. IEEE International Conference on Fuzzy Systems*, pp. 1 - 8 , July 2010.

[9] D. Wu, J .M. Mendel, "Linguistic Summarization Using IF–THEN Rules and Interval Type-2 Fuzzy Sets," in *IEEE Transactions On Fuzzy Systems*, vol. 19, no. 1, pp 136-151, Feb. 2011.

[10] J. Kacprzyk, A. Wilbik, S. Zadrozny, "Linguistic summarization of time series using a fuzzy quantifier driven aggregation," in *Fuzzy Sets and Systems*, vol. 159, pp. 1485–1499, 2008.

[11] K. Hirota and W. Pedrycz, “Fuzzy computing for data mining,” in *Proc. of IEEE*, vol. 87, no. 9, pp. 1575–1600, Sep. 1999.

[12] R. Yager, “A new approach to the summarization of data,” in *Information Sciences.*, vol. 28, pp. 69–86, 1982.

[13] J. Kacprzyk, S. Zadrozny "Computing With Words Is an Implementable Paradigm: Fuzzy Queries, Linguistic Data Summaries, and Natural-Language Generation," in *IEEE Transactions On Fuzzy Systems*, vol. 18, no. 3, pp. 461-472, June 2010.

[14] F.M. Pouzols, A. Barriga, D.R. Lopez, S. Sanchez-Solano, "Linguistic summarization of network traffic flows, " in *Proc. of IEEE International Conference on Fuzzy Systems 2008*, pp 619-624, June 2008.

[15] M. Ros, M. Pegalajar, M. Delgado, A. Vila, D. T. Anderson, J. M. Keller, M. Popescu, "Linguistic summarization of long-term trends for understanding change in human behavior," in *Proc. of 2011 IEEE International Conference on Fuzzy Systems*, pp. 2080-2087, 27-30 June 2011.

[16] R. Castillo-Ortega, N. Mann, D. Sanchez, "Linguistic local change comparison of time series," in *Proc. of 2011 IEEE International Conference on Fuzzy Systems*, pp. 2909-2915, June 2011.

[17] F. Diaz-Hermida, A. Bugarin, "Semi-fuzzy quantifiers as a tool for building linguistic summaries of data patterns," in *Proc. of IEEE Symposium on Foundations of Computational Intelligence*, pp. 45-52, April 2011.

[18] L. A. Zadeh, "Fuzzy Sets," *Inf. Control*, vol. 8, pp. 338-353, 1965.

[19] L. X. Wang, J. M. Mendel , "Generating fuzzy rules by learning from examples," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, Nov/Dec 1992.

[20] A. A. Freitas , "On rule interestingness measures," in *Knowledge-Based Systems*, vol. 12, pp. 309-315, 1999.

[21] H. Ishibuchi, T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," in *IEEE transactions on fuzzy systems*, vol. 13, no. 4, pp. 428-435, Aug. 2005.

[22] A. Laha, "Developing credit scoring models with SOM and fuzzy rule based k-NN classifiers," in *Proc. of IEEE International Conference on Fuzzy Systems*, pp. 692-698, July 2006.

[23] T. C. Havens, J.M. Keller, M. Popescu, "Computing With Words With the Ontological Self-Organizing Map," in *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 473-485, June 2010.

[24] C. Pateritsas, S. Modes, A. Stafylopatis, "Extracting Rules From Trained Self-Organizing Maps," in *Proc. of IADIS International Conference Applied Computing*, pp. 183-190, 2007.

[25] W. S. van Heerden, A. P. Engelbrecht, "HybridSOM A Generic Rule Extraction Framework for Self-Organizing Feature Maps," in *Proc. of IEEE Symposium on Computational Intelligence and Data Mining*, pp. 17-24, April 2009.

[26] T. Nomura, T. Miyoshi, "An Adaptive Fuzzy Rule Extraction Using Hybrid Model of the Fuzzy Self-Organizing Map and the Genetic Algorithm with Numerical Chromosomes," in *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol.6, no.1, Jan 1998.

[27] T. Weijters, A. van den Bosch, H. J. van den Herik, "Intelligible Neural Networks with BP-SOM," in *Proc. of the 9th Dutch Conference on Artificial Intelligence*, University of Antwerp, p.27-36, November 1997.

[28] K. W. Wong, T. D. Gedeon, C. C. Fung, P. M. Wong, "Fuzzy Rules Extraction Using Self-Organizing Neural Network and Association Rules," in *Proc. of IEEE Region 10 International Conference on Electrical and Electronic Technology*, vol. 1, pp. 403-408, Aug. 2001.

[29] D. Plikynas, L. Simanauskas, A. Rasteniene, "Portable Rule Extraction Method for Neural Network Decisions Reasoning," in *Journal of Systemics, Cybernetics and Informatics*, vol. 3, no. 4, pp. 79-84, 2005.

[30] J. Malone, K. McGarry, S. Wermter, C. Bowerman, "Data mining using rule extraction from Kohonen self-organizing maps," in *Neural Computing & Applications*, vol. 15, no. 1, pp. 9-17, 2006.

[31] C. Hung, "Knowledge-Based Rule Extraction from Self-Organizing Maps," in *Proc. of 15th Intl. Conf. on Neural Information Processing of the Asia-Pacific Neural Network Assembly*, Nov. 2008.

[32] D. Wijayasekara, O Linda, M. Manic, "SOM-LS : Linguistic Summarization of Data Using SOM," in *Information Sciences*, in Review.

[33] T. Kohonen, “Automatic Formation of Topological Maps of Patterns in a Self-Organizing System, ” in *Proc. SCIA, E. Oja, O. Simula, Eds.* Helsinki, Finland, pp. 214-220, 1981.

[34] S. Haykin, *Neural Networks and Learning Machines – Third Edition*, Prentice Hall, 2008.

[35] D. Yu, H. Shen, J. Yang, "SOMRuler: A Novel Interpretable Transmembrane Helices Predictor," *IEEE Transactions on NanoBioscience,* , vol. 10, no. 2, pp. 121-129, June 2011.

[36] A. Asuncion, D. J. Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.