2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)

A Novel Tool (FP-KC) for Handle the Three Main Dimensions Reduction and Association Rule Mining

Samaher Hussein Ali
Department of Software, college of Information Technology
University of Babylon
Hilla, Iraq (00964)
Samaher_hussein@yahoo.com

Abstract—**: This work attempt to developing the FP-Growth data mining algorithm through use several knowledge constructions to build up a novel tool called Frequency Pattern-Knowledge Constructions (FP-KC) to find the association rules and to satisfy the goal of dimension reduction methods is using the correlation structure among the predicator variables by reduction the main three dimensions (features, samples and value of features). FP-KC attempts to combine between the features of principle component analysis and frequency pattern growth. This done using the three criteria (Eigenvalue, cumulative variability and Scree plot). There are many reasons for developing the FP-Growth data mining algorithm in build up a novel algorithm FP-KC to find the association rules: (a) the size of an FP-tree is typically smaller than the size of the uncompressed data because many records in dataset often share a few items in common.(b) Given the best result, if all the records have the same set of items, and this point always satisfy in the scientific dataset. (c) FP-growth is an efficient algorithm because it illustrates how a compact representation of the transaction data set helps to efficiently generate frequent item sets. (d) The run-time performance of FP-growth depends on the compaction factor of the data set. The performance of FP-KC test using five huge databases including (Primate splice-junction gene sequences, Diabetes, DNA, GIS and Watermarking). The confidence' degree of the all association rules yield by FP-KC is equal to 95%.**

Keywords- ***PCA; FP-Growth; Knowledge Constructions; Watermarking Database.***

I. INTRODUCTION

The choice of data representation, and selection, reduction or transformation of features is probably the most important issue that determines the quality of an intelligent data analysis solution. Besides influencing the nature of a KDD algorithm, it can be determined whether the problem is solvable at all, or how powerful the resulting model of KDD is. A large number of features can make available samples of data relatively insufficient for mining. In practice, the number of features can be as many as several hundreds. If we have only a few hundred samples for analysis, dimensionality reduction is required in order for any reliable model to be mined or to be of any practical use. On the other hand, data overload, because of high dimensionality, can make some data-mining algorithms non applicable, and the only solution is again a reduction of data dimensions. The three main dimensions of preprocessed data sets, usually represented in the form of flat files, are columns (features), rows (cases or samples), and values of the features [9]. But how one can combine among these three main dimensions of preprocessed data sets without loss any inform is still open problem as explain in Fig. 1. [1]

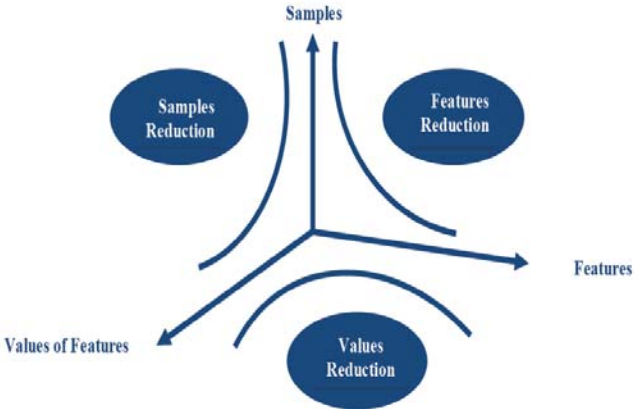

Figure 1. Relation among the Three Main Dimensions

Dimension reduction methods have the goal of using the correlation structure among the predictor variables to accomplish the following[5]:

- To reduce the number of predictor components
- To help ensure that these components are independent
- To provide a framework for interpretability of the results.

.

II. CURRENT STATUS

The data mining process requires highly computations when evolving large data sets. Dimensionality Reducing (the number of attributed or the number of records) reduce this computations. This step is closely related (often dimensional reduction is used as a step in feature extraction), but the goals can differ. Dimensional reduction has a long history as a method for data visualization, and for extracting key low dimensional features. It is including wavelet transforms and principal components analysis, clustering, sampling. The main Taxonomies of dimension reduction problem are decreasing

learning cost, increasing learning performance, reduction of irrelevant dimension, reduction of redundant dimension.

Hussein K.,2002[7] suggests algorithm to discover terminated item sets(DOTI) and he explains how can maintenance the terminated item sets to extract dynamic rule miner.

Mutthew G. and Larry B., 2003 [9] describe a method of feature construction and selection using the traditional genetic programming, genetic algorithm and stander c4.5 on number of databases to improve the classification performance of the well-known induction algorithm c4.5.

Nian Yan., 2004[11] introduced the classification of data mining approaches and focused on back propagation neural network and its enhanced applications. The multilayer neural network has been applied in building the classification model. There are two data sets used for the study. HIV data set from bioinformatics research and credit card data set for the risk management. A pre-training process is designed to construct the neural network classifier fast also proposed a new ensemble method(performance weighted ensemble based on the p value and has proved the strongpoint of it compared to the traditional ensemble method).

Mahdi ,2005[8] suggests a technique that can discover knowledge from any database via soft computing techniques which involve fuzzy set, neural network and genetic algorithm by build DBRule Extractor system.

Samaher, 2008[13] presents a method to design a programming system using hybrid techniques represented by soft computing and data mining to discover knowledge in databases. This work proposes a fuzzy c-mean model for attributes clustering. The genetic algorithm is used to determine which of features are most predictive after that; radial basis functions are embedded in two-layer neural which topology is used in real applications to classify the database records.

Christopher, 2010[2] showed a tutorial overview of several geometric methods of feature extraction and dimensional reduction. He divides the methods into projective methods and methods that model the manifold on which the data lies. For projective methods, he review projection pursuit, principal component analysis (PCA), kernel PCA, probabilistic PCA, and oriented PCA; and for the manifold methods, he review multidimensional scaling (MDS), landmark MDS, Isomap, locally linear embedding, Laplacian eigenmaps and spectral clustering

The Table I presents the comparison among the different works mentioned previously according to (Authors, tools used in these studies, types of datasets that handle by these studies, kind of preprocessing, and natural of results).

TABLE I. COMPARISONS AMONG THE PREVIOUS DIMENSIONAL REDUCTION STUDIES

Authors	Tools	Data set	Preprocessing	Results
Hussein [Huss02]	DOTI	Business	Find frequency itemset	Extract Dynamic Rule Miner
Mutthew, et al. [Mutt03]	GP , GA, and C4.5	Multivariate	GP(feature contraction), GA(feature selection)	Classification
Nian Yan [Nian04]	BPNN	Bioinformatics (HIV), Credit care	Remove irrelevant attributes	Classification
Mahdi [Mahd05]	Fuzzy set, NN and GA	Multivariate	Fuzzy set (covert description attributes), GA(find best seed for each cluster)	DBRule Extractor System
Samaher [Sama08b]	FCM, GA, and RBFN	Multivariate	FCM(Clustering attribute) GA(find best feature)	Classification
Christopher [Chri10]	Geometric Methods	Multivariate	pursuit, PCA, kernel PCA, probabilistic PCA, and oriented PCA	Overview of several geometric methods for feature extraction and dimensional reduction

III. FP-GROWTH ALGORITHM

FP-growth is a method of mining frequent item sets with-out candidate generation. It constructs a highly compact data structure (an *FP-tree*) to compress the original transaction database. Rather than employing the generate-and-test strategy of Apriori-like methods, it focuses on frequent pattern (fragment) growth, which avoids costly candidate generation, resulting in greater efficiency. In addition, FP-growth Algorithm can be define as powerful computational tools in a generation association rules compare with A priori algorithm. It is base on FP-tree.

