Design of Vector Quantizer for Image Compression
Using Hybrid ANT GLA algorithm

V.S.Jayanthi¹, S.Ashwin¹, A.Shanmugam²

¹Dept.of ECE, Sri Ramakrishna Engineering college, Coimbatore, India.
²Bannariamman Institute of Technology, Sathyamanagalam, India.

ABSTRACT
Compression is a key technique for digital transmission and storage of images due to the vast amount of data associated with it. Image compression algorithms using Vector Quantization (VQ) have been receiving considerable attention. In VQ, minimization of Mean Square Error (MSE) between code book vectors and training vectors is a non-linear problem. Traditional GLA algorithm used for designing the codebooks for Vector Quantizer converges to a local minimum, which depends on the initial code book. Ant colony algorithm is a population-based meta-heuristic search approach that has been receiving increasing attention in the recent years. Ant colony algorithm is inspired by the behavior of real ant colonies. It has shown to be successful and popular for solving optimization problems. In this paper we present a new hybrid approach in which Ant colony algorithm is combined with GLA to design a codebook for vector quantization. Simulation results indicate that ANTGLA has better performance in designing the optimal codebook for Vector Quantizer than GLA. The Peak Signal to Noise Ratio (PSNR) is used as an objective measure of reconstructed image quality.

Keywords: Vector quantization, ant colony optimization, the global optimum, image compression, codebook design

1. INTRODUCTION
Digital images are used in applications such as medicine(computer tomography, magnetic resonance imaging, &digital radiology), satellite data, weather prediction, and multimedia software. Uncompressed digital images are comprised of enormous amount of data. Image compression techniques play a role in applications, which require the minimization of the storage, and/or transmission requirements, such as multimedia communications, storage of medical images, archiving of finger prints and transmission of remote sensing images. Generally, image compression techniques exploit certain data redundancies to convert image to a more compact form. Vector Quantization(VQ) [3,6 7] has proven to be an efficient technique for allowing high compression rates in image compression algorithms. Vector quantization, a compression technique that is a multidimensional extension of scalar quantization, partitions the input image to be encoded into two-dimensional vectors, each of which is compared by an encoder with every code vector in a previously defined codebook. The index of the code vector which best matches the input block according to some distortion metric is sent to the decoder, which uses the index to look up reconstruction vector in its copy of the codebook. The reconstruction vectors are tiled to form a lossy compressed version of the input image. It is important to mention that the VQ codebook design is essential in any VQ based image compression technique.

Many optimization problems are fundamentally hard. Essentially, a ‘hard’ problem is one for which solution cannot be guaranteed in a reasonable amount of time.
The ant colony algorithm is adopted from the natural foraging behavior of real ants and has been used to solve the hard optimization problems. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Ants are well known for their complex group behaviors emerging from the cooperative behaviors of many small and simple members. Without any leader or centralized control, an ant colony solves problems that are far beyond the capability of any individual ant, such as finding food or building nest. The success achieved using the ant colony algorithm is due to the introduction of randomness in the search procedure, permitting to escape from local minima and, thus achieving a more globally favorable solution. In this algorithm, the information gathered by simple autonomous mobile agents is shared and exploited to solve the problem.

In this paper, our key focus is to investigate the application of Ant colony algorithms to the NP-hard problem of generating the optimal codebook for Vector Quantizer. This paper is organized as follows: Section 2 describes and outlines the Ant colony algorithms. In section 3 a new approach of designing a codebook for VQ is presented which is based on the combination of Ant colony algorithm and a standard local search strategy. The Ant colony algorithm is responsible for exploring the search space, and the local search algorithm is used to find a local minimum. The feasibility of this approach is demonstrated by presenting the results for some test images. Section 4 discusses the experimental results. Finally Section 5 gives the conclusion.

2. ANT COLONY OPTIMIZATION

Ant colony optimization (ACO), a population-based Meta heuristic approach that can be used to find approximate solutions to difficult optimization problems, is inspired by the behavior of real ants in finding the shortest path from their nest (colony) to a food source without visual sensing. It has been found that a natural optimization process occurs in ant colonies when ants trying to locate food sources and traveling to them. Initially, some ants leave the nest and locate their food source and return to their nest later. At the beginning, they may follow different paths to the food source. After sometime, it can be observed that more and more ants follow a particular path and almost all of them converge to this path ultimately. This path has been found to be the shortest path between the nest and the food source. When the ants attempt to find short paths between their nest and food sources, they communicate indirectly by using pheromone to mark the decisions they made when building their respective paths. That is, the ants while walking lay tiny drops of liquid known as pheromones on their path. These pheromones trails guide other ants to follow the paths taken by previous ants. Ants can smell pheromone and when choosing their way, they tend to choose paths marked by strong pheromone concentrations. When more pheromones are found on a particular path, the probability is high in choosing that path by ants following later. It can be seen that the ants following the shortest path will be back to their nest sooner than the ants taking the longer path. As shorter paths to food will be traversed more quickly, they have a better chance of being sought out and reinforced by other ants before the volatile pheromones evaporate. This will result in having more pheromones in the shortest path that influences other ants also to follow this path. Thus, these pheromones lead to emergence of shortest paths. Although leaving
the pheromone trails by an ant on the path seems to be a primitive behavior, a colony of ants engaging in this primitive behavior will emerge as a source of collective intelligence [1, 2]. This intelligent behavior of ant colonies is the inspiration for artificial ant colonies, which are developed to solve combinatorial optimization problems. By the time the algorithm starts the next iteration, some pheromone evaporation would have taken place. This property helps to avoid stagnation during the process. Pheromone evaporation has also the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. Thus, when one ant finds a good (short, in other words) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leaves all the ants following a single path. The approach of mimicking ant behavior is termed as Ant Colony Optimization, which were originally proposed by Dorigo and colleagues in order to solve difficult combinatorial optimization problems like the traveling salesman problem (TSP) [1,2] and the quadratic assignment problem (QAP) [2] code for Ant colony optimization algorithm. Fig 1 describes the pseudo code for Ant colony optimization algorithm.

3. PROBLEM STATEMENT

A vector Quantizer (VQ) Q of dimension k can be defined as a mapping of data vectors X in k-dimensional Euclidean space $R^k$, into a finite subset C of $R^k$. Thus

$$kQR C \rightarrow$$

Let X be a set of training vectors of size M and dimension k, i.e., $X = \{x_1, x_2, ..., x_M\}$, $x_i R^k \forall i = \{1,2, ..., M\}$. Let Y be a set of code words of size N and dimension k that is. $Y = \{y_1, y_2, ..., y_N\}$, $y_i C \forall i \in \{1,2, ..., N\}$. A data vector x $R^k$ is encoded by identifying the index j of the codevector $y_j \in C$ such that $|| x - y_j || \leq || x - y_i || \forall i \neq j$. The decoder uses the received index j to retrieve the code word from the codebook and generates the reconstruction vector $y_j$ corresponding to x. The distortion measure used is mean square error (MSE) given by $d(x, y_j) = || x - y_j ||^2$. If a VQ minimizes the average distortion, it is called the optimal VQ of size N.

Several algorithms have been proposed to design a codebook using the information provided by the training data. One of the most widely used is Generalized Lloyd algorithm (GLA), which is also known as the k-means algorithm. Design of optimal VQ using GLA has been proposed and studied in [4,6, 7]. GLA algorithm is an iterative gradient descent algorithm that tries to minimize an average squared error distortion measure. It starts with an initial solution, which can be chosen arbitrarily. The existing solution is then improved iteratively using the optimality criteria in turn until a minimum is reached. The algorithm is relatively easy to implement and it gives reasonable results in most cases. Unfortunately the algorithm makes only local changes to the original codebook and it settles with the first local minimum instead of global minimum. Its performance is sensitive to the initialization of the codebook.
One iteration of $N$ ants ($N$ code vectors) finding their respective paths (hence, the codebook) is referred to as a tour. Each tour is operated on $m$ populations of ants, the best of which is followed by an application of the GLA [4,6,7]. Then, the intensity of trail for this population with improved performance codebook is further incremented in a global sense by an amount which is inversely proportional to the minimum distortion of the corresponding codebook. The newly computed pheromone trail is used as the initial values for the next tour.

3.1 CONSTRUCTION OF INITIAL SOLUTION:

We assume that the number of ants is equal to the number of code vectors in the codebook. Initially $N$ ants are placed randomly on different training vectors. Each ant stores the path information, which is the sequence of indices of the training vectors it has followed so far in Tabu list. Tabu $k$ is the dynamically growing vector which contains the Tabu list of the $k^{th}$ ant. The first entry of the Tabu $k$ is the index of the training vector assigned to ant $k$.

