Brain Inspired Cognitive Systems
August 29 – September 1, 2004
University of Stirling, Scotland, UK

ANALYSIS OF UNSUPERVISED CLUSTERING BY CROSSING MINIMIZATION

Ahsan Abdullah
National University of Computers & Emerging Sciences
Islamabad, Pakistan
ahsan@nu.edu.pk

ABSTRACT

In [3] it was demonstrated for the first time that crossing minimization of bipartite graphs can be used to perform unsupervised clustering. In this paper, we will present the detailed analysis of the bipartite graph model used to perform unsupervised clustering as in [1, 2, 3]. We will also discuss the effect of data discretization, followed by simulation results demonstrating the noise immunity of the technique.

INTRODUCTION

Our ability and capacity to generate, record and store multidimensional, apparently unstructured data is increasing rapidly in the areas of natural, life and social sciences. Finding that valuable piece of information in a mountain of data is a hard problem. But when you know that you are looking for a needle, finding it in the haystack is easy, as compared to when you don't know what you should be looking for, such as existence of hidden patterns and relationships in data, the formal study of such problems is called *Data Mining*. Data mining problems for large data sets can only be attempted using automatic means. One such interesting Data mining problem is called *clustering*. Clustering is a process of partitioning a set of (data or objects) in a set of meaningful sub-classes or clusters.

There are several techniques and methods for clustering such as Hierarchical, Partition, Graph theoretic, Agglomerative, Artificial Neural Networks and also Simulated Annealing. In [3] it was shown for the first time that graph drawing technique i.e. crossing minimization can be successfully used for unsupervised clustering. Plain crossing minimization technique (CMH) when used for crossing minimization has limited application because of deterioration in the presence of noise. Therefore, in [3] a recursive technique (CRA_CMH) was also proposed resulting in high noise immunity. Both techniques as well as meta techniques [1] using [4] since than have been used to successfully mine biological and agriculture data.

NOMENCLATURE

Knowledge Representation, Information Theoretic Approaches, Statistical and Numerical Aspects.

2.0 DEFINITIONS, AND MODEL FORMULATION

The input data for a clustering problem is typically given in one of the two forms [11]:

- Data matrix (or *object-by-variable structure*) is an $n \times p$ matrix, where corresponding to each of the n objects there are p variables, also called measurements or attributes. Usually $n >> p$.
- Similarity (or dissimilarity) matrix (or *object-by-object structure*) $\mathbf{S}$ is an $n \times n$ symmetric matrix, which contains the pair-wise similarity (or dissimilarity) that is usually computed from p for all pairs of n objects.

Let the given data matrix to consist of interval scaled variables. Using the given data matrix a similarity matrix $\mathbf{S}$ is generated using Pearson's Correlation defined as:

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}.$$

In the context of our problem, X and Y are two rows of the data matrix.

Let $\mathbf{S}$ correspond to the matrix representation of a weighted bipartite graph, such that the vertices correspond to the rows of the data matrix, and the edge weight w is the pair-wise correlation between the rows such that $-1 \le w \le 1$. For working with weighted graphs, weighted MH may be used (problem 10.2 [6]), however, in this study we will be using the CMH developed for simple graphs. To get a simple graph from a weighted graph, we will keep only those edges in $\mathbf{S}$ for which the correlation is higher than a reasonable threshold based on the problem itself. This consequently converts $\mathbf{S}$ to a binary similarity matrix denoted by $\mathbf{S}_B$. Note that the transformation of $\mathbf{S}$ to $\mathbf{S}_B$ results in removal of weak and un-interesting correlations, even noise. However, the disadvantage being, this can also result in loss of actual data too, as we can not differentiate between noise and data, especially for positive correlations.

Let the bipartite graph (bi-graph) corresponding to $\mathbf{S}_B$ be denoted by G_B, most likely G_B will not consist of pure clusters and zero noise. To facilitate defining our model, we first assume that we have a similarity matrix that consists of pure disjoint clusters and zero noise. We denote such a similarity matrix by $\mathbf{S}^*_B$ and the corresponding bi-graph by $G^*_B(V_0, V_1, E)$. G^*_B is a union of $K_{i,j}$ i.e. bipartite graph cliques, and V_0, V_1 is the bipartition of vertices such that $V_0 \cap V_1 = \varnothing$. E is the edge set such that $e = |E|$. As we are considering one-way

clustering, therefore, i = j and n = $|V_1| = |V_0|$ and density of G_B denoted by δ i.e. $\delta(G_B) = e/n^2$.