FP-Growth adopts a divide and conquer strategy by (1) compressing the database representing frequent items into a structure called FP-tree (frequent pattern tree) that retains all the essential information and (2) dividing the compressed database into a set of conditional databases, each associated with one frequent item set and mining each one separately. It scans the database only twice. In the first scan, all the frequent items and their support counts (frequencies) are derived and they are sorted in the order of descending support count in each transaction. In the second scan, items in each transaction are merged into an FP-tree and items (nodes) that appear in common in different transactions are counted. Each node is associated with an item and its count.

Nodes with the same label are linked by a pointer called a node-link. Since items are sorted in the descending order of frequency, nodes closer to the root of the FP tree are shared by more transactions, thus resulting in a very compact representation that stores all the necessary information. Pattern growth algorithm works on FP-tree by choosing an item in the order of increasing frequency and extracting frequent item sets that contain the chosen item by recursively calling itself on the conditional FP-tree, that is, FP-tree conditioned to the chosen item.

FP-growth is an order of magnitude faster than the original Apriori algorithm.

FP-growth is a seminal algorithm proposed in this thesis for mining frequent item sets for association rules.

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

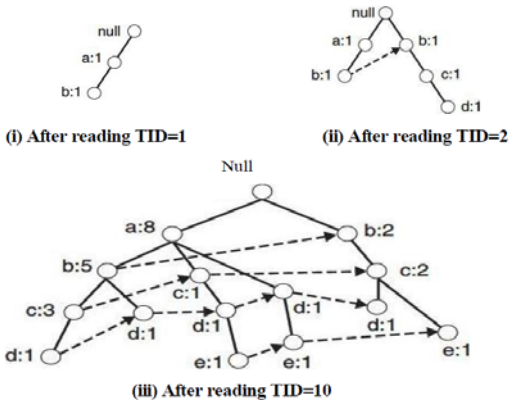

Figure 2. Construction of an FP-tree [12]

Fig.2 shows a data set that contains ten transactions and five items. Each node in the tree contains the label of an item along with a counter that shows the number of transactions mapped onto the given path. Initially, the FP-tree contains only the root node represented by the null symbol. The FP-tree is subsequently extended as in the following [12] :

The data set is scanned once to determine the support count of each item. Infrequent items are discarded, while the frequent items are sorted in decreasing support counts. For the data set shown in Fig. 2, *a* is the most frequent item, followed by b, c, *d,* and e.

The algorithm makes a second pass over the data to construct the FP-tree. After reading the first transaction, *{a,b},* the nodes labeled as *a* and *b* are created. A path is then formed from **null** —> *a* —> *b* to encode the transaction. Every node along the path has a frequency count of 1.

After reading the second transaction, {b,c,d}, a new set of nodes is created for items *b, c,* and *d.* A path is then formed to represent the transaction by connecting the nodes **null** —> *b* —> *c* —> *d.* Every node along this path also has a frequency count equal to one. Although the first two transactions have an item in common, which is *b,* their paths are disjoint because the transactions do not share a common prefix.

The third transaction, *{a,c,d,e},* shares a common prefix item (which is a) with the first transaction. As a result, the path for the third transaction, **null** —> *a* —> *c* —> *d* —> *e,* overlaps with the path for the first transaction, **null** —>*a* —> *b.* Because of their overlapping path, the frequency count for node *a* is incremented to two, while the frequency counts for the newly created nodes, c, *d,* and e, are equal to one.

This process continues until every transaction has been mapped onto one of the paths given in the FP-tree. The resulting FP-tree after reading all the transactions is shown at the bottom of Fig. 2.

In this work, we are suggest developing of the FP-Growth data mining algorithm to build up a novel tool to find the association rules called Frequency Pattern-Knowledge Constructions (FP-KC). Mainly for many reasons

- **First:** The size of an FP-tree is typically smaller than the size of the uncompressed data because many records in dataset often share a few items in common.
- **Second:** Given the good results, if all the records have the same set of items, and this point always satisfy in the scientific dataset.
- **Third:** FP-growth is an interesting algorithm because it illustrates how a compact representation of the transaction data set helps to efficiently generate frequent item sets**.**
- **Forth:** The run-time performance of FP-growth depends on the compaction factor of the data set.

As a result, FP-growth finds all the frequent item sets ending with a particular suffix by employing a divide-and-conquer strategy to split the problem into smaller sub problems.

IV. PRINCIPAL COMPONENTS ANALYSIS

It is not easily to satisfy the relation among the main components of Fig.1 "samples reduction, features reduction and values of feature reduction). But, in this work, we try to establish this relation by using the suitable tools and suggest the FP-KC algorithm to verify this goal. At beginning, we can use analyze the records of database and find the interest components by using PCA. After that, passing the result to FP-KC algorithm to verify these relations. In order to compute the PCA we can follow the following steps [Dian06]:

Input: Process database

Output: PC-database

• **Step1:** Compute the standardized data matrix **Z** =[Z1, Z2, . . . , Zm] base on

Zi = (Xi − μi) /σii from the original dataset.

• **Step2:** Compute the Eigen values:

- if B be an m × m matrix, and let I be the m × m identity matrix (diagonal matrix with 1's on the diagonal). Then the scalars (numbers of dimension 1 × 1) λ1, λ1, . . . , λm are said to be the Eigenvalues of B if they satisfy |B − λI| = 0.

• **Step3:** Compute Eigenvectors:

- Let **B** be an m × m matrix, and let λ be an Eigenvalues of B. Then nonzero m × 1 vector **e** is said to be an eigenvector of B if Be = λe.

• **Step4:** Compute ith principal components:

- The ith principal component of the standardized data matrix **Z** = [Z1, Z2, . . . , Zm] is given by Yi = $\mathbf{e}^T$i**Z**,

• **Step5:** End PCA procedure

Where **e**i refers to the ith eigenvector (discussed below) and $\mathbf{e}^T$i refers to the transpose of **e**i . The principal components are linear combinations Y1, Y2, . . . , Yk of the standardized variables in Z such that (1) the variances of the Yi are as large as possible, and (2) the Yi are uncorrelated. The first principal component is the linear combination Y1 = $\mathbf{e}^T$1Z = e11Z1 + e12Z2 + ··· + e1m Zm, which has greater variability than any other possible linear combination of the Z variables. Thus:

- The first principal component is the linear combination Y1 = eT1Z, which maximizes Var(Y1) = eT1ρ e1.
- The second principal component is the linear combination Y2 = eT2Z, which is independent of Y1 and maximizes Var(Y2) = eT2ρ e2.
- The ith principal component is the linear combination Yi = e_iX, which is independent of all the other principal components Yj , j < i , and maximizes var(Yi) = eTiρ ei.