The Euclidean distance $d(x_i, y_j)$ is used to measure the distortion of reproducing the input training vector $x_i$ with code vector $y_j$. $d(x_i, y_j) = \|x_i - y_j\|^2$ The average distortion between the codebook and the input image at the $\nu^{th}$ iteration is given by

$$D_{\nu} = \frac{1}{M} \sum_{j=1}^{M} \text{min}_{i} d(x_i, y_j)$$

3.2 CONSTRUCTION OF TOUR

When being on a training vector $x_i$, an ant $k$ chooses a still unvisited training vector at each step based on the transition probability given by

$$P^k_{ij}(t) = \begin{cases} \frac{\tau_{ij}(t) + \rho \min_{j} \Delta \tau_{ij}}{\sum_{j \in \{1,2, ..., M\} \setminus \text{Tabu}_k} \tau_{ij}(t) + \rho \min_{j} \Delta \tau_{ij}} & \text{if } x_i \text{ and } x_j \text{ are visited by the same ant} \\ 0 & \text{Otherwise} \end{cases}$$

where $\eta_{ij}$ denotes the visibility between $x_i$ and $x_j$, The represents the heuristic desirability of choosing $x_j$ when $k^{th}$ ant is in $x_i$, which is defined as the reciprocal of the Euclidean distance between $x_i$ and $x_j$, and $\alpha$ and $\beta$ are positive real parameters whose values determine the relative importance of pheromone trail versus heuristic information (visibility).

3.3 PHEROMONE UPDATING

Once all ants have completed building a tour, the pheromone trails are updated according to the following equation

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta \tau_{ij}$$

where $\rho$ is a coefficient such that $\rho^{-1}$ represents the evaporation the pheromone trail of between and $(\nu + 1)^{th}$ tour.

$\Delta \tau_{ij}$ is given by

$$\Delta \tau_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are visited by the same ant} \\ 0 & \text{Otherwise} \end{cases}$$

The amount $\Delta \tau_{ij}$ of pheromone deposited depends on the quality of the solution found. The smaller the average distortion, the greater the amount of pheromone deposited. The pheromone update is used to increase the pheromone values associated with the good solution and to decrease those that are associated with bad ones. Subsequently the ants utilize the pheromone information.
3.4 THE ANT-GLA ALGORITHM

The ANT-GLA algorithm is as follows:

**Step 0:**
- Initialize the parameters
- Randomly generate codebook for m populations (Place N ants randomly on N different training vectors for each population)
- Set the tour number \( v = 1 \).
- For every pair \( x_i \) and \( x_j \) of ant \( k \), set the pheromone trail intensity \( \tau_{ij} = 0 \) (a small positive constant)

**Step 1:**
For each population

**Step 1.1:**
- Place N ants on N different training vectors that that are the nearest neighbors of the N codewords \( y_j \) respectively.

**Step 1.2:**
- Set \( S = 1 \) (S is the tabu list index)
- For \( k = 1 \) to \( N \) Do
  - Place the index of the starting training vector of the \( k^{th} \) ant in Tabu \( s \)
- Repeat until a pre-specified number of iteration reaches.
  - Set \( S = S + 1 \)
  - For \( k = 1 \) to \( N \) do

**Step 1.3:**
- Choose training vector \( x_j \) to move to with the probability given by equation (3).
- Move the ant \( k \) to training vector \( x_j \)
- Insert the index of the training vector \( x_j \) of the \( k^{th} \) ant in Tabu \( s \)

**Step 2**
- Determine the population with best codebook \( Y(v) \) based on the recorded path information of each ant (the codebook with minimum average distortion)
- Apply GLA for the population with best codebook to recalculate the codebook
- Update the intensity of the pheromone trail using equations ((4) and (5)).

**Step 3**
- Increment the tour number by 1, i.e. \( v := v + 1 \) if the tour number < tour maximum
- Empty all tabu lists.
- Go to Step 1
- else
  - Print the best codebook
  - Stop

4. EXPERIMENTAL RESULTS

To evaluate the validity of the proposed algorithm, ANTMGLA has been implemented in Matlab 7.0 and run on Pentium IV computer, along with GLA, MGLA and ANTGLA. The Lena image of resolution 256 x 256 is used to generate 4096 training vectors of dimension 16 (4 x 4).
The parameters of the algorithm are chosen as $\alpha=1.5$, $\beta=1$, $\lambda=0.99$ and $m=3$. The best codebook in the final tour is selected as the generic codebook consisting of $k$ code vectors with dimension 16 for vector quantization. Further, lossless Huffman encoding is applied to the indices generated by the encoder. Codeword assignment for the indices is based on the frequency distribution of the code vectors in the encoded image, more compression is achieved.

The performance of ANTMGLA is compared with those of the conventional GLA, MGLA and ANTGLA in terms of PSNR and compression ratio. To show the efficiency of the proposed algorithm the objective quality of encoded images are measured using PSNR (Peak value Signal-to-Noise Ratio), which is defined as

$$\text{PSNR} = 10 \log_{10} \left( \frac{\sum_{i,j} (x_{ij} - x'_{ij})^2}{\sum_{i,j} (x_{ij})^2} \right)$$

Where $x_{ij}$ is the value of the $ij$th pixel in the original image and $x'_{ij}$ is that of the reconstructed image.

Experiments were carried out for different codebook sizes (64, 128, 256, 512) with five standard images and the results are summarized in Table 1. For different codebook sizes, the codebooks obtained by ANTGLA outperform those by GLA by 1.09 – 4.62 dB in PSNR. As can be seen from this table for the codebook of different sizes, the percentage improvement of ANTGLA over GLA is above 13.4 for the image "Mona". Thus the effectiveness of ANTGLA is proved. Figs.2-6 show the variation of PSNR with respect to the variation of the size of the code book for the standard images Lena, Mona, rice, Lift and Pout respectively.

Table 1 Performance Comparison Of PSNR In dBs f or GLA and ANTGLA

<table>
<thead>
<tr>
<th>IMAGES</th>
<th>CODEBOOK SIZE = 64</th>
<th>CODEBOOK SIZE=128</th>
<th>CODEBOOK SIZE=256</th>
<th>CODEBOOK SIZE=512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GLA</td>
<td>ANTGLA</td>
<td>GLA</td>
<td>ANTGLA</td>
</tr>
<tr>
<td>Lena</td>
<td>28.24</td>
<td>32.86</td>
<td>30.39</td>
<td>33.06</td>
</tr>
<tr>
<td>Mona</td>
<td>27.7</td>
<td>31.42</td>
<td>28.64</td>
<td>32.55</td>
</tr>
<tr>
<td>Rice</td>
<td>29.26</td>
<td>31.58</td>
<td>30.04</td>
<td>31.94</td>
</tr>
<tr>
<td>Lift</td>
<td>28.43</td>
<td>33.05</td>
<td>31.29</td>
<td>32.43</td>
</tr>
<tr>
<td>Pout</td>
<td>31.94</td>
<td>35.6</td>
<td>34.44</td>
<td>35.65</td>
</tr>
</tbody>
</table>

Fig 2 Codebook size versus PSNR in dB for Lena

Fig 3 Codebook size versus PSNR in dB for Mona

Fig 4 Codebook size versus PSNR in dB for Rice

501
Design of Vector Quantizer for Image Compression
Using Hybrid ANT GLA algorithm

respect to the variation of the size of the code book for
the standard images Lena , Mona, Rice, Lift and Pout
respectively.

Fig 5 Codebook size versus PSNR in dB for lift

Fig 6 Codebook size versus PSNR in dB for Pout

Table 2 compares the compression ratio (CR) achieved
using different algorithms for the five standard images.
Figs. 7-11 show the variation of Compression ratio with

Table 2 Performance Comparison Of Compression ratio for GLA and ANTGLA

<table>
<thead>
<tr>
<th>CODEBOOK SIZE=64</th>
<th>CODEBOOK SIZE=128</th>
<th>CODEBOOK SIZE=256</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGES</td>
<td>GLA</td>
<td>ANTGLA</td>
</tr>
<tr>
<td>Lena</td>
<td>2.56</td>
<td>4.42</td>
</tr>
<tr>
<td>Mona</td>
<td>2.27</td>
<td>3.01</td>
</tr>
<tr>
<td>Rice</td>
<td>2.68</td>
<td>3.75</td>
</tr>
<tr>
<td>Lift</td>
<td>1.48</td>
<td>3.56</td>
</tr>
<tr>
<td>Pout</td>
<td>2.28</td>
<td>3.63</td>
</tr>
</tbody>
</table>

Fig 7 Codebook size vs Compression Ratio for Lena

Fig 8 Codebook size vs Compression Ratio for Mona

Fig 9 Codebook size versus Compression Ratio for Rice

Fig 10 Codebook size versus Compression Ratio for Lift
It is evident from the Table 2 and Figs 7-11 that the compression ratio achieved by the ANTGLA is quite superior to the GLA. Fig 12 shows the original images “Mona”.