Let a bi-graph drawing (or layout) of G^*_B be obtained by placing the vertices of V_0 and V_1 on distinct locations on two horizontal lines y = 1 i.e. TOP (top partition) and y = 0 i.e. BOT (bottom partition) in the XY-plane, respectively. The vertices of every clique are located on consecutive and identical x-coordinates for TOP and BOT. Now drawing each edge with one straight-line segment which connects the points on y = 0 and y = 1 where the end vertices of the edges were placed. This will result in a bi-graph drawing, in which only those edges intersect, that belong to the same clique. Let φ^* be the corresponding bi-graph drawing and the order of vertices in the bipartitions V_0 and V_1 is denoted by π^*_0 and π^*_1, respectively. Note that for $K_{i,j}$ (as a convention) we assume that the vertices in bi-partition *i* will be placed in TOP and the vertices in *j* will be placed in BOT. For more on graph drawing reader is referred to [12].

Now we take the first step towards non-ideal data sets and "contaminate" G^*_B by randomly adding edges (white noise) between $v \in V_0$ (i.e. TOP) and $u \in V_1$ (i.e. BOT) with probability $\alpha_E < ½$, similarly, edges are randomly removed (white noise) from each of $K_{i,j}$ with probability $\alpha_I < ½$ resulting in G_B such that $\alpha_E = \alpha_I$. As a second step towards non-ideal data sets, the vertices in each bipartition of G_B are now randomly permuted. The effect of these operations on S_B would be replacement of 1's by 0's and 0's by 1's (the original 0's not the converted 0's) with given probabilities followed by permutation of rows/columns. This is explained with the help of the following graph drawings.

Figure-1(a) shows a G_B with n = 16 and e = 112.

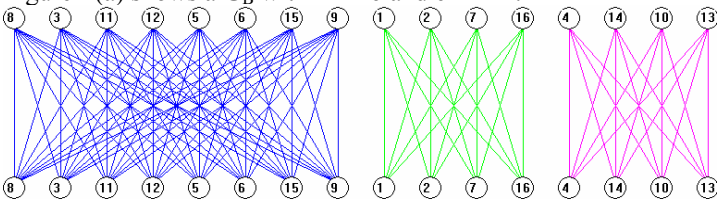

Figure-1(a): $G_B = K_{8,8} \cup K_{4,4} \cup K_{4,4}$ with 856 crossings

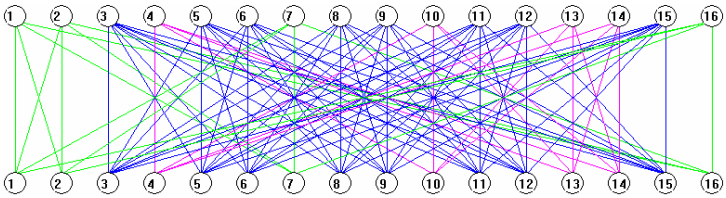

Figure-1(b): Permuted $K_{8,8} \cup K_{4,4} \cup K_{4,4}$ with 1,908 crossings

Figure-1(b) shows the G_B shown in Fig-1(a) with vertices randomly permuted, this being one of the steps towards non-ideal data sets.

Breakdown point β i.e. the minimum proportion of outliers (noise) in a data set that could make an estimator unbounded [8] or the infimum of the proportion of data elements which can be changed without resulting in an arbitrarily-large change in the estimator, taken over all finite data sets X drawn from the underlying distribution.

Let the order of vertices in the bipartitions V_0 and V_1 of G_B (corresponding to the generated similarity matrix) be denoted by π_0 and π_1, respectively. Let $\Phi(G)$ denote the set of all possible bi-graph drawings of the bi-graph G_B, there will be a total of n! such drawings for one-way clustering. Now we define the optimum clustering problem: given a bi-graph G_B (V_0, V_1, E) find φ^* among Φ (G) with relative[1] permutation of vertices π^*_0 and π^*_1.