There are several methods to specify a certain association rules. One of these methods depend on finding frequency itemset, such as traditional Apriori, FP-growth are described in rules are given which specifies the attributes that determine membership of the target. While the goal of this part of thesis is not only determining the association rules but also finding a solution of the three dimension reduction at the same time. Therefore, this part combines between the result of PCA and develop FP-growth called FP-KC.

V. A NOVEL TOOL FP-KC

The goal of dimension reduction methods is to use the correlation structure among the predicator variables to reduce the three main dimensions (features, samples and value of features). But how we can combine among these three dimensions without losing any important information it is still as one of the challenge in KDD [Bara10]. The following points show how the novel tool FP-KC deals with this problem and find new solutions of it. These points relate of the Table 1. The following points can be attaching as new row of that table.

- **Tools:** FP-KC
- **Data set:** Multivariate
- **Preprocessing:** Original PCA, Knowledge Constructions (eigenvalue criterion, proportion of variance explained criterion and scree plot criterion)**.**
- **Result**: Set of Association Rules**.**

The main steps of the propose tool can be describe as follow with the block diagram of FP-KC show in Fig.3.

FP-KC Algorithm

Input: Principle Component (PC) Database**,** Set of knowledge construction**,** min-sup

Output: set of association rules

• **Step1:** Construct the FP-tree

- Scan the PCDatabase
- Collect F, the set of frequent items and their support counts
- Sort F in support count descending order as L, the list of frequent items
- Create the root of an FP-tree and label it as "Null"

• **Step2:** Test the Knowledge Constrictions Conditions

For each record in PCdatabase test the KC as follow

- If the record not verification eigenvalue criterion

  Then (remove the record from PCdatabase).
- Else, the record not verify proportion of variance explained criterion Then (remove the record from PCdatabase).
- Else, the record not verification The scree plot criterion Then (remove the record from PCdatabase). Else, Goto Step3.

• **Step3:** Built FP-KC

- For each record verifying Knowledge Constrictions Conditions
- Select and sort the frequent items in records according to the order of L
- Call insert-Tree procedure

• **Step4:** Mining the FP-KC using FP-growth procedure

- If Tree contain single path then

▪ For each combination of nodes CN in path

- Generate pattern CN U α
- Support_count=min_support count of nodes

- Else, if tree contain multi paths

▪ For each $node_i$ in the header of tree

- Generate pattern CN = $node_i$ U α
- Support_count= $node_i$ _support count of nodes
- Construct CN condition pattern then CN conditional FP-$Tree_{CN}$
- If $Tree_{CN}$ < > ϕ Then : go to step 4.

• **Step5:** End FP-KC Algorithm

Procedure of Insert-Tree

- *Step1: if Tree has child N Then*
  - *Set i=i+1*
  - *Else, Create New Node(N)*
  - *i=1; N.link=Tree*
  - *End if*
- **Step2: if remind list< > ϕ Then**
  - *for j=1 to No. of element*
  - *Call insert-tree(P,N)*
  - *Next j*
  - *End If*

VI. Experiments on the Watermarking Dataset

A. Description

1.Title: IVC-DWT vs DTCWT (Watermarking) Database

2. Sources:

- Creators: all examples taken from LIRMM (Laboratoire d'information de Robotique et de Microelectronique de Montpellier)
- Donor: Methaq T. Gataa
- Date: 21/6/2011

3. Relevant Information Paragraph:

Problem Description: Twelve Original images and one hundred twenty distorted images were generated by two watermarking algorithms with five different embedding strength.

- Discreet Wavelet Transform (DWT)
- Dual Tree Complex Wavelet Transform (DTCWT).

4. Number of Instances: 120

5.. Number of Attributes: 7
- class (one of Mos1, Mos2, Mos3, Mos4 and Mos5)
- six features(PSNR, WPSNR, WSNR, SSIM, UQI and C4)

6. Missing Attribute Values: No

7. Attribute information:

IVC database provides MOS values by using Double Stimulus Impairment Scale method with 5 categories (Excellent, Good, Fair, Poor and Bad)

# Attribute	Description:
1-6	six fields are main features PSNR, WPSNR, WSNR, SSIM, UQI and C4
7	one field represent the class that take one of five types Mos1, Mos2, Mos3, Mos4 and Mos5