Figs 13 – 16 compare the quality of the reconstructed images based on the codebooks of different sizes using algorithms GLA and ANTGLA for the image “Mona”.

From the Figs 2 -16 and Table 1-2, it is inferred that the codebook designed by ANTGLA provides high PSNR and Compression Ratio (CR) than the codebooks generated using GLA for different codebook sizes. This indicates that VQ based on ANTGLA provides better visual quality and competitive compression ratio than other algorithms.
From the figs 2–16 and Table 1–2, it is inferred that the codebook designed by ANTGLA provides high PSNR and Compression Ratio (CR) than the codebooks generated using various algorithms for different codebook sizes. This indicates that VQ based on ANTGLA provides better visual quality and competitive compression ratio than other algorithms.

5. CONCLUSION

This paper proposed a new codebook design for vector quantization based ANTMGLA that has proved to be very effective in image compression. This new codebook design method incorporates both the ant colony algorithm and GLA to find codebook with better performance. The conventional GLA codebook design technique is a kind of steepest descent searching algorithm that is not exempt from the local minima problem and very susceptible to the empty Voronoi region problem. The ANTGLA outperforms the conventional GLA, without significant increase in computation complexity. The feasibility of the proposed approach is demonstrated by presenting the results for five test images. Experimental results show that the new algorithm outperforms other well-known codebook design algorithms, and particularly, the improvement of PSNR exceeds 1.09 dB compared with the conventional GLA algorithm. It is shown that the visual quality of reconstructed images is improved by evolving the codebooks with ANTGLA. The proposed algorithm ANTGLA as a whole achieves compression at low bit rates with good reconstructed quality. Consequently, ANTGLA may offer an alternative approach for locally optimal code book design of VQ.

REFERENCES


AUTHORS

Dr. A. Shanmugam, received the BE degree in 1972 and ME degree in 1978 from Madras University. He received his PhD in computer networks from Bharathiar University in 1994. Currently, he is the Principal, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu. He has been awarded Bharatiya Vidya Bhavan National Award for Best Engineering College Principal for the year 2005 by The Indian Society for Technical Education, New Delhi. He has published more than 150 Journal/conference papers in the area of Electronics and Communication Engineering. His current research interest includes Digital Communication, VLSI Signal Processing, Speech Processing, Image Processing, Optical Communication and Wireless Networking.

Mrs V S Jayanthi is working as Assistant Professor, in Electronics and Communication Engineering Department, Sri Ramakrishna Engineering College Coimbatore, Tamil Nadu, India. She obtained BE in Electronics and Communication Engineering from Government College of Technology, Coimbatore, Tamil Nadu, India in 1985, ME in Applied Electronics from P.S.G College of Technology, Tamil Nadu, India in the year 1991 and currently pursuing PhD at P.S.G College of Technology, Bharathiar University, Tamil Nadu, India. Her research interest includes Digital Communication, VLSI design, Signal Processing, Image Processing, and Soft computing.
Scalable Incremental Dimensional Complexity Algorithm in Gene Sequence Clustering

M. Punithavalli

ABSTRACT

This paper addresses the possibilities of using clustering techniques for gene sequence analysis. Experiments have shown that performance of data clustering algorithms degrade in higher dimensions and can be off by factor of five. This paper proposes a partition based clustering algorithm called scalable incremental dimensional complexity clustering algorithm (SIDC) for clustering the gene sequences. The algorithm is almost similar to k-means clustering algorithm. It will also take several iterations or repetitions for better result. But will converge very fast when compared to the standard k-means algorithm. The enhancement in speed is achieved by a novel incremental dimension selection approach, during the iterative procedure with respect to the total number of dimensions and iterations. Since there is a lot of mathematical operations involved in the distance metric calculations, the reduction in dimension of the data significantly reduce the computation time. The results obtained were analyzed using the t-test and were found to be significant in terms of speed and accuracy (rand index). This results in a new paradigm for Protein and Gene sequence clustering which can be used to visualize the species in a virtual taxonomical space.

Key words: Classification, Clustering, Multi-dimensional data clustering, Gene-sequence clustering.

1. INTRODUCTION

Clustering is used to classify homogeneous and well-separated groups of objects in databases [28]. It is a classical problem in database, artificial intelligence, theoretical literature and it also plays an important role in many fields of business and science. The clustering algorithms differ from each other in terms of their ability to handle different attributes, speed, accuracy etc. However, the clustering algorithm should satisfy some basic features like scalability, dealing with different types of attributes, insensitivity to order of input records, high dimensionality etc. [5]. The various features of the k-Means, fuzzy C-means and hierarchical algorithms make them suitable to be used in a number of applications including gene sequence clustering, document clustering etc.[4][17][6].

Multidimensional data are quite complex to deal with. However, in many real life applications, such as data mining, the data usually reside in very high dimensional space [23]. A number of difficulties are encountered while clustering the high dimensional data. It is because, the clusters are not well separated, due to the fact that high dimensional data often contains a large amount of noise. Also, in most cases the high dimensional data points are not well separated, this can cause the common clustering methods to produce inconsistent results [26]. Due to the large number of dimensions in multidimensional data, the traditional clustering algorithms developed for low dimensional space are found to be ineffective [23].

Director, Department of Computer Science,
Sri Ramakrishna College of Arts and Science for Women, Coimbatore, India
It is found that k-means algorithm tends to generate extremely imbalanced clusters in high dimensional spaces, when the desired number of clusters is large [2].

When the number of dimensions and size of the data base increase, the computational time for clustering also increases. So, time is an important factor for deciding an effective clustering algorithm [4]. Also, it is very difficult to achieve accuracy when multidimensional data are used [3].

2. Background Study
Gene sequence clustering is a new discipline of data mining, most often applied to extraction of useful knowledge from data. It is an emerging technology of Bio-Informatics, which studies the genetic make-up of all living organisms to answer complex biological questions through the development of new information technologies. It is a fast growing and multidisciplinary field which combines information technology, computer science and biology. This empirical approach complements more traditional data analysis.

Sequence scheme consists of a fixed numbers of features; each feature can accommodate one probe. A probe is a string of symbols from the alphabet \( S = \{A, C, G, T, -\} \), where \( - \) denotes the ‘blank’ symbol. It provides information about \( k \)-mers present in the DNA string, but does not provide information about the positions of the \( k \)-mers. Moreover, SP is said to be the spectrum of sequence SEQ, if SP is a multi-set of all \( k \)-long substrings of SEQ, assuming that the number of occurrences of each \( k \)-mer is also known. For example, SEQ = ATGCAGGTCC and SP = \{ ATG, TGC, CAG, GCA, CGT, GTC, TCC, TGC \} [29].

A number of clustering algorithms have been applied for gene sequence clustering. These algorithms can be classified into four categories: partition based approaches, hierarchical approaches, density-based approaches and pattern based approaches.

In Gene based clustering, genes are treated as data objects. Gene expression data are often highly connected and clusters may be highly intersected with each other or even embedded one in another [11][12]. Therefore, algorithms for gene based clustering should be able to effectively handle the situation. The k-means algorithm is a typical partition-based clustering method [16]. Given a pre-specified number \( k \), the algorithm partitions the data set into \( k \) disjoint subsets. Gene expression data typically contain a huge amount of noise; however, the k-means algorithm forces each gene into a cluster, which may cause the algorithm to be sensitive to noise [22][21]. The Self Organizing Map (SOM), another partition based algorithm produces output with the data set that is abundant with irrelevant data points (such as genes with invariant patterns), which will populate the vast majority of clusters [10]. Alon et al split the genes through a divisive approach (Hierarchical) called deterministic-annealing algorithm (DAA) [1][19][18]. Model-based clustering approaches provide a statistical framework to model the cluster structure of gene expression data. The data set is assumed to come from a finite mixture of underlying probability distributions, with each component corresponding to a different cluster [8][27][9][15]. Density-based Hierarchical Clustering (DHC) method is developed based on the notions of density and attraction of data objects [11].