Let G'_B denote the isomorphic graph as a result of using CRA_CMH on G_B and the corresponding bi-graph drawing be φ'. S'_B is a visualization of the clustering solution generated after running CRA_CMH on G'_B. Note that for *n* and *e* in hundreds φ (similar to Figure-1(b)) is visually meaningless and only S_B makes sense. However, when *n* and *e* are in thousands, screen resolutions prohibit displaying S_B completely and then only extracted results are meaningful.

3.0 DATA DISCRETIZATION

For the data sets being considered, an attribute is either categorical or numeric. Values of a categorical attribute are discrete. Values of a numeric attribute are either discrete or continuous. Discretization is the process of partitioning continuous variables/attributes into categories. For large data sets, discretization results in a reduced representation of the data set that is much smaller in volume yet produces the same (or almost the same) analytical results. In the context of our problem, weighted graph corresponding to S will always be a clique with quadratic edges. Working with S will force us to consider each and every edge of the un-discretized graph, hence a highly undesirable $\Omega(n^2)$ time complexity. Thus the viable way out is discretization of S leading to S_B.

3.1 Discretization Methods

There are basically three major axis of discretization (i) static vs. dynamic (ii) local vs. global and (iii) supervised s. unsupervised. As our work deals with unsupervised clustering, hence we will not consider supervised discretization methods such as Fuzzy discretization, Entropy Minimization discretization etc instead consider unsupervised discretization i.e. making use of information about the distribution of values of attributes without class information. For a detailed comparison of different discretization methods see [13]. The discretization methods of our interest are discussed in the subsequent sub-sections.

3.1.1 Equal Width Discretization (EWD)

EWD divides the number line between v_{min} and v_{max} into *k* intervals of equal width. Thus the intervals have width $w = (v_{max} - v_{min})/k$ and the cut points are at $v_{min} + w$; $v_{min} + 2w \ldots v_{min} + (k$

[1] Relative: vertices may change positions within a clique, but not across the cliques.

- 1)w. k is a user predefined parameter and is set as 2 in our work.

3.1.2 Equal Frequency Discretization (EFD)

EFD divides the sorted values into k intervals so that each interval contains approximately the same number of instances. Thus each interval contains $n=k$ (possibly duplicated) adjacent values. k is a user predefined parameter and is set as 2 in our work.

Both methods surprisingly work well in practice. One reason could be that discretization usually assumes discretized attributes having Dirichlet priors, and 'Perfect Aggregation' of Dirichlets can ensure that discretization appropriately approximates the distribution of a numeric attribute.

4.0 CROSSING MINIMIZATION HEURISTICS

Crossing minimization problem has been studied for over two decades, and its two variants i.e. one-layer and two layers are known to be NP-Complete problems. For a detailed study od different types of crossing minimization heuristics see [12].

4.1.1 Median Heuristic (MH)

It reorders the vertices on the changeable layer (BOT if TOP is fixed or vice-versa) according to the median weight [9]

$$Median(v) = \begin{cases} \frac{1}{2}(P(w_{\lceil \frac{d}{2} \rceil}) + P(w_{\lfloor \frac{d}{2} \rfloor})) & \text{if } d \geq 1 \\ 0.0 & \text{otherwise} \end{cases}$$

where v is a vertex on the changeable layer, d is the number of vertices adjacent to vertex v, and $w_1,..., w_d$ is the sequence of vertices adjacent to v on the fixed layer ordered according to the position P. We have slightly modified it to ensure integer medians for fast (worst case linear time) sorting, and redefined it as follows:

$$Median(v) = \begin{cases} P(w_{\lceil \frac{d}{2} \rceil}) & \text{if } d \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

4.1.2 MaxSort Heuristic

It reorders the vertices on the changeable layer according to the MaxSort weight [4]

$$MaxSort(v) = \begin{cases} Max(P(w)) & \text{if } |adj(v)| \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where v is a vertex on the changeable layer, $adj(v)$ is the set of neighbors of v on the fixed layer, and P(w) is the position of w on the fixed layer. In this work, instead of using MaxSort i.e. sorting on maximum value, we will use its concept MinSort (MS) i.e. sorting on minimum value, as it provides better results as compared to MaxSort. Sorting on minimum value but in descending order is called MinSortReverse (MSR). A variation used is **MinSort Star (MS*)** where **v**ertices with same MinSort weight are sorted in ascending order based on |adj(v)|.