8. Class Distribution:
Mos1: 27 (23%)
Mos2: 24 (20%)
Mos3: 22 (18%)
Mos4: 34 (29%)
Mos5: 13 (10%)

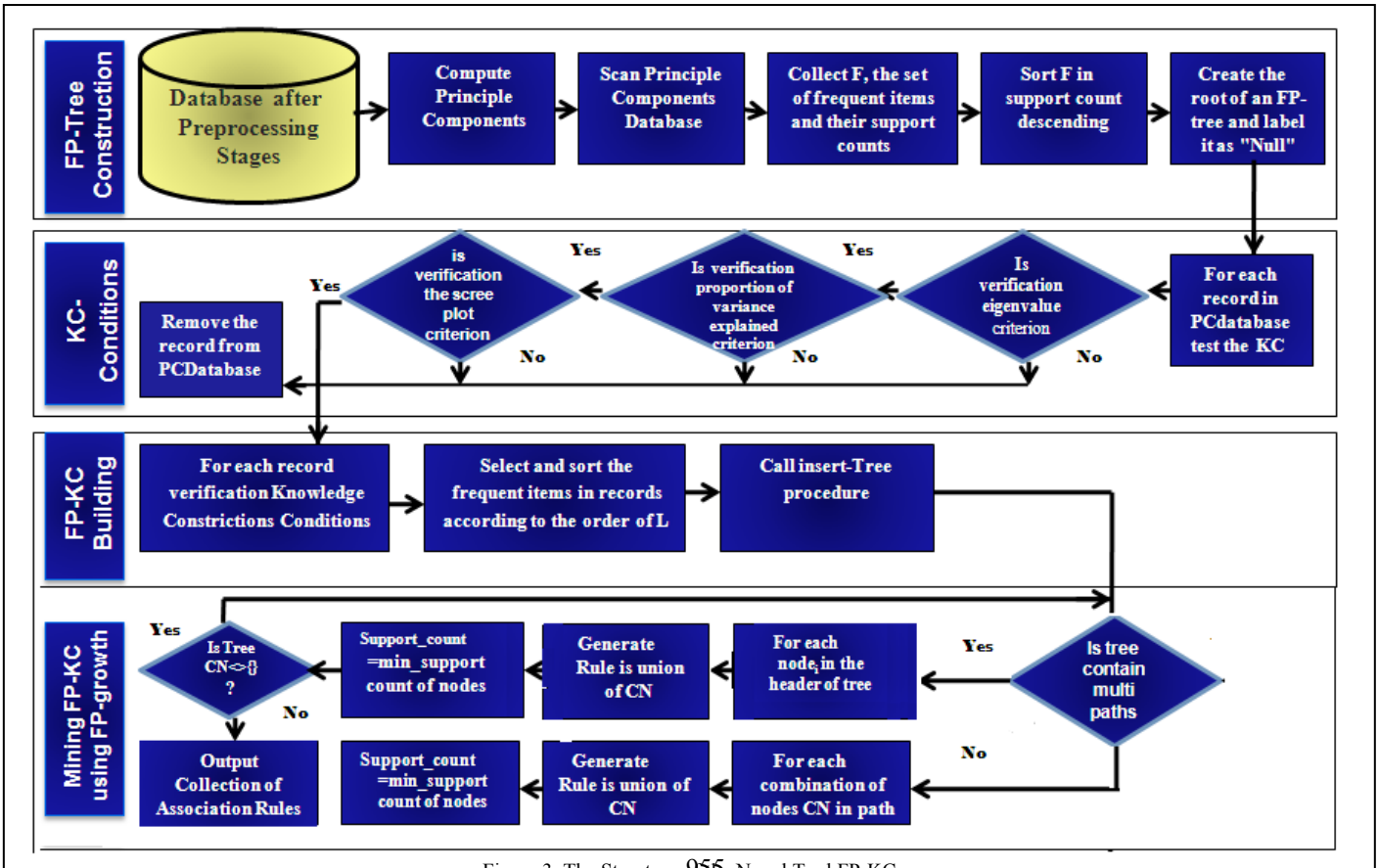

Figure 3. The Structure of the Novel Tool FP-KC

B. Result .A

First: compute the standardized data matrix Base on the mean and stander division of columns of dataset Zi = (Xi − μi) /σii.

Second: We can compute Eigenvalues as in Table II.

TABLE II. EIGENVALUES OF THE STANDARDIZED DATASET

	F1	F2	F3	F4	F5	F6
Eigenvalue	**5,525**	0,359	0,060	0,033	0,019	0,004
Variability (%)	92,089	5,983	0,997	0,546	0,325	0,061
Cumulative %	92,089	98,072	99,068	99,614	99,939	100,000

The value of Eigenvalues greater than one for a given dataset is mentioned in bold font. This means the features of that Eigenvalues is using in the FP-KC algorithm for a given dataset. Where the number of these features is one from the total six Features. Figure 4 shows the relationship of Eigenvalues and cumulative variability**.**

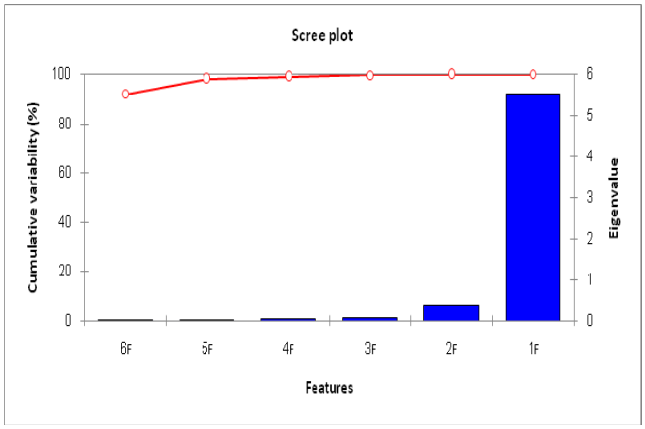

Figure 4. Relation of Eigenvalue and Cumulative variability

Third: we compute Eigenvectors as in Table III.

TABLE III. EIGENVECTORS OF EIGENVALUES MORE THAN ONE

	F1	F2	F3	F4	F5	F6
PSNR	0,413	-0,373	0,015	-0,444	0,067	0,699
WPSNR	0,416	-0,312	0,057	-0,461	-0,207	-0,686
WSNR	0,404	-0,450	-0,420	0,675	0,019	-0,044
SSIM	0,404	0,481	-0,289	-0,128	0,700	-0,124
UQI	0,398	0,561	-0,219	0,029	-0,674	0,152
C4	0,415	0,116	0,830	0,343	0,089	0,009

Fourth: Compute principal components as show in Table IV.

TABLE IV. PRINCIPAL COMPONENTS MATRIX

	F1	F2	F3	F4	F5	F6
PSNR	0,970	-0,224	0,004	-0,080	0,009	0,042
WPSNR	0,977	-0,187	0,014	-0,083	-0,029	-0,042
WSNR	0,950	-0,270	-0,103	0,122	0,003	-0,003
SSIM	0,950	0,288	-0,071	-0,023	0,098	-0,007
UQI	0,935	0,336	-0,054	0,005	-0,094	0,009
C4	0,975	0,070	0,203	0,062	0,012	0,001

Fifth: Set the Main Parameters of FP-KC

- Remove all the factors not verification eigenvalue criterion (i.e., the eigenvalue less than one).
- Remove all the factors not verification proportion of variance explained criterion (i.e., the value of Cumulative variability large than 99,068).
- Remove all the factors not verification The scree plot criterion (i.e., the maximum number of components that should be extracted is just prior *to* where the plot begins to straighten out into a horizontal line.
- Set min_sup=2
- Set min confidence=95%, where

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

Support _count is frequency of occurrence of an itemset**.**

Sixth: Generation the Assoaction Rules base on the Main Split Tree

- **Rule1:** If WPSNR <= 34.194 Then class = 1
- **Rule2:** If WPSNR > 34.194 and WPSNR <= 37.645 and WSNR <= 32,17 Then class = 2
- **Rule3:**If WPSNR > 34.194 and WPSNR <= 37.645 and WSNR > 32,17 and WSNR <= 35.464 Then class = 1
- **Rule4:** If WPSNR > 34.194 and WPSNR <= 37.645 and WSNR > 35.464 Then class = 2
- **Rule5:**If WPSNR > 37.645 and WPSNR <= 41.699 and WSNR <= 38.139 and UQI <= 0,9065 Then class = 2