3. Methodology
There are many methods for data classification. Generally the selection of a particular methodology for data classification may depend on the application volume of data and the number of classes present in that data. If the dimension of the data increases, then the problem
becomes more complex and will take a very long time to get a meaningful result. This work uses the Principle Component Analysis for Feature Vector Selection from the Gene Sequence information. The proposed Scalable Incremental Dimensional Complexity (SIDC) algorithm for gene sequence clustering comes under the partition-based clustering.

3.1 Principle Component Analysis [PCA]

Principal components analysis (PCA) is a technique for simplifying a data set. It is a common technique for finding patterns in high dimensional data [13]. It is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. In this paper, Principle Component Analysis (PCA) is used to order the data in feature-wise.

The steps involved in PCA are:

Step1: Obtain the two dimensional Table
Step2: Find out Adjust Matrix
Step3: Calculate the covariance matrix
Step4: Calculate the eigenvectors and forming a feature vector or deriving the new data set.

3.2 Rand Index:

In 1971 Rand proposed the so called Rand Index or Rand measure, a technique for measuring the similarities between the two data clusters. It is also used to assess the performance of different clustering algorithms [7].

Given a set of n objects S = {O₁, ..., Oₙ} and two data clusters of S are X and Y which is to be compare. Let X = {x₁, ..., xₖ} and Y = {y₁, ..., yₗ} where the different partitions of X and Y are disjoint and their union is equal to S. The following values can be computed.

- a is the number of elements in S that are in the same partition in X and Y,
- b is the number of elements in S that are not in the same partition in X and Y,
- c is the number of elements in S that are in the same partition in X but are not in the same partition in Y,
- d is the number of elements in S that are not in the same partition in X but are in the same partition in Y.

Intuitively, one can think of a + b as the number of agreements between X and Y and c + d the number of disagreements between X and Y. Then the rand index, R becomes,

\[ R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}} \]

The rand index has a value between 0 and 1 with 0 indicating that the two data clusters that do not agree on any pair of points and 1 indicating the data clusters that are exactly the same. This research uses the Rand Index to measure the clustering quality.

3.3 t-test Analysis

The t-test is used to perform the statistical analysis for this thesis. This test was given by Sir William Gosset. The t-test is in an application to access whether the mean of a sample drawn from a normal population deviate significantly from a stated value. According to the t-test analysis, if the calculated value modulus t exceeds t 0.05, then the difference between X and μ is significant at 5%
level, if it exceeds \( t \approx 0.01 \), the difference is said to be significant at 1% level. If modulus \( t < 0.05 \), it can be concluded that the difference between \( X \) and \( \mu \) is not significant and hence the sample might have been drawn from a population with mean=\( \mu \) [24].

### 3.3 The steps involved in SIDC Clustering Algorithm for gene sequence clustering

1. Open the Gene/Protein Sequence Database.

2. Select N Sequence of Length ‘D’ from the Database for Clustering and form a N X D Matrix of numbers from it.

3. Create a feature-wise ordered Matrix from the Numerical Equivalent of the Gene Sequence using PCA and get a N X D Ordered Matrix.

4. Select \( k \) Center in the problem space (it can be random).

5. For Iteration \( i \), select \( d \) dimensions with respect to the sigmoid function \( f(i, D) \).

6. Partition the data into \( k \) clusters using \( d \) dimensions of the ordered matrix and grouping the points that are closest to those \( k \) centers. This will provide the data with a new class label.

7. The new centroids for \( k \) clusters are found by computing the mean of \( D \) dimensions.

8. Repeat steps 5 and 7 until mean square error condition or up to the maximum iterations.

9. Project the clustered data in a two dimensional space by using first two principle components.

10. Find the Rand Index using the original class labels and the calculated class labels.

### 3.4 The steps involved in k-means Clustering Algorithm for gene sequence clustering

1. Open the Gene/Protein Sequence Database.

2. Select N Sequence of Length ‘D’ from the Database for Clustering and form a N X D Matrix of numbers from it.

3. Create a feature-wise ordered Matrix from the Numerical Equivalent of the Gene Sequence using PCA and get a N X D Ordered Matrix.

4. Select \( k \) Center in the problem space (it can be random).

5. Partition the data into \( k \) clusters by grouping points that are closest to those \( k \) centers using all the features and assign the class label.

6. Use the mean of these \( k \) clusters to find new centers. For this use all the \( D \) dimensions of the original data.

7. Repeat steps 5 and 7 until mean square error condition or up to the maximum iterations.

8. Project the clustered data in a 2 dimensional space by using first two principle components.

9. Find the Rand Index using the original class labels and the calculated class labels.
Scalable Incremental Dimensional Complexity Algorithm in Gene Sequence Clustering

Open the gene/Protein Sequence

Select N Sequence of Length ‘D’ from the Database for Clustering and Form a N X D Matrix of numbers from it.

Create a feature-wise ordered Matrix from the Numerical Equivalent of the Gene Sequence using PCA and get a N X D Ordered Matrix

Make k number of clusters from the N X D Ordered Matrix using Normal K-Means algorithm and record the time

Make k number of clusters from the N X D Ordered Matrix using Proposed Algorithm and record the time

Project the Clustered Data in the virtual 2 D Taxonomical Space using the first two Principle Components

Find Rand index Using Original Class Labels and Calculated Class Labels from k-Means Clustering

Find Rand index Using Original Class Labels and calculated Class Labels from Proposed Clustering

The Figure 3.1 explains gene sequence clustering in a detail manner:

4 RESULTS AND DISCUSSION

To evaluate the performance of the two algorithms in terms of speed (CPU time) and accuracy (Rand index), a set of gene sequence data is used. The results are discussed in the following paragraphs.

The data set is made up of 200 gene sequences of each 200 characters in length. It belongs to 5 different species; each class of species consists of 40 sequences. This set was prepared from NCBI web site. Since the gene sequence is properly aligned, it is used to measure the speed and accuracy of clustering. The dataset used are

The Total Number of Sequences : 200
The Total Number of Clusters : 5
The Total Number of Dimensions : 50,100,150,200
Number of Repetitions : 20

4.1 Plotting:

The figure 4.1 depicts the plotting of original and k-means, the figure 4.2 depicts the plotting of original and the SIDC clustering, the figure 4.3 depicts the plotting of k-means and SIDC obtained for the above Dataset.

Figure 4.1 Comparison of original and k-means in gene sequence clustering

From the Figure 4.1 it is clear that there is no much difference between the original plotting and k-means plotting. It is also true that the accuracy of clustering is good in k-means clustering.
On comparison with the original plotting, it is found that the SIDC is more or less similar to the original. Thus the clustering accuracy (Rand Index) of SIDC is good.

Figure 4.3 Comparison of k-means and SIDC with data set in gene clustering

By comparing $k$-means and SIDC it is obtained that there is no much difference between them. And it is also noticed that the increase in the clustering dimensional interval gives a significant performance of SIDC than the performance of $k$-means.

4.2 PERFORMANCE WITH RESPECT TO INCREASE IN DIMENSION:

The table 4.1 shows the overall performance results obtained for the above dataset. The time taken for clustering and the accuracy (Rand Index) to cluster is noted for four sets of dimensions with an interval of 50 dimensions. The sequences of 200 genes are used to form 5 clusters in 20 repetitions. This table is used to analyze more results about the performance of the proposed SIDC algorithm.

Table 4.1: Performance of Gene sequence clustering for data set

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Number of Dimensions</th>
<th>Time Taken for Clustering the Gene Sequences</th>
<th>Rand Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$k$-Means</td>
<td>SIDC</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>0.010000</td>
<td>0.010000</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>0.050000</td>
<td>0.050000</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>0.080000</td>
<td>0.060000</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>0.090000</td>
<td>0.070000</td>
</tr>
</tbody>
</table>

It is observed from the above table that the time taken for $k$-means and SIDC are same for the 50 and 100 dimensions and for the 150 and 200 there exists some difference between them. It is also noticed that the time taken for SIDC is less than the $k$-means for the last two but the rand index of SIDC is high, for all the dimensions than the $k$-means.

It is concluded that there is a rise in difference of time taken with the increase in dimensions and the time consumed by the SIDC to form the clusters is less than the $k$-means. The accuracy (Rand index) seems to be high always. Therefore the accuracy is high than the $k$-means and the overall result is, the performance of SIDC is better than the $k$-means, especially in increased dimensions.