5.0 OUTCOME OF CMH FOR CLUSTERING UNDER DIFFERENT NOISE CONDITIONS

In section 5.1 we will formally prove the optimality of one of the CMH i.e. MH for pure clusters i.e. $\alpha_E = \alpha_I = 0$. In section 5.2 we will discuss the performance of CMH when both α_E and α_I are >0.

5.1 Optimality of CMH for clustering under noiseless conditions

The following theorems are applicable both for one-way as well as two-way clustering, for one-way clustering $|V_1| = |V_0| = n$ i.e. rows of data matrix and $p = q$. Although in this section we will discuss MH, but similar reasoning can be used to show the optimality of Barycenter (BC) and MS for a union of $K_{p,q}$.

Theorem 1

For a $K_{p,q}$ the median of each vertex in the BOT will be $\left\lceil \frac{p}{2} \right\rceil$ and in TOP it will be $\left\lceil \frac{q}{2} \right\rceil$.

Proof:

Every vertex in BOT is of degree q, and each vertex in BOT will have an edge with the vertex at the $\left\lceil \frac{p}{2} \right\rceil^{th}$ position i.e. the median position in the TOP and numbered $\left\lceil \frac{p}{2} \right\rceil$. By virtue of this adjacency, the median of each vertex in the BOT will be $\left\lceil \frac{p}{2} \right\rceil$. Similarly, the median of each vertex in the TOP will be $\left\lceil \frac{q}{2} \right\rceil$. ♦

Theorem 2

For an optimum permutation of vertices for a union of $K_{p,q}$ i.e. G^*_B the medians of vertices are in ascending order in both TOP and BOT partitions.

Proof:

Consider a union of C cliques i.e. $K_{p,q}$ with vertices randomly permuted in TOP and BOT partitions. Let the vertices in TOP be numbered first sequentially from left-to-right. From Thereom-1, there will be C unique median values in the BOT, such that all identical median values will belong to the same clique. Sorting the vertices on their medians will place all vertices belonging to a cluster at consecutive positions. Now vertices in BOT are numbered 1 through n, starting from left to right and then sorting the vertices in TOP based on their median position. Again from Thereom-1, there will be C unique medians in the TOP, such that all identical medians will belong to the same clique.

Consider a clique $K_{i,j}$ whose vertices are now numbered in the TOP and occupy consecutive positions 1 through i. The j vertices of $K_{i,j}$ in BOT will also be at consecutive positions 1 through j and their median value will be $\left\lceil \frac{i}{2} \right\rceil$. Now consider the next set of i' vertices in TOP belonging to clique $K_{i',j'}$. The vertices in the top will be numbered from i + 1 to i' and adjacent to vertices in BOT with median $i + \left\lceil \frac{i'}{2} \right\rceil$. Now $i + \left\lceil \frac{i'}{2} \right\rceil > \left\lceil \frac{i}{2} \right\rceil \forall\, i, i' > 0$, inductively all vertices in the BOT have their medians in ascending order. When this is true, numbering vertices in BOT will result in vertices in TOP with medians in ascending order. Further numbering of vertices in TOP and BOT will not bring any change in their respective positions, indicating an optimal permutation of vertices in TOP and BOT. ♦

Theorem 3
For a union of $K_{p,q}$ if all the medians of vertices are in ascending order in TOP and BOT, then it is an optimum permutation.

Proof:
Let the vertices in TOP be numbered sequentially from left-to-right, such that the first clique picked for numbering is $K_{i,j}$ and its first vertex is assigned number 1. For the medians of vertices in BOT to be in ascending order, the j vertices belonging to $K_{i,j}$ must occupy the first j positions. These positions can only be ensured to remain fixed, if their median is the smallest. This in turn can only be ensured, if the i vertices of $K_{i,j}$ occupy the first i positions in TOP. Without loss of generality, extending this argument, vertices of all cliques are placed, such that there are no crossings between edges belonging to different cliques. Further numbering of vertices in TOP and BOT will not bring any change in their respective positions. This is an optimum permutation of vertices in TOP and BOT as it can not be optimized. ♦

Theorem 4
A permutation of union of $K_{p,q}$ is optimum ⇔ All the vertices of the union of $K_{p,q}$ are sorted in ascending order of their medians.