- **Rule6:** If WPSNR > 37.645 and WPSNR <= 41.699 and WSNR <= 38.139 and UQI > 0,9065 and C4 <= 0,910167 Then class = 2
- **Rule7:** If WPSNR > 37.645 and WPSNR <= 41.699 and WSNR <= 38.139 and UQI > 0,9065 and C4 > 0,910167 and C4 <= 0,938965 Then class = 3
- **Rule8:** If WPSNR > 37.645 and WPSNR <= 41.699 and WSNR <= 38.139 and UQI > 0,9065 and C4 > 0,938965 Then class = 2
- **Rule9:** If WPSNR > 37.645 and WPSNR <= 41.699 and WSNR > 38.139 Then class = 2
- **Rule10:** If WPSNR > 41.699 and WPSNR <= 45,02 and UQI <= 0,9685 Then class = 3
- **Rule11:** If WPSNR > 41.699 and WPSNR <= 45,02 and UQI > 0,9685 Then class = 4
- **Rule12:** If WPSNR > 45,02 and UQI <= 0,9855 and WSNR <= 47.558 Then class = 4
- **Rule13:** If WPSNR > 45,02 and UQI <= 0,9855 and WSNR > 47.558 Then class = 5
- **Rule14:** If UQI > 0,9855 and WPSNR > 45,02 and WPSNR <= 46.242 Then class = 4
- **Rule15:** If UQI > 0,9855 and WPSNR > 46.242 Then class = 5

After the previous six steps, the dataset have 120 record and 7 features become knowledge base have 15 rule and 4 features. Fig.5 show the surface of Watermarking database base on the main three features more important in that database .

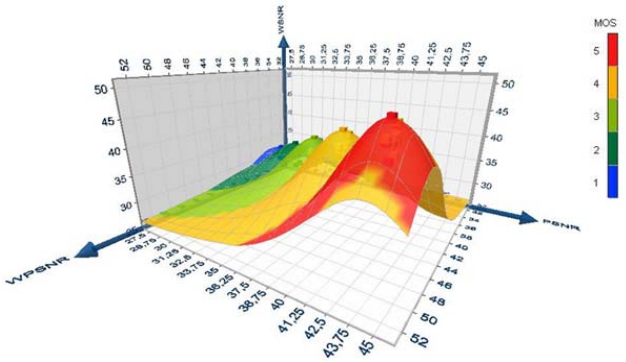

Figure 5. *Surface of IVC-DWT vs DTCWT (Watermarking) Database*

In this work, we are using five huge databases to test the performance of the FP-KC, these databases including (Primate splice-junction gene sequences, Diabetes, DNA, GIS and Watermarking). The description of these databases and the results of FP-KC explained in Table V. All the association rules generated by a novel tool FP-KC of that datasets explained in appendix. Addition, the Relationship among Eigenvalues, Cumulative and Scree Plot of each database explain in Figures 6, 7, 8, and 9.

TABLE V.THE DESCRIPTION THE FEATURES OF ORIGINAL DATABASES

Name of Database	Attribute Characteristics	# Instances	# Association Rules Generation by FP-KC	# Attributes	# Attributes Generation by FP-KC	Area
Splice-junction Gene Sequences [Spli09]	Categorical	3190	39	61	13	Life
Diabetes [Diab10]	Integer, Real	767	10	9	4	Health
DNA [DNA92]	Binary	1186	36	181	15	Life
GIS [GIS10]	Integer, Real	1001	25	9	5	Geographic
Watermarking [Wate11]	Real, Categorical	120	15	6	4	Image

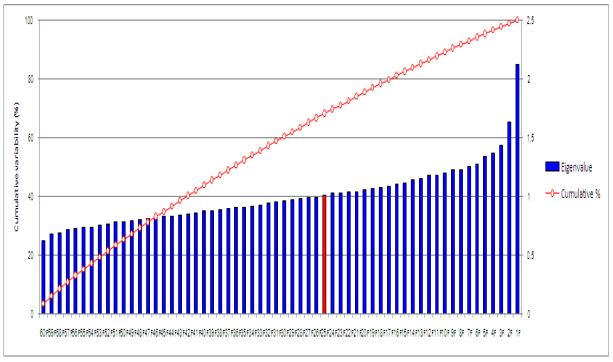

Figure 6. Relation of Eigenvalues and Cumulative and Scree Plot of Primate splice-junction gene sequences

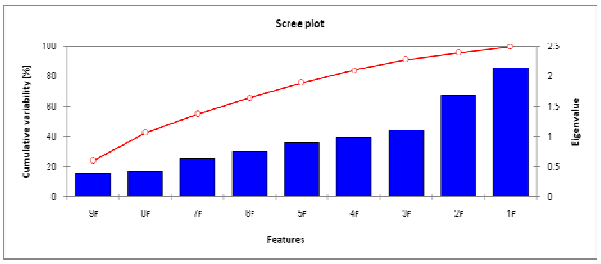

Figure 7. Relationship among Eigenvalues, Cumulative and Scree Plot of Diabetes

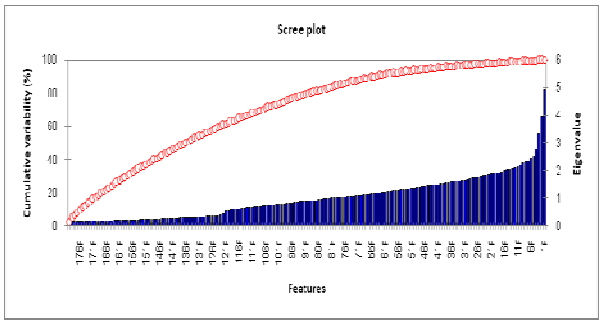

Figure 8. Relationship among Eigenvalues, Cumulative and Scree Plot of DNA Database

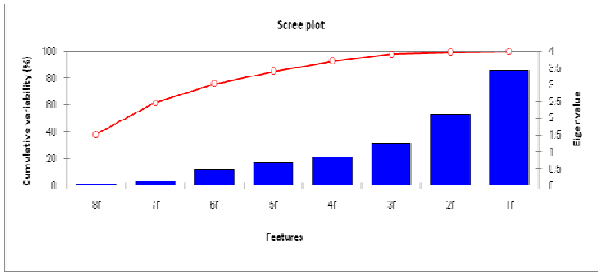

Figure 9. Relationship among Eigenvalues, Cumulative and Scree Plot of GIS Database

VI. Conclusions

In This work, we attempt to handle the three dimensions reduction problem, by proposes cooperation tool between PCA and FP-growth, these tool called FP-KC. The generated FP-KC can get accurate association rules and compress the dataset in three dimensions (number of features" by PCA", number of records and features "by FP- Growth" and value of features through standardization matrix). The confidence' degree of the all association rules yield by FP-KC is equal to 95%.

There are many reasons for developing the FP-Growth data mining algorithm in build up a novel tool FP-KC to find the association rules explained in section two from this paper.

As a result, this work success to satisfy the goal of dimension reduction methods is using the correlation structure among the predicator variables to reduction the three main dimensions (features, samples and value of features). FP-KC is combining between the features of principle component analysis and frequency pattern growth through using three criteria including Eigenvalue, cumulative variability and Scree plot related to PCA as knowledge contractions of FP-growth algorithm.