SUMMARY STATISTICS:

In this section the summary statistics, the mean, standard deviation and coefficient of variance are calculated and the t-test is carried out between $k$-means and SIDC for increase in dimension to find out the significant level of speed and accuracy for dataset, and it is tested at 5% level of significance. The results are given in the following table.
SUMMARY STATISTICS OF TIME:
The following table 4.2 shows the mean, standard deviation and the coefficient of variance of time for the \( k \)-means and the SIDC.

<table>
<thead>
<tr>
<th>Summary Statistics</th>
<th>( k )-Means</th>
<th>SIDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.058</td>
<td>0.048</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.036</td>
<td>0.026</td>
</tr>
<tr>
<td>Co-efficient of Variance</td>
<td>62.504</td>
<td>55.368</td>
</tr>
</tbody>
</table>

\( t \) value = 0.449, \( p \) value = 0.669

It is observed from the table 4.2 that the mean, standard deviation and coefficient of variance of SIDC are less than \( k \)-means and also the SIDC is more consistent because the coefficient of variance is low. It is also observed by t-test analysis that the \( t \) value is 0.449 and \( p \) value is 0.669 and there exists an insignificant correlation between the normal \( k \)-means and SIDC because the \( p \) value should be less than or equal to 0.05.

It is concluded that there is an insignificant difference between the \( k \)-means and SIDC with respect to time.

HISTOGRAM OF SIDC & \( k \)-MEANS WITH RESPECT TO TIME:
The histogram in figure 4.4 shows the performance of \( k \)-means and SIDC algorithms in terms of Speed with Dataset.

SUMMARY STATISTICS OF ACCURACY:
The table 4.3 shows the mean, standard deviation and the coefficient of variance of accuracy for the \( k \)-means and the SIDC.

<table>
<thead>
<tr>
<th>Summary Statistics</th>
<th>( k )-Means</th>
<th>SIDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.853</td>
<td>0.895</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.036</td>
<td>0.022</td>
</tr>
<tr>
<td>Co-efficient of Variance</td>
<td>4.220</td>
<td>2.458</td>
</tr>
</tbody>
</table>

\( t \) value = -1.968, \( p \) value = 0.097

It is observed from the table that the SIDC mean is high, standard deviation and coefficient of variance is less than \( k \)-means and also the SIDC is more consistent because the coefficient of variance is low. It is also observed by t-test analysis that the \( t \) value is -1.968 and \( p \) value is 0.097 and there exists an insignificant correlation between the normal \( k \)-means and SIDC because the \( p \) value should be less than or equal to 0.05.

It is concluded that there is an insignificant difference in accuracy between the normal \( k \)-means and SIDC.
5 CONCLUSIONS

The application of SIDC and k-means in Gene Sequence Clustering have been implemented and tested successfully. The clustering performance of both the SIDC and K-means has been tested against huge Gene sequence databases gathered from Internet sources. The performance of the algorithms was evaluated in terms of speed (CPU time) and accuracy (Rand index) for a set of gene sequence data. The arrived results were comparable and most promising.

It is concluded that the time consumed by the SIDC to form the clusters is less than the k-means. The rand index seems to be high always therefore the accuracy is high than the k-means and the overall result is, the performance of SIDC is better than the k-means, especially in increased dimensions. Further the algorithm can be extended to handle very huge databases with long sequences.

References


Scalable Incremental Dimensional Complexity Algorithm in Gene Sequence Clustering


Authors

Dr. M. Punithavalli received the Ph.D degree in Computer Science from Alagappa University, Karaikudi in May 2007. She is currently serving as the Director and Head of the Computer Science Department, Sri Ramakrishna college of Arts and Science for Women, Coimbatore. Her research interest lies in the area of Data mining, Genetic Algorithms and Image Processing. She has published 10 Technical papers in International, National Journals and conferences.
Enabling Trust And Privacy Based Access for Grid Services

Sarbjeet Singh, Seema Bawa

ABSTRACT

Grid Computing enables sharing of resources/services distributed over large scale geographically dispersed heterogeneous environments. Security is a big and challenging issue in such an environment as it involves the federation of multiple autonomous administrative domains. The dynamic nature of grid introduces several challenging trust and privacy related issues that require new technical approaches. In this paper we describe different facets associated with these issues and propose trust and privacy models for services/resources exposed in such environment. These models can be incorporated in the authorization framework to provide trust and privacy based access to grid services. The access control decisions can be based on results/information obtained from these models. By incorporating these models into authorization framework, organizations can handle trust and privacy requirements in a flexible and scalable manner. The paper also discusses a prototype implementation of the proposed trust and privacy based authorization framework.

Keywords: Grid Computing, grid services, trust and privacy models, trust and privacy based authorization framework.

1. INTRODUCTION

Grid is a large scale resource sharing and distributed computing environment that couples thousands of computers, storage systems, scientific instruments and other devices distributed over heterogeneous wide area networks [1], [2]. The goal is to enable the creation of a large scale distributed computing environment consisting of Virtual Organizations. Virtual organizations consist of one or more physical organizations or administrative domains. These organizations provide resources/services that can be accessed by other organizations based on their authorization status. The resources/services in an administrative domain are protected by security mechanisms that are local to that domain. In order to access a resource/service, the requester must conform to the set of rules/requirements/policies defined by that resource/service. These rules/requirements/policies are exposed by resources/services in the environment. Such a dynamic environment presents a distinctive set of challenges that are not addressed by traditional client-server based distributed computing environments. A host of security issues need to be addressed to gain wide acceptance. In addition to providing basic security requirements like authentication, authorization, confidentiality and integrity, grid security architecture must be able to support more advanced security features like dynamic delegation of access rights, single sign-on/sing-off, dynamic establishment of trust relationships among multiple domains, privacy and policy related issues in federated environments etc [3-6]. There are several factors that make security hard e.g. user population and
resource pool is large and dynamic, resources and services have different authentication and authorization requirements, job executions span over multiple domains, users have different rights in different domains etc [7-13]. These factors make security a challenging issue. Trust and privacy are two important concerns that we have focused for this paper.

2. Background

Trust is generally defined as having confidence that a resource/service will behave in an expected manner despite the lack of ability to monitor or control the environment in which the resource/service operate [3]. The trust status of a resource/service from a different physical organization is hard to determine as users generally have no idea whether the resource/service is compromised or is malicious [14]. Trust can take a complex form in grid environment. E.g. Users of a particular domain may trust services of domain A and not of domain B, or, they may trust domain A for service-1 but not for service-2, etc. A number of trust scenarios exist in a typical grid environment. The main problem is how to establish and determine the trust relationship of services and resources of a domain with services and resources of other domains.

Besides trust, privacy is another major concern among users and resources/services of different administrative domains. Privacy requirements are generally ignored but to gain wide acceptance, it is necessary to address privacy issues by establishing a privacy model and incorporating that model into the authorization framework [16]. In a Grid Environment, services should be able to draw a clear dividing line between private information and public information of users. Trust of a domain with other domains plays an important role in addressing privacy requirements of users with resources and services of other domains. The main problem is how we can ensure the privacy of data sent to a particular domain that has established a particular type of trust relationship with the source domain.

To address these issues we propose trust and privacy models that can be incorporated into authorization framework to make trust and privacy based access control decisions. The next Section defines different elements of these models. Sections 4 and 5 discuss the design of trust and privacy models. Section 6 describes implementation details and lastly, Section 7 summarizes the work and discusses future plans.

3. Elements of Models

Trust and Privacy Models can be defined as systems that are capable of handling identified trust and privacy related security requirements and a set of use cases. These present a detailed description of all aspects that relate to trust and privacy among entities of different administrative domains. In order to understand the models well, we have identified and defined the following elements:

Subject (SU): Subject is an entity that wants to access services/resources. It can be a user, a grid/web service or any other entity on behalf of user/service.

Service (SR): Service is a piece of software that provides some functionality and can be accessed by Subjects or Services. Services are exposed in the environment and are found by Subjects.

Resource (R): Resource is an object that is accessed by Subjects. It can be a CPU, a storage device, software, data, scientific instrument or any other peripheral. Subjects access Resources through Services. In other words, a Resource is a Service.
Enabling Trust And Privacy Based Access for Grid Services

Service Policy (SP): Service Policy refers to the set of rules/requirements associated with a Service. A Subject must conform to Service Policy in order to Access that Service.

Domain (DO): Domain refers to the set of Subjects and Services under a unique Domain Policy (DP). The Services in a Domain are provided by Service Providers and they may belong to same or different physical organizations/institutions.

Domain Policy (DP): Domain Policy refers to the set of rules/regulations/requirements that an entity must conform to in order to be in that Domain.

Access (AC): Access is an operation that a Subject performs on Resource/Service. The access is provided based on conformance to Security Policy (SP) that is associated with that Resource/Service.