Proof: Follows from theorem 2 and 3. ♦
Theorem 5
MH terminates with an optimum permutation of a union of $K_{p,q}$.

Proof:
Follows from above theorems and the MH. ♦

Figure-1(b) shows a G_B that consist of three permuted cliques i.e. $K_{8,8} \cup K_{4,4} \cup K_{4,4}$. Figure-1(a) shows the outcome of running any of the CMH resulting in G'_B.

5.2 Performance of CMH for clustering under noise
Consider G_B that is a union of cliques with both α_E and $\alpha_I > 0$ resulting in corruption of the representative value of each vertex. The effect of noise can be minimized (even eliminated), if the true representative value of each vertex can be estimated in the presence of noise. In other words, given a set of points in uni-dimensional space, find a point which best describes the set. When dealing with problems such as above, it is important to consider the issue of *robustness*: how much does the representative value changes if some of the data is disturbed? An example of a non-robust estimator is the *mean*. If any point in the data set is placed at infinity, the *mean* will literally shift to infinity; same is true for maximum and minimum values. This explains the relatively poor performance of BC and MS under noisy conditions when used for clustering.

Theorem 6
For asymptotically large *n*, MH redraws G_B into G^*_B with very high probability, if probability of white noise i.e. $\alpha < 0.5$ (i.e. $\alpha_E = \alpha_I < 0.5$).

Proof:
From theorems 1 and 2 when $\alpha = 0$, a cluster corresponds to all vertices that have the same representative value; therefore, any CMH groups the vertices belonging to a clique. Permuting the vertices may change the representative value of every vertex, however, all the vertices corresponding to a clique will still have the same representative value, thus G^*_B is recovered in just two iterations of any CMH.

However, when $\alpha > 0$, the representative values do change, and that change is dependent on the amount of noise and the robustness of the representative value used. For a sample of size *n* and uni-dimensional data set the breakdown point of the median is $\frac{n+1}{2n}$ for odd *n* or $\frac{1}{2}$ for even *n* because at least half of the points in the sample need to be replaced with sufficiently high or sufficiently low values before the median would be higher or lower than any bound. Intuitively when an estimator of location has breakdown point β, then the estimator would still reveal location even if $\lfloor \beta n \rfloor$ data elements were corrupted. ♦

8.0 COMPLEXITY ANALYSIS OF CRA_CMH
In [12] it was observed that the performance of the crossing minimization heuristics drastically deteriorates as the graphs become sparser. In the context of clustering, we have observed that this is being more profound for MH then other heuristics, such as BC and MS.

Among CMH studied, MH performs best clustering under noise, but only for relatively dense clusters, thus limiting its practical utility under noisy conditions and weak clusters.

These are some of the reasons for developing CRA_CMH i.e. Crossing Minimization with Recursive Application of CMH.

The effect of clustering on similarity matrix is the accumulation of data along the diagonal, and noise across the diagonal. CRA_CMH works by completely dropping the "noise" resulting in dropping 50% of the similarity matrix. Now a CMH runs only on the remaining similarity matrix. This is done recursively, and proceeds till the resulting matrix is reduced to a single point, or the stopping condition occurs. The default stopping condition is when the density of the data part of the similarity becomes greater than $\delta(G_B)$ or above a user specified threshold

8.1 Time Complexity

To find the median of vertices in BOT, we use the worst case linear time selection algorithm, for BC mean can be found in linear time, so is the minimum value for MS. For each vertex i, the time to find the median will be $O(d_i)$, where d_i is the degree of vertex i. Since $\sum_{i=1}^{n} d_i = e$, hence the time complexity to find the median of all vertices in BOT will be O(e). To ensure that the medians are always integer, the $\left\lceil \frac{n}{2} \right\rceil^{\text{th}}$ value is taken to be the median. Note that for the bi-graph model being considered, median will always be ≤ n, using counting sort to sort the median values takes O(n) time [6]. Thus the time complexity of one iteration of the MH comes to be $O(\max\{e, n\}) = O(e)$. The number of iterations taken to termination is O(1), thus the time complexity is O(e). Note that the number of iterations is dependent on the amount of noise. For δ(NDS) = 0%, only two iterations of MH are performed. The recurrence relation for CRA_CMH is as follows:

$$T(n) = 2T(n/4) + 2\sum_{i=1}^{n} d_i$$

At each recursive call (after dropping noise) we have two problems, each of size ¼ of the original problem, so the time taken is 2T(n/4). At each recursive call two functions are performed, one is to run the MH, which running on n/4 vertices takes atmost e/4 steps time, and the other is the cost of calculating the density of the partition that also takes atmost e/4 steps. Solving this recursion using the Master's theorem we get $T(n) = 4\sum_{i=1}^{n} d_i$ or 4e or O(e). Note that using median approximators the above time improves by a constant, but results in a solution that is more susceptible to noise as compared to the one working with exact medians.

Unlike CAST [5] and MCL [7], CRA_CMH works on the data matrix as well as the similarity matrix, thus can also be used for two-way or biclustering (introduced in the seventies), though biclustering is not the objective of this work. Note that even algorithms with quadratic time complexities are unacceptable for most KDDM (Knowledge Discovery by Data Mining) applications [10], and unlike other solutions briefly discussed in this section, our solution does provide very good sub quadratic performance for sparse graphs [3].

8.2 Space complexity

For performance reasons, to store the graph we use a dual data structure which is a combination of linked lists of size O(e) and two 1D arrays, each of size O(n). Thus the space complexity is also O(e).

10.0 SIMULATION RESULTS

Both CRA_CMH and CAST were coded in VC++ ver. 6 and simulations as well as the runs on real data were performed on a typical PC with 512 MB RAM.

In this experiment we compare the performance of CRA_CMH as compared to plain CMH and CAST using the standard statistical measures of Purity (P) and Entropy (E). Figure-2(a) shows a S_B corresponding to $\mathbf{K_{64,64} \cup K_{64,64} \cup K_{64,64} \cup K_{32,32} \cup K_{16,16} \cup K_{16,16}}$ with α= 30% (colors added for clarity). Figure-2(b) shows S_B randomly permuted. Figure-2(c) shows the result of using CRA_MH with all 6 clusters extracted with near perfect values of P and E. Figure-2(d) is the outcome of using CAST, showing the two smaller clusters destroyed, and two largest clusters treated as a single cluster (shown by red/grey boundary). Results of using plain MH and BC are shown in Figures-2(e, f), respectively with fairly poor results. MH resulted in 28 clusters, while BC resulted in 36 clusters. Note that cluster boundaries are drawn automatically by the cluster extraction routines.