By experiments, we found FP-KC given good compression results therefore, we suggest exploiting this point in the compression the digital video clip tracks and images because FP-KC provides with important information as association rules

APPENDIX

A. Association Rules generation by FP-KC of Primate splice-junction gene sequences database

- **Rule1:** If A30 <= -0.75 and A35 <= -0.75 and A31 <= -0.75 and A32 <= -0.75 Then A26 <= -0.75
- **Rule2**: If A30 <= -0.75 and A35 <= -0.75 and A31 <= -0.75 and A32 <= -0.75 Then A26 > -0.75
- **Rule3:** If A30 <= -0.75 and A35 <= -0.75 and A31 <= -0.75 and A32 > -0.75 Then A32 <= -0.25
- **Rule4**: If A30 <= -0.75 and A35 <= -0.75 and A31 <= -0.75 and A32 > -0.25 Then A29 <= 0.25
- **Rule5**: If A30 <= -0.75 and A35 <= -0.75 and A31 <= -0.75 and A32 > -0.25 Then A29 > 0.25
- **Rule6**: If A30 <= -0.75 and A35 <= -0.75 and A31 > -0.75 Then A29 <= 0.25
- **Rule7**: If A30 <= -0.75 and A35 <= -0.75 and A31 > -0.75 and A29 > 0.25 Then A28 <= -0.75
- **Rule8**: If A30 <= -0.75 and A35 <= -0.75 and A31 > -0.75 and A29 > 0.25 Then A28 > -0.75
- **Rule9:** If A30 <= -0.75 and A35 > -0.75 and A29 <= 0.25 and A31 <= -0.75 Then A33 <= 0.25
- **Rule10**: If A30 <= -0.75 and A35 > -0.75 and A29 <= 0.25 and A31 <= -0.75 and A33 > 0.25 Then A34 <= -0.25
- **Rule11**: If A30 <= -0.75 and A35 > -0.75 and A29 <= 0.25 and A31 <= -0.75 and A33 > 0.25 Then A34 > -0.25
- **Rule12**: If A30 <= -0.75 and A35 > -0.75 and A29 <= 0.25 Then A31 > -0.75
- **Rule13**: If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 <= -0.75 and A33 <= 0.25 Then A22 <= -0.75
- **Rule14**: If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 <= -0.75 and A33 <= 0.25 and A22 > -0.75 Then A31 <= -0.75
- **Rule15:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 <= -0.75 and A33 <= 0.25 and A22 > -0.75 Then A31 > -0.75
- **Rule16**: If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 <= -0.75 and A33 > 0.25 Then A31 <= -0.75
- **Rule17:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 <= -0.75 and A33 > 0.25 Then A31 > -0.75
- **Rule18:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 <= -0.75 and A31 <= -0.75 Then A34 <= -0.25
- **Rule19**: If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 <= -0.75 and A31 <= -0.75 and A34 > -0.25 Then A32 <= -0.75
- **Rule20**: If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 <= -0.75 and A31 <= -0.75 and A34 > -0.25 Then A32 > -0.75
- **Rule21:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 <= -0.75 Then A31 > -0.75
- **Rule22:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 > -0.75 Then A21 <= 0.25
- **Rule23:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 > 0.25 Then A33 <= 0.25
- **Rule24:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 > 0.25 and A33 > 0.25 Then A31 <= -0.75

- **Rule25:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > -0.75 and A28 <= 0.25 and A21 > 0.25 and A33 > 0.25 Then terminal A31 > -0.75
- **Rule26:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 <= -0.25 Then A31 <= -0.75
- **Rule27:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 <= -0.25 Then A31 > -0.75
- **Rule28:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 > -0.25 Then A24 <= -0.75
- **Rule29:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 > -0.25 and A24 > -0.75 and A20 <= 0.25 Then A21 <= -0.75
- **Rule30:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 > -0.25 and A24 > -0.75 and A20 <= 0.25 Then A21 > -0.75
- **Rule31:** If A30 <= -0.75 and A35 > -0.75 and A29 > 0.25 and A28 > 0.25 and A32 > -0.25 and A24 > -0.75 Then A20 > 0.25
- **Rule32**: If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 Then A32 <= -0.75
- **Rule33:** If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 and A32 > -0.75 and A32 <= -0.25 and A34 <= 0.25 and A33 <= 0.25 Then A22 <= -0.75
- **Rule34:** If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 and A32 > -0.75 and A32 <= -0.25 and A34 <= 0.25 and A33 <= 0.25 Then A22 > -0.75
- **Rule35:** If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 and A32 > -0.75 and A32 <= -0.25 and A34 <= 0.25 Then A33 > 0.25
- **Rule36:** If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 and A32 > -0.75 and A32 <= -0.25 Then A34 > 0.25
- **Rule37:** If A30 > -0.75 and A31 <= -0.75 and A35 <= -0.75 Then A32 > -0.25
- **Rule38:** If A30 > -0.75 and A31 <= -0.75 Then A35 > -0.75
- **Rule39:** If A30 > -0.75 Then A31 > -0.75

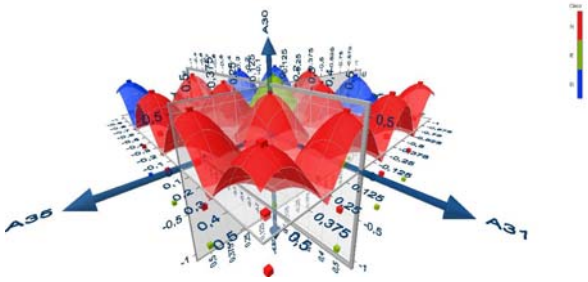

Figure 10. Surface of Primate splice-junction gene sequences Database base on the main three features (A30, A35 and A31)

B. Association Rules generation by FP-KC of Diabetes database

- **Rule1:** if PLASMAG <= 127.5 and AGE <= 28.5 Then class = 1
- **Rule2 :** if AGE > 28.5 and PLASMAG <= 99.5 Then class = 1
- **Rule3:** if AGE > 28.5 and PLASMAG > 99.5 and PLASMAG <= 127.5 and PEDIGREE <= 0.202 Then class = 1
- **Rule4:** if AGE > 28.5 and PLASMAG > 99.5 and PLASMAG <= 127.5 and PEDIGREE > 0.202 and BODYMASS <= 34.75 Then class = 2
- **Rule5:** if AGE > 28.5 and PLASMAG > 99.5 and PLASMAG <= 127.5 and PEDIGREE > 0.202 and BODYMASS > 34.75 and BODYMASS <= 2.665E+008 Then class = 1
- **Rule6:** if AGE > 28.5 and PLASMAG > 99.5 and PLASMAG <= 127.5 and PEDIGREE > 0.202 and BODYMASS > 2.665E+008 Then class = 2
- **Rule7:** if PLASMAG > 127.5 and PLASMAG <= 154.5 and BODYMASS <= 9.8E+007 Then class = 2
- **Rule8:** if PLASMAG > 127.5 and PLASMAG <= 154.5 and BODYMASS > 9.8E+007 and BODYMASS <= 3.02E+008 Then class = 1
- **Rule9:** if PLASMAG > 127.5 and PLASMAG <= 154.5 and BODYMASS > 3.02E+008 Then class = 2