Policy (PO): Policy is a set of rules/requirements that can be associated with a Subject/Service/Domain. It can be represented as

\[ PO = (AP, AuP, TP, PP, MP, OP) \]

where

- AP is Authentication Policy
- AuP is Authorization Policy
- TP is Trust Policy
- PP is Privacy Policy
- MP is Management Policy
- and OP refers to any Other Policy

Service Policy and Domain Policy, both are subsets of Policy, i.e. \( SP \subseteq PO \) and \( DP \subseteq PO \).

Filter: The rights/privileges of a Subject are different in different Domains. Filter is a component through which rights/privileges of a Subject are filtered for a particular Domain. There are two types of filters: Filter-in and Filter-out. These will be explained later in this section.

MAP (MAP): is an operation that maps/transforms the identity of a Subject of one administrative domain to Subject in another administrative domain. E.g. SUi (DOk) \( \mapsto \) SUj (DOl) means that Subject SUi in Domain DOk has been mapped to Subject SUj in Domain DOl.

![Figure 1. Schematic showing Grid Environment consisting of two Domains along with other elements.](image)

In a typical Grid Environment, the elements described above interact in a complex manner. Figure 1 shows a Grid Environment consisting of two Domains (DO1 and DO2) along with other elements of the models. In this Figure Squares represent Subjects, Diamonds represent Resources, Triangles represent Policies and Rectangles represent Filters.

As shown in the figure, Subject’s access request for Service SR first passes through Filter-out component of source Domain and then through Filter-in component of the target Domain. During this passage, Subject’s access rights are filtered for the target Domain. The incoming
request is filtered for the target Domain using Filter-in component. Filter-out filters the rights of an outgoing request from the source Domain for the target Domain. At the target Domain, Subject’s conformance to Service Policy is checked. During this step, trust and privacy policies also get evaluated. If Subject conforms to Service Policy then mapping operation (MAP) is performed to provide the Subject an identity that is local to that Domain. This identity is then used by the Target Domain to provide access to requested Resource/Service. If Subject does not conform to Service Policy, the access is denied.

To discuss the trust and privacy requirements for this type of environment, the following models are proposed.

4. TRUST MODEL

Trust takes a complex form in grid environment. Grid Service requests can span multiple Domains and establishment of trust relationship dynamically among multiple Domains is a complex task [9]. It may be a one time activity per session or it may be dynamic (requiring the establishment of trust for each request). So mechanisms should be there to express and enforce trust and trust policies.

The trust of an entity with other entity is not a fixed value but can change dynamically depending on the behavior of the entity and context in the environment. Trust should be established from the viewpoint of both the parties. Requester’s trust with the service provider may be different from the service provider’s trust with the requester. In a service oriented environment like grid, there are two important types of trust: code trust and execution trust. Execution trust exist from Subject’s side to service provider’s side that service provider will correctly and faithfully allocate Resources for the efficient execution of job with respect to established policies [14]. Code trust exist from service provider’s side to Subject’s side that Subject will generate a legitimate request consisting of virus free code and will not produce malicious results and does not temper other results/information/code present at service provider’s end [14].

There may be other types of trust also, e.g.:

Direct Trust: is the trust that a Subject holds on Service/service provider without any intermediate entity.

Indirect Trust: is the trust that a Subject has on a Service/service provider through some other entity.

Full Trust: A Subject is said to have full trust on a service provider/Domain if it trust all the Services provided by that service provider/Domain.

Partial Trust: A Subject is said to have partial trust on a service provider/Domain if it trust some of the Services provided by that service provider/Domain.

Recommended Trust: is the trust of one entity on second entity that is recommended by other entities.

Authentication Trust: is the trust of an entity on the authenticity of an identity certificate signed by a certificate authority.

Privacy Trust: is the trust of an entity on the privacy features supported by other entity.

Trust Model is a system that can be represented as TS = (EN, TR, OP), where EN refers to different entities in a grid environment among which trust relationships exist. These entities can be Subjects, Services, service providers, Domains etc. TR is the set of trust relationships according to the types of trust discussed above. OP is the set of operators used to express and enforce trust relationships.
Enabling Trust And Privacy Based Access for Grid Services

Trust Relationship can be represented as: \( TR = (SU_1, DO_1, TR, SR_1, DO_2, t, tv) \) i.e. Subject \( SU_1 \) belonging to Domain \( DO_1 \) trust Service \( SR_1 \) belonging to Domain \( DO_2 \) with trust value \( tv \) with respect to trust type \( TR \) for time \( t \).

In order to handle trust related issues, we propose every Domain to implement a Trust Model. This model can be used by Authorization Framework to provide trust based access to grid services. Figure 2 shows a high level view of the Trust Model. Trust Model is implemented as Trust Manager Service.

As shown in Figure 2, access request coming from Subject through Authorization Engine is first intercepted by Trust Evaluator component which is responsible for evaluating request based on established trust relationships and generating trust value \( tv \). To calculate \( tv \), Trust Evaluator make use of Trust Inference Engine. As an example, consider Subject \( SU-1 \) from Domain A wants to access service \( SR-1 \) from Domain B. Let \( s \) and \( f \) denote the success and failure evidences experienced by \( SU-1 \) about \( SR-1 \). Then \( s \) and \( f \) can be calculated as:

\[
\begin{align*}
    s &= s / (s + f + 2) \quad (1) \\
    f &= f / (s + f + 2) \quad (2) \\
    u &= 2 / (s + f + 2) \quad (3)
\end{align*}
\]

where \( u \) is uncertainty about trust. We are making use of these equations to manage trust. These equations play critical role in calculating trust value \( tv \). In case of full trust, \( f \) can be set to 0, in case of partial trust, \( s \) and \( f \) can take different values depending on the level of trust. In case of no trust, \( s \) can be set to 0. A history of these values is maintained by every Domain. Trust Evaluator update trust through Trust Updation component. Trust is updated with every transaction that takes place between a Subject and a Service. Trust Policy base contains established trust relationships and trust management rules. We are making use of this model to express, establish and evaluate trust and trust policies. Trust policies have been expressed in XACML [17]. In XACML, Policy is constructed as a set of rules against the target defined as a triod (Subject, Resource, Action). The integration of Trust Model with the Authorization Framework is explained in Section 6.

5. PRIVACY MODEL

As mentioned earlier, the main problem of privacy is how to ensure the secrecy of data and check misuse of information sent to the target Domain that has established a trust relationship with the source Domain. Though privacy can be guaranteed through XML-Encryption and XML-Signature when information is transmitted over wire but very difficult to achieve once information reaches the target Domain. It is important for Domains to address privacy issues by establishing a Privacy Model and incorporating that model into the Authorization Framework to provide privacy based access to Services.

To address privacy, a variable, say, \( PI \) (Privacy Index) can be attached to the information being sent to indicate its privacy level. Following \( PI \) values can be considered:

\( PI=0: \) No Privacy (information can be used anyway)
PI=1: Partial Privacy (information can be used with permission only)

PI=2: Time Limited Privacy (information can be used with permission but for a limited time period)

PI=3: Full Privacy (can not be used under any circumstances)

The trust of the Subject with the Domain and service provider plays an important role while marking the information with PI. In case of full trust, information can be marked with lower values of PI but in case of partial or unknown trust, higher PI values can be used. We have identified the following privacy elements to implement the Privacy Model:

Subjects: They provide their private/public information to service providers which is necessary to access their Services/Resources.

Information: can be either private information or public information. This information is marked with PI by Subjects according to its sensitivity.

Purpose: tells why the private/public information is required by service provider/Domain. It also tells how the collected information will be used.

Actions: tells what operations can be performed on the collected information.

Conditions: These are privacy statements/policies that describe conditions to be satisfied before access is granted.

Obligations: are activities that must be carried out by service providers, after access, according to established privacy policies. Privacy policies have been expressed in XACML.

A privacy relationship is represented as PR = (SU, IN, PU, ACT, CO, OB) i.e. Subject SU’s information IN can be used for purpose PU only if conditions CO are satisfied and only those actions which are described in ACT can be performed. After access, obligations OB are also performed.

Figure 3. Interaction between a Subject and a Service showing how information is protected and full privacy (PI=3) is maintained.

As an example, consider that a Subject SU wants to access Service SR-1 from Domain A. Further consider that for this, SR-1 require access to Service SR-2 from Domain B on SU’s behalf with SU’s private information. Now SU may not want to provide his private information to SR-1 (say, SU has does not trust SR-1). So SU can decide to use full privacy (PI=3) and send this information with encryption and signature that can be decrypted and verified only by Service SR-2 of Domain B. This interaction is shown in Figure 3.