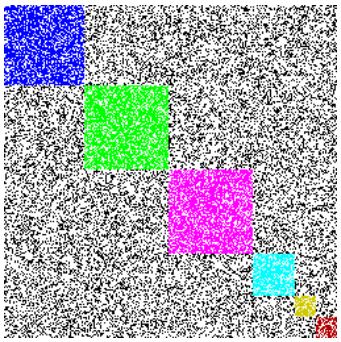

Fig-2(a): Input, α = 30%

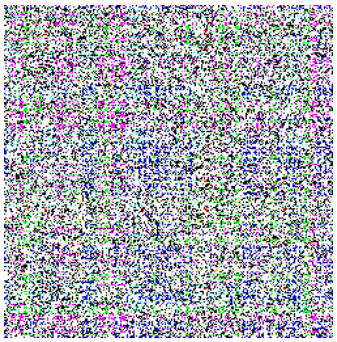

Fig-2(a): Permuted input

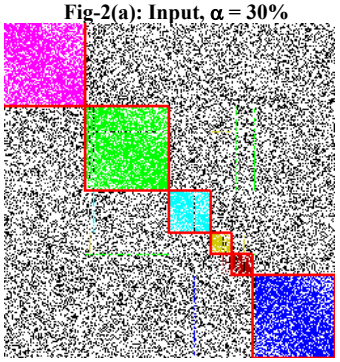

Fig-2(a): CRA_MH: P = 0.98, E = 0.05

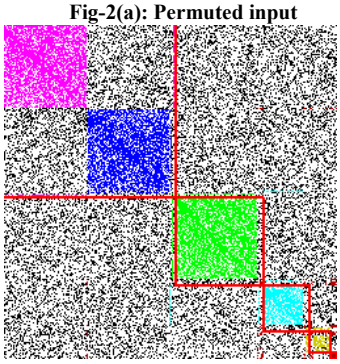

Fig-2(a): CAST: P = 0.68, E = 0.34

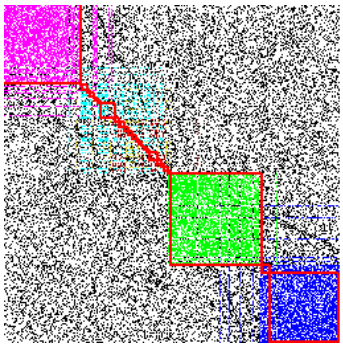

Fig-2(a): MH: P = 0.86, E = 0.21

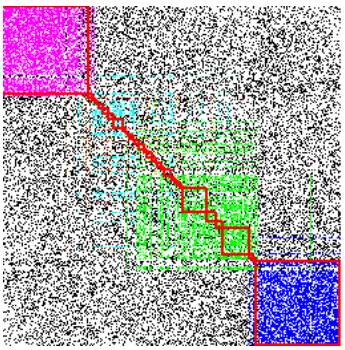

Fig-2(a): BC: P = 0.81, E = 0.29

12.0 CONCLUSIONS

We have shown that when crossing minimization paradigm is used for clustering of pure clusters i.e. 0% noise, perfect results are obtained. We then extended this idea and considered white noise. It was shown that why the crossing minimization using the MH works well for noisy data. Extending this idea to recursive noise removal gives very high noise immunity, and increases the practical utility of the technique.

REFERENCES

1. A. Abdullah , “Discovering pesticide application practices using Data Mining”, to appear in proceedings of the *7th International Conference on Precision Agriculture and Other Precision Resources Management*, Minneapolis, USA, Jul 25-28, 2004

2. A. Abdullah "Heuristics and Meta-heuristics for one-way clustering of gene expression data", to appear in proceedings of the *8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004)* Orlando, USA, July 18-21, 2004

3. A. Abdullah, and S. Brobst, “Clustering by recursive noise removal” with S. Brobst, in proceedings *Atlantic Symposium on Computational Biology and Genome Informatics (CBGI'03) Cary, North Carolina, USA*, pp. 973-977, Sep. 2003.

4. A. Abdullah, "On placement of logic schematics", in Proc. IEEE TENCON'93, Beijing, China, pp. 885-888, Oct 1993.

5. A. Ben-Dor, R. Shamir and Z. Yakhini, “Clustering Gene Expression Patterns", Journal of Computational Biology, vol. 6, no. ¾, pp. 281-297, 1999.

6. T. H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, Pub. The MIT Press, 2000.

7. S. Dongen, “Graph Clustering by Flow Simulation”, PhD thesis, University of Utrecht, May 2000.

8. Donoho, D. L. and P. J. Huber “The notion of a breakdown point,” in Fetschrift for Erich L. Lehmann, edited by P. J. Bickel, K. A. Doksum, J. L. Hodges Jr., Wadsworth: Belmont, CA.

9. P. Eades and N. Wormald. “The median heuristic for drawing 2-layers networks”. Technical Report 69, Department of Computer Science, University of Queensland, 1986.

10. U. Fayyad and R. Uthurusamy, Data Mining and Knowledge Discovery in Databases. *Comm.* ACM, 39(11). Pp. 24-26, Special Issue on Data Mining, 1996

11. Han and Kamber, Data Mining: Concepts and Techniques, Pub by Morgan Kaufmann Publishers, August 2000

12. R. Marti and M. Laguna, “Heuristics and Meta Heuristics for 2-layer straight line crossing minimization”, Discrete Applied Mathematics, 2001

13. Y. Yang, G. I Webb, “A Comparative Study of Discretization Methods for Naive-Bayes Classifiers”, the 2002 Pacific Rim Knowledge Acquisition Workshop, 2002.