Rule10: if PLASMAG > 154.5 Then class = 2

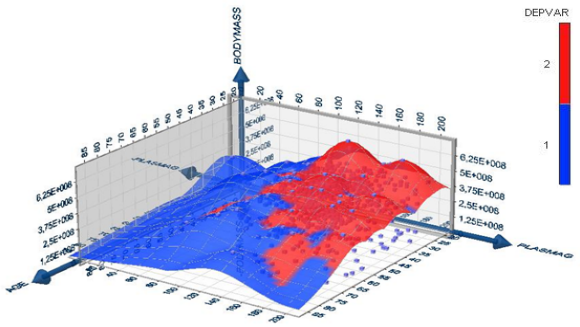

Figure 11. Surface of Diabetes Database

C. Association Rules generation by FP-KC of DNA Database

- **Rule1:** If A85 <= 0,5 and A93 <= 0,5 Then class = 3
- **Rule2:** If A85 <= 0,5 and A93 > 0,5 and A105 <= 0,5 Then class = 3
- **Rule3:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 <= 0,5 and A88 <= 0,5 and A97 <= 0,5 and A72 <= 0,5 Then class = 3
- **Rule4:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 <= 0,5 and A88 <= 0,5 and A97 <= 0,5 and A72 > 0,5 Then class = 1
- **Rule5:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 <= 0,5 and A88 <= 0,5 and A97 > 0,5 Then class = 1
- **Rule6:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 <= 0,5 and A88 > 0,5 Then class = 3
- **Rule7:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 > 0,5 and A95 <= 0,5 and A98 <= 0,5 Then class = 1
- **Rule8:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 > 0,5 and A95 <= 0,5 and A98 > 0,5 Then class = 3
- **Rule9:** If A85 <= 0,5 and A93 > 0,5 and A105 > 0,5 and A100 > 0,5 and A95 > 0,5 Then class = 3
- **Rule10:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 <= 0,5 and A100 <= 0,5 Then class = 2
- **Rule11:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 <= 0,5 and A100 > 0,5 and A97 <= 0,5 Then class = 2
- **Rule12:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 <= 0,5 and A100 > 0,5 and A97 > 0,5 and A94 <= 0,5 and A95 <= 0,5 Then class= 1

- **Rule13:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 <= 0,5 and A100 > 0,5 and A97 > 0,5 and A94 <= 0,5 and A95 > 0,5 Then class= ٢
- **Rule14:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 <= 0,5 and A100 > 0,5 and A97 > 0,5 and A94 > 0,5 Then class = 2
- **Rule15:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 > 0,5 and A93 <= 0,5 Then class = 2
- **Rule16:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 > 0,5 and A93 > 0,5 and A94 <= 0,5 and A100 <= 0,5 and A97 <= 0,5 Then class= 2
- **Rule17:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 > 0,5 and A93 > 0,5 and A94 <= 0,5 and A100 <= 0,5 and A97 > 0,5 Then class= 1
- **Rule18:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 > 0,5 and A93 > 0,5 and A94 <= 0,5 and A100 > 0,5 Then class = 1
- **Rule19:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 <= 0,5 and A63 > 0,5 and A93 > 0,5 and A94 > 0,5 Then class = 2
- **Rule20:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 <= 0,5 and A89 > 0,5 Then class = 3
- **Rule21:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 > 0,5 and A93 <= 0,5 class = 3
- **Rule22:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 > 0,5 and A93 > 0,5 and A94 <= 0,5 and A95 <= 0,5 Then class = 1
- **Rule23:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 > 0,5 and A93 > 0,5 and A94 <= 0,5 and A95 > 0,5 Then class = 2
- **Rule24:** If A85 > 0,5 and A105 <= 0,5 and A88 <= 0,5 and A82 > 0,5 and A93 > 0,5 and A94 > 0,5 Then class = 2
- **Rule25:** If A85 > 0,5 and A105 <= 0,5 and A88 > 0,5 Then class = 3
- **Rule26:** If A85 > 0,5 and A105 > 0,5 and A93 <= 0,5 and A83 <= 0,5 and A82 <= 0,5 Then class = 3
- **Rule27:** If A85 > 0,5 and A105 > 0,5 and A93 <= 0,5 and A83 <= 0,5 and A82 > 0,5 Then class = 2
- **Rule28:** If A85 > 0,5 and A105 > 0,5 and A93 <= 0,5 and A83 > 0,5 and A88 <= 0,5 Then class = 2
- **Rule29:** If A85 > 0,5 and A105 > 0,5 and A93 <= 0,5 and A83 > 0,5 and A88 > 0,5 Then class = 3
- **Rule30:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 <= 0,5 and A100 <= 0,5 and A74 <= 0,5 Then class = 1
- **Rule31:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 <= 0,5 and A100 <= 0,5 and A74 > 0,5 and A83 <= 0,5 and A98 <= 0,5 Then class = 1
- **Rule32:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 <= 0,5 and A100 <= 0,5 and A74 > 0,5 and A83 <= 0,5 and A98 > 0,5 Then class = 2
- **Rule33:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 <= 0,5 and A100 <= 0,5 and A74 > 0,5 and A83 > 0,5 Then class = 2
- **Rule34:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 <= 0,5 and A100 > 0,5 Then class = 1
- **Rule35:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 <= 0,5 and A94 > 0,5 Then class = 2
- **Rule36:** If A85 > 0,5 and A105 > 0,5 and A93 > 0,5 and A95 > 0,5 Then class = 2