In this case we are assuming that SU has already established a trust relationship with SR-2. In case of full trust with SR-1, SU may decide to send this information to SR-2 via SR-1 without encryption. Both, SU and SR-1 agree on the privacy relationships beforehand i.e. what information can be disclosed to which party, for what purpose and under which circumstances, along with obligations, if any. This information is stored in the policy.
Enabling Trust And Privacy Based Access for Grid Services

database. The integration of Privacy Model with the Authorization Framework is explained in the next section.

6. IMPLEMENTATION DETAILS

For implementation purpose, we are making use of web services security specifications supported by WSE 3.0. Today the world is witnessing the convergence of grid and web services. The two (grid and web) started far apart in specifications, technology and applications but now they are converging into a common set of standards and specifications. The security requirements of grid services overlap deeply with the security requirements of web services because grid services are stateful web services. Web services security specifications consisting of WS-Security, WS-Policy, WS-Trust etc. have been submitted to OASIS by IBM, Microsoft and other leading organizations [18]. These specifications addresses security issues like how to associate security tokens with messages (WS-Security), how to express constraints and requirements of a web service (WS-Policy), how to request and issue security tokens to establish trust (WS-Trust and WS-Federation), how to establish and share security contexts (WS-SecureConversation) etc. We are making use of these specifications to implement Trust and Privacy Models proposed in Section 4 and 5. Figure 4 shows how Trust and Privacy Models fit into the access control framework. Figure also shows the specifications used for implementation and interaction among different components.

As shown in Figure 4, authorization request coming from subject SU is first intercepted by PEP (Policy Enforcement Point). PEP constructs an authorization decision query and sends it to authorization handler. The result of this query determines if the request is to be granted access to the Service/Resource. The query has details about the identity of the subjects, the service requested and the operations to be performed. Authorization Engine passes this information to PDP (Policy Decision Point). PDP extracts policy information from policy database through PRP (Policy Retrieval Point). If the policy information is not available at PRP, it may be retrieved from Policy Store. The Policies are written by administrator using PAP (Policy Administration Point). Both, the trust and privacy policies have been expressed in XACML. The Policy Store is capable of importing/exporting XACML. In XACML, Policy is constructed as a set of rules against the target defined as a triod (Subject, Resource, Action). PIP (Policy Information Point) is used to retrieve Resource, Subject and Environment attributes. Trust Manager and Privacy Handler implement Trust and Privacy Models as described in Sections 4 and 5.

Figure 4. Trust and Privacy based Access Control Framework

These models provide trust and privacy based access control information to Authorization Engine. Trust model is used to determine the trustworthiness of target domain. If the trust value comes out to be greater than threshold value then the domain is treated as trusted domain and resource/service can be accessed provided incoming request conforms to other security policies also. Privacy model ensures that incoming request is accessing the
resource/service only for the purpose for which resource/service has been exposed and only those actions can be performed which the resource/service provider has allowed on the resource/service. All this information is used by the Authorization Engine to prepare final authorization result. The result is then passed to PEP. Based on the result, PEP then grants/deny access to the requested Service/Resource. During all these steps, the relevant information is stored in log tables to address auditing and accounting requirements. Obligation Service, if any, is also executed by PEP.

All information is exchanged as SOAP messages. While constructing these messages, WS-Security information is embedded to handle encryption and signature requirements. We are using SOAP over HTTP for message communication. For trust establishment and secure context, we are making use of WS-Trust, WS-Federation and WS-SecureConversation specifications. At the target Domain, the SOAP message is analyzed and Subjects credentials and other attributes are checked against Service Policy. At this point the Authorization Framework evaluates the request. Depending on the result of the request, access to the Service/Resource is provided. Here MAP operation is also performed, if required, to provide the Subject an identity that is local to that organization. During all these steps, the relevant information is stored in log tables to address auditing and accounting requirements.

The prototype implementation has shown that the framework is able to meet identified privacy and trust related security requirements. This suggests that the approach is workable and the proposed models can be used to provide privacy and trust based access to grid services.

7. SUMMARY AND FUTURE PLANS
The models provide trust and privacy based access control to grid services. For implementation we are making use of web services standards and specifications as grid services are stateful web services. So these models can be applied to web services also. Currently we have prototype implementation. In future we are planning to use this framework in some real environments. We are also in the process of giving a formal treatment to Authorization Framework based on the proposed Trust and Privacy Models.

8. REFERENCES


AUTHORS

Sarbjeet Singh received his B.Tech degree in Computer Science & Engineering from Punjab Technical University, Punjab, India in 2001 and M.E. degree in Software Engineering from Thapar University, Patiala, India in 2003. He is pursuing his Ph.D in Computer Science & Engineering working on grid security systems architecture at Thapar University, Patiala, India. Currently he is working as a lecturer in Computer Science & Engineering at Panjab University, Chandigarh, India. His research interests include parallel and distributed systems, distributed security architectures, distributed services like grid and web services, privacy and trust related issues etc. He has more than 10 publications in international conferences and journals to his credit.

Seema Bawa received her M.Tech degree in Computer Science from IIT Kharagpur and Ph.D degree in Computer Science and Engineering from Thapar University, Patiala, India. She is currently Professor and Head of Computer Science & Engineering Department at Thapar University, Patiala, India. Her research interests include parallel and distributed computing, VLSI testing, resource discovery, security and scheduling in distributed environments. Prof. Bawa is member of IEEE, ACM, Computer Society of India,
Texture Classification : A Quantitative representation based approach

C.Umarani  S. Radhakrishnan  L.Ganesan

ABSTRACT
A combined statistical and structural approach has been used in this paper for quantitative texture representation. A set of Texture Primitives have been suggested for local representation and the texture primitive spectrum as the global descriptor. Since the occurrence of primitives and their placement rules uniquely define a texture image, the primitive spectrum is also unique, for a texture image. The novelty in this proposal is that the primitives are tested for the presence of texture by performing suitable statistical design of experiments based hypothesis test. These spectrums are shown to be effectively used for unsupervised texture classification for Brodatz and Vistex data bases of texture images. An average correct classification of 97.5% has been obtained and the results are promising.

Key-words : Texture- Primitives - Texture Primitive Spectrum - Un supervised texture classification - minimum distance decision criterion - performance measure.

1. INTRODUCTION
The texture image analysis became so popular and warranted as most of the real life objects, surfaces and remote sensed images are of textured in nature. Main drawback was the lack of proper definition for texture so far. A number of definitions for texture have been proposed and used by various researchers in their context of analysis with their own perspectives. Few schemes for texture analysis have been suggested, either they are based on Statistical or Structural approaches. Using statistical approaches, several schemes have been suggested right from Co-occurrence matrix, run length matrix based, auto correlation, auto regression, MRF based, moments based etc. as found in literature. [4, 6, 8, 9, 15].

In structural approach, the texture region is defined to have a constant texture if a set of local statistics or other local properties of the image are constant, slowly varying or approximately periodic. An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of the primitives. Textures could be rated as coarse, micro, macro, regular, periodic, aperiodic, directional, random, or stochastic[13]. Texture images are analyzed by identifying the local and global properties of the images under consideration. Very few schemes have been suggested which attends both local and global descriptors. One such method is texture spectrum scheme[10].

Observable textures can be characterized by the primitives and placement rules. The main idea of our work is from the intuitive suggestion offered by this fact [5]. Any texture image can be generated by a set of primitives and their placement rules. Conversely any texture image can be identified or perceived if it consists of primitives at regular or random places of occurrences. If it is possible to find out the set of primitives and how they get distributed or the frequency of occurrences of such
primitives, we are succeeding in the representation of texture image. The success of any texture analysis scheme is inferred from the way it captures the textural features and the uniqueness. Since, the textural properties of texture images carry more useful information for the discrimination purposes, it is important to develop features for texture based on suitable quantitative representation of some of these properties. The frequency of occurrences of such primitives is computed for the entire image, the representation becomes global descriptor. Other recent approaches for texture analysis and synthesis may be referred from [3,12,14,16].

The main objective of this paper is to use the proposed local and global descriptor for texture description in the usage of unsupervised texture classification. Initially, a statistical method for detecting the presence of micro texture in a given small image region is discussed. A set of texture primitives which are found to be textured are labeled as local descriptor and the frequency of occurrences of these primitives called texture primitive spectrum is used as the global descriptor. With the proposed set of primitives, number of spectrums have been obtained for various texture images collected from Brodatz Album [2] and Vistex Album[19]. Since most of the real life applications and remote sensed images insist for the classification of different textured regions present without any priori knowledge. Our proposed method has been successful when compared with He and Li Wangs texture number scheme[10] in two respects viz, (i) number of texture features are less and (ii) the time complexity of the proposed method for classification is lesser compared to the texture number scheme.

This paper is organized as follows. Second section explains the model behind our work and the method of detecting the presence of texture by conducting suitable statistical test. A set of primitives along with labeling is presented. In the third section, with the proposed set of primitives textured images are represented by the global descriptor, namely Texture Primitive Spectrum and the experimentation with a set of standard Brodatz Textural Album[2]. In fourth section, texture classification scheme has been explained and the classification results are presented. Finally, the conclusion about our approach and its application for unsupervised texture classification has been highlighted along with further scope of this work.

2. Texture Primitives and Representation

In this section, the presence of texture is detected by proposing a new statistical design of experiments based method for representation of micro texture. This is obtained based on the significant orthogonal effects due to spatial variations of gray levels. In this section, it is shown that an image region represented by a set of orthogonal polynomials and the effects are decomposed into two sets of orthogonal effects. The first set is considered to be contributing towards the presence of texture and the second one is for the noise. Suitable mathematical model and statistical tests for detecting the texture presence is presented[18]. Then a set of primitives are presented which are used as local descriptor. Finally any texture image is represented using the frequency of occurrence of the primitives called the Texture Primitive Spectrum. Various primitive spectrums have been obtained using the set of local descriptors for texture images considered from Brodatz Textural album.

A small image region which is a function of two spatial coordinates, is represented by a set of orthogonal polynomials[7]. In this representation, the image region is considered to be a linear combination of uncorrelated (orthogonal) effects due to spatial variations. The
uncorrelated effects due to the presence of textures have been separated successfully from those due to the presence of Gaussian noise[7]. The presence of texture in the image region under analysis is detected on the basis of the strength of the appropriate orthogonal effects.

In order to characterize texture, the set of orthogonal effects is divided into two disjoint subsets, namely, the set of effects due to the presence of Gaussian noise and the set of interaction effects due to the presence of texture. The criteria for the separability of orthogonal effects due to noise from the effects due to texture can be tested by using the hypothesis that in the presence of texture, the mean square variances corresponding to the orthogonal effects due to the noise are, in fact, estimates of the same noise variance and therefore be used as an estimate of error. These can be tested by computing their divergence in the average variance. If the computed divergence in the average variance is less than the corresponding tabulated value, it is concluded that the divergence is insignificant and the hypothesis is accepted. The method of computing the divergence and the significant values for the divergence of various degrees of freedom are given in[1].

A texture primitive TP_i is defined as a set of more than two adjacent pixels having intensities from (0 to 255) connected by an attribute. Pixels having the same attribute (or within a tolerance limit) contribute to the formation of texture primitive. Or in other words several primitives can be formed by combining different groups of pixels satisfying the attributes within the chosen region. Within (3 x 3) image regions, 92 such primitives are proposed as a set of primitives. These primitives are labeled from TP_1 to TP_92 and are presented in[18]. These primitives are tested using the divergence principle to ensure the presence of texture as depicted in [1].

3. **Texture Representation**

Texture images are considered to have these proposed set of 92 texture primitives. Different textured images look different because of the variations in the presence of such primitives. Two images are considered from Brodatz Textural album which are shown in fig 1. They are Raffia (D84), and Bubbles (D112), each of size (128x128) pixels. The texture primitive spectrums for these two images are shown in fig 2. These can be obtained at different tolerance levels, namely, at 10, 20 and 30. The usage of the primitive spectrums are found successful when using in the texture un supervised texture classification which is discussed in the following section.
4. Unsupervised Texture Classification

By texture classification, it is meant to assign each possible region or a pixel in the image to a class i.e., partitioning the image into mutually exclusive regions and each region corresponds to a particular class of texture. Any texture description scheme is successful only if it has discriminating features. That is, within a texture region the properties or features will be more or less uniform and the properties will be different when the regions are from two different textures. This principle is used for the unsupervised classification. There is no a priori information about the texture class present in the experiment.

As the proposed texture descriptor, primitive spectrum contains the important textural information the same has been used as features during classification. The reference feature vector is considered here as the primitive spectrum of the texture class considered from the top left corner of the target image. A window of larger width (say 30*30 size) is slid over the target image. For each (30*30) region selected, the primitive spectrum is computed. This primitive spectrum is subjected to minimum distance classifier as the input feature vector X computed for the previous region. The texture class, that is assigned by the classifier to the input vector X is the same as that of the previous one if the differences are zero or almost closer to zero. This process gets repeated considering the adjacent regions of the target image. Comparing the feature vector of the present window and the previous window, if the difference is greater than the threshold, then a new region starts to form. Finally when the entire target image is completely scanned, it gets classified into a number of regions. Thus the proposed primitive spectrum has been successfully used for the unsupervised classification.

Experimentation and Results

In the classification experiment four target images are considered, collected from Brodatz databases and are shown in Figure 3. The target images are composed of various textures present. Keeping the tolerance levels at 20, by applying the procedure discussed in the previous section, the target images are classified. The overall classes the target image gets classified are listed in the table 1. and leads to an average correct classification up to 97.5%. The experimentation gets repeated for different tolerance levels both for Brodatz, and the Vistex images. For the sake of comparison of performances with reference to the classification, the experimentation has been attempted for the He and Li Wang’s approach[10] where it is obtained only 84 %. The reason for the good performance is justified by the accuracy in the representation and the apriori identification of texture presence.

Figure 3. TARGET IMAGES (A to D)
Table 1 RESULTS FOR UNSUPERVISED CLASSIFICATION

<table>
<thead>
<tr>
<th>Target Images</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
<th>Class 6</th>
<th>Class 7</th>
<th>Correct Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>56</td>
<td>42</td>
<td>1</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>98</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>42</td>
<td>14</td>
<td>18</td>
<td>24</td>
<td>1</td>
<td>---</td>
<td>98</td>
</tr>
<tr>
<td>C</td>
<td>56</td>
<td>2</td>
<td>41</td>
<td>1</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>97</td>
</tr>
<tr>
<td>D</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>2</td>
<td>16</td>
<td>13</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97</td>
</tr>
</tbody>
</table>

5. Conclusion

A combined statistical and structural approach based quantitative representation for texture description is proposed in this paper. Texture description at local and global stages have been experimented. The usage of the global descriptor has been highlighted by performing un supervised texture classification using popular Brodatz and Vistex textural album. An average correct classification up to 97.5% has been obtained. Currently, the usage of these primitive spectrums in texture segmentation problem is undertaken by the authors.

References


**AUTHORS**

C.Umarani graduated her B.E. in Computer Science and Engineering from Madurai Kamaraj University during the year 1996. Subsequently she qualified her Master degree in Computer Science and Engineering from Madurai Kamaraj University during the year 2000.She is presently working as a faculty in CSE Department at A.C.College of Engineering and Technology, Karaikudi, Tamilnadu. She is pursuing her research in the area of Digital Image Processing from the year 2003.Her field of interests include Data structures and algorithms, Computer Vision and Texture analysis.

Dr. S.Radhakrishnan is presently working as Senior Professor and Head, Department of Computer Science and Engineering at Kalasalingam University, Krishnan oil, Tamilnadu. He has completed his M.Tech. and Ph.D in Biomedical Engineering from Institute of Technology, Banaras Hindu University. He has more than 18 years of experience in teaching and research. He has guided more than 50 M.E./M.Tech. projects and 10 M.Phil. Thesis. His fields of interests are Network Engineering, Computer Applications in medicine and evolutionary computing. He is also serving as Project Director(Network technologies) in TIFAC CORE in Network Engineering at Kalasalingam University. He has more than 25 publications to his credit.

Dr. L.Ganesan is presently working as Head, Department of Computer Science and Engineering, at A.C.College of Engineering and Technology, Karaikudi, Tamilnadu. He has graduated his B.E. in ECE in the year 1983 from Madurai Kamaraj University. Subsequently he has graduated his B.E. in ECE in the year 1983 from Madurai Kamaraj University. Subsequently he has graduated his M.Tech. and Ph.D. from Government College of Technology, Bharthiar University, Coimbatore. He qualified his Doctoral degree from Indian Institute of Technology, Kharagpur, West Bengal, India in the year 1996. His area of interests include Computer Vision, Texture analysis and Software Engineering. He is a life member of ISTE.