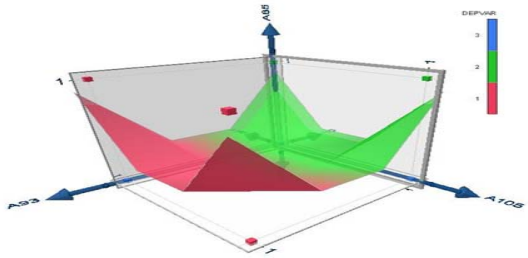

Figure 12 . Surface of DNA Database

D. Association Rules generation by FP-KC of GIS Database

- **Rule1:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 <= 3 and AREA <= 1686.5 Then class = 2
- **Rule2:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 <= 3 and AREA > 1686.5 Then class = 3
- **Rule3:** If TIME_1 > 3 and TIME_1 <= 4.5 and NEURITE_ARBOR_RATIO <= 1.73 and AVG_ARBOR_LENGTH <= 92.345 Then class = 3
- **Rule4:** If TIME_1 > 3 and TIME_1 <= 4.5 and NEURITE_ARBOR_RATIO <= 1.73 and AVG_ARBOR_LENGTH > 92.345 and AVG_ARBOR_LENGTH <= 95.8 Then class = 6
- **Rule5:** If TIME_1 > 3 and TIME_1 <= 4.5 and NEURITE_ARBOR_RATIO <= 1.73 and AVG_ARBOR_LENGTH > 95.8 and AVG_ARBOR_LENGTH <= 167.62 Then class = 1
- **Rule6:** If TIME_1 > 3 and TIME_1 <= 4.5 and AVG_ARBOR_LENGTH > 167.62 and AVG_ARBOR_LENGTH <= 240.13 and NEURITE_ARBOR_RATIO <= 1.64 and AREA <= 2234 Then class = 3
- **Rule7:** If TIME_1 > 3 and TIME_1 <= 4.5 and AVG_ARBOR_LENGTH > 167.62 and AVG_ARBOR_LENGTH <= 240.13 and NEURITE_ARBOR_RATIO <= 1.64 and AREA > 2234 Then class = 6
- **Rule8:** If TIME_1 > 3 and TIME_1 <= 4.5 and AVG_ARBOR_LENGTH > 167.62 and AVG_ARBOR_LENGTH <= 240.13 and NEURITE_ARBOR_RATIO > 1.64 and NEURITE_ARBOR_RATIO <= 1.73 Then class = 1
- **Rule9:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 > 3 and TIME_1 <= 4.5 and AREA <= 1526 and NEURITE_ARBOR_RATIO > 1.73 and NEURITE_ARBOR_RATIO <= 2.15 Then class = 3

- **Rule10:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 > 3 and TIME_1 <= 4.5 and AREA <= 1526 and NEURITE_ARBOR_RATIO > 2.15 Then class = 4
- **Rule11:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 > 3 and TIME_1 <= 4.5 and NEURITE_ARBOR_RATIO > 1.73 and AREA > 1526 and AREA <= 1542.5 Then class = 1
- **Rule12:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 > 3 and TIME_1 <= 4.5 and NEURITE_ARBOR_RATIO > 1.73 and AREA > 1542.5 Then class = 6
- **Rule13:** If AVG_ARBOR_LENGTH <= 240.13 and TIME_1 > 4.5 and TIME_1 <= 5.5 Then class = 2
- **Rule14:** If TIME_1 <= 5.5 and AVG_ARBOR_LENGTH > 240.13 and AREA <= 1468 and ARBORS <= 3.5 Then class = 1
- **Rule15:** If TIME_1 <= 5.5 and AVG_ARBOR_LENGTH > 240.13 and AREA <= 1468 and ARBORS > 3.5 Then class = 4
- **Rule16:** If AVG_ARBOR_LENGTH > 240.13 and AREA > 1468 and TIME_1 <= 4.5 Then class = 6
- **Rule17:** If AVG_ARBOR_LENGTH > 240.13 and AREA > 1468 and TIME_1 > 4.5 and TIME_1 <= 5.5 Then class = 4
- **Rule18:** If AREA <= 1715.5 and TIME_1 > 5.5 and TIME_1 <= 6.5 Then class = 3
- **Rule19:** If AREA <= 1715.5 and TIME_1 > 6.5 and TIME_1 <= 8 Then class = 4
- **Rule20:** If AREA <= 1715.5 and TIME_1 > 8 Then class = 2
- **Rule21:** If TIME_1 > 5.5 and TIME_1 <= 8.5 and AREA > 1715.5 and AREA <= 2679 and NEURITE_ARBOR_RATIO <= 1.385 Then class = 5
- **Rule22:** If AREA > 1715.5 and AREA <= 2679 and NEURITE_ARBOR_RATIO > 1.385 and TIME_1 > 5.5 and TIME_1 <= 6.5 Then class = 3
- **Rule23:** If AREA > 1715.5 and AREA <= 2679 and NEURITE_ARBOR_RATIO > 1.385 and TIME_1 > 6.5 and TIME_1 <= 8.5 Then class = 4
- **Rule24:** If TIME_1 > 5.5 and TIME_1 <= 8.5 and AREA > 2679 Then class = 5
- **Rule25:** If AREA > 1715.5 and TIME_1 > 8.5 Then class = 2

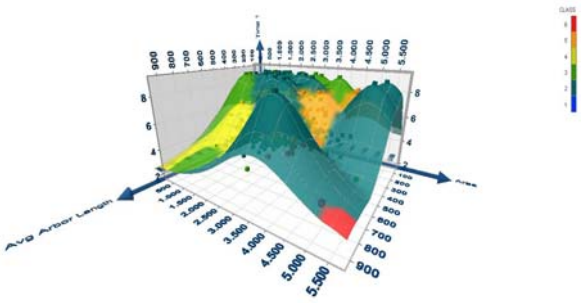

Figure 13. Surface of GIS Database

REFERENCES

[1]Barak Chizi and Oded Maimon," Dimension Reduction and Feature Selection", Data Mining and Knowledge Discovery Handbook, 2nd ed., Springer Science and Business Media, LLC 2010.

[2] Christopher J.C. Burges ," Geometric Methods for Feature Extraction and Dimensional Reduction - A Guided Tour". Data Mining and Knowledge Discovery Handbook, 2nd ed. Springer Science Business Media, LLC . 2010. DOI 10.1007/978-0-387-09823-4_4,

[3] Diagnoses the medical cases, the LUPEM hospital, UK , 2010.

[4] Genbank 64.1 (ftp site: genbank.bio.net)

http://archive.ics.uci.edu/ml/machine-learning datasets/DNA/.1992.

[5] Daniel T. Larose,"Data Mining Methods and Models" Department of Mathematical Sciences Central Connecticut State University 2006.

[6] Center for Machine Learning and Intelligent Systems .USA. http:// ://idke.ruc.edu /GIS/2010.

[7] Hussein K., "Knowledge Discovery in database by using data mining" Ph.D. Thesis, University of Technology, 2002.

[8]Mahdi A.," Extracting Rules from Databases Using Soft Computing", M.Sc Thesis, University of Babylon, 2005.

[9] Mehmed Kantardzic ," Data Mining: Concepts, Models, Methods, and Algorithms" IEEE Computer society, Sponser ,2003.

[10] Mutthew G. and Larry B., "Feature construction and selection using genetic programming and a genetic algorithm", LNCS, P 229-237, 2003.

[11] Nian Yan : Classification Using Neural Network Ensemble with Feature Selection., Ph.D thesis Linköpings university , Sweden, 2004.

[12] Pang-Ning Tan, Michael Steinbach and Vipin Kumar, " Introduction to Data Mining ", University of Minnesota,USA,2006.

[13] Samaher Hussein Ali" Designing a Software for Knowledge Discovery in Database Using Data Mining and Soft Computing Techniques .IEEE, 2nd International Conference: E-Medical Systems October 29-31, 2008

[14] Genbank 64.1 (ftp site: genbank.bio.net).

http://idke.ruc.edu.cn/news/2009/dataset.htm. 2009.