World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:4, 2008

MIBiClus: Mutual Information based Biclustering Algorithm

Neelima Gupta, and Seema Aggarwal

***Abstract*—**Most of the biclustering/projected clustering algorithms are based either on the Euclidean distance or correlation coefficient which capture only linear relationships. However, in many applications, like gene expression data and word-document data, non linear relationships may exist between the objects. Mutual Information between two variables provides a more general criterion to investigate dependencies amongst variables. In this paper, we improve upon our previous algorithm that uses mutual information for biclustering in terms of computation time and also the type of clusters identified. The algorithm is able to find biclusters with mixed relationships and is faster than the previous one. To the best of our knowledge, none of the other existing algorithms for biclustering have used mutual information as a similarity measure.

We present the experimental results on synthetic data as well as on the yeast expression data. Biclusters on the yeast data were found to be biologically and statistically significant using GO Tool Box and FuncAssociate.

***Keywords*—**Biclustering, Mutual Information.

I. Introduction

WITH the help of microarray experiments biologists are able to study the expression of thousands of genes under a large number of conditions simultaneously. The large scale of the data makes it challenging to analyse it to extract any biologically significant information from it. The output of a microarray experiment is the gene expression data. The gene expression data has the expression of thousands of genes under thousands of conditions. Standard clustering algorithms like k-means clustering work well for small data sets but fair poorly when the number of experimental conditions is large as they cluster the genes based on their expressions under all the conditions whereas the cellular processes are generally affected by a small subset of conditions. Most of the other conditions which do not contribute to the cellular process add to the background noise. Moreover, these algorithms compute non-overlapping clusters i.e. a gene belongs to at most one cluster whereas in fact a gene may be responsible for several cellular activities and hence must be included in more than one cluster.

Projected clustering is a technique in which data points are projected onto a relevant set of dimensions and a cluster is defined as a set of data points and a set of dimensions that are most relevant to these data points. However, the algorithms for *projected clustering* ([1], [2], [3]) also compute non-overlapping clusters. The clusters overlap on the conditions but not on the genes. In [4], Cheng and Church introduced the notion of biclustering in which the clusters are defined to be a set of genes and a set of conditions under which these genes are most tightly regulated. By definition, biclusters are overlapping.

The existing algorithms ([4], [5], [6], [7], [8], [9], [10], [11]) for biclustering/projected clustering use some kind of similarity measure like Euclidean distance or correlation coefficient. Though these measures have been successfully and satisfactorily used for several years they capture only the linear relationships between the objects. In particular, a vanishing correlation coefficient implies absence of linear dependencies. However, in many applications, like gene expression data and word-document data, non linear relationships may exist between the objects. Moreover, with advances in experimental technology, increasing methodologies are available for unveiling more complex relationships. Hence, we need similarity measures which exploit non linear dependencies.

In [12], Steur et al have shown that mutual information can be used as a measure of similarity to cluster data. They show that mutual information provides a better and more general criterion to investigate relationships (positive, negative correlation and non linear dependencies) between variables by showing that higher correlation coefficient implies higher mutual information but two variables having very low values of correlation coefficient (implying no linear relationship) may still be related to each other (non linear dependencies).

Many researchers ([13], [14], [15], [16]) have used mutual information for one way clustering (clustering of genes on the entire set of conditions). These algorithms also support that information theoretic measure is responsive to any type of dependencies, including strongly non linear structures as compared to traditional measures which search only for linear relations. In [13], Priness et al have compared the mutual information measure with respect to both Euclidean distance and correlation coefficient as a similarity measure for one way clustering . They show that the mutual information is a more generalized measure of statistical dependence and is resistant to outliers and missing data. They also show that mutual information based methods give better quality clusters. With some procedural modifications they incorporated mutual information measure in some clustering algorithms like k-means [17], self organized maps [18], Click [19] and sIB [20]. They found that the clusters obtained from these algorithms using mutual information were similar to each other but different from the clusters obtained when using different distance measures with these algorithms, once again endorsing the need of a different similarity measure. In [14], Butte and Kohane compute pairwise mutual information for all genes against each

Neelima Gupta, Department of Computer Science, University of Delhi, Delhi, India, email: ngupta@cs.du.ac.in Seema Aggarwal, Department of Computer Science, University of Delhi, India (e-mail: saggarwal@mh.du.ac.in).
The project was supported by University of Delhi.

other. They hypothesize that an association between two genes indicated by high amount of mutual information between them would also signify biological relationship. In [21], we have given the first such algorithm that uses mutual information for biclustering. To the best of our knowledge no other work has been done to use mutual information for biclustering. In [23], Dhillon et al and in [24], Slonim et al have used mutual information for co-clustering (simultaneous clustering of rows and columns). In co-clustering, the clusters are obtained by partitioning the rows and the columns, hence the resulting bi-clusters are non-overlapping. In this paper, we present an algorithm which improves upon our previous algorithm in terms of computation time. Also the type of clusters identified can have mixed relationships in contrast to our previous work where the clusters identified had only single relationship.

We define a bicluster as a pair (G', C'), where G' is the subset of genes which are most closely related to each other under the subset C' of conditions and C' is the subset of conditions under which the genes of G' are more closely related to each other as compared to other conditions. Here, we would like to mention that though we have defined the problem in the context of gene expression data, it has its application in other problems like word-document clustering, stock prices monitoring and others.

As in [21] algorithm works in three stages. In the first stage we take a gene as a seed and find the set of genes which are most related to the input seed gene. For this we compute the pairwise mutual information of all the genes with the seed gene over all the conditions and select the genes having mutual information above some threshold. In the second stage, the algorithm identifies the experimental conditions under which the selected subset of genes are most related to each other. In the third and the final stage the algorithm selects those genes which are most co-related with the seed gene under the reduced set of conditions identified in stage two. The main difference between the previous approach and the one presented here is in stage 2 i.e. selecting the relevant conditions. In the previous approach we take a reference condition C^* and compute the set of conditions closely related to C^* for the reduced set of genes; since we do not know which C^* is best we do it for all the conditions one by one. However, to save time, we select a random set of conditions to be treated as C^* instead of doing it for all of them. In the current approach, instead of finding the relevant conditions by finding mutual information between them, we find the subset of conditions on which the mutual information between the selected genes is maximum. Thus the time spent in computing pairwise mutual information between all pairs of conditions is saved which is a significant improvement in time. Moreover, since now we are not looking for relationship amongst the conditions, we are able to exploit mixed kind of relationships amongst the genes.

We tested the performance of our algorithm on computer generated synthetic data and S. cerevisiae expression data [25]. The main idea behind the synthetic data was to model non-linear relationships between genes of the bicluster over a subset of conditions. The biclusters have different types of relationships amongst the genes. We created the synthetic expression data for two overlapping biclusters for 100 genes and 100 conditions. We tested our algorithm on yeast expression data. Gene expression data for Saccharomyces cerevisiae was downloaded from the site http://www.weizmann.ac.il. The dataset contains expression profiles of 6206 genes under 1011 conditions. Our algorithm was able to successfully extract the related groups of genes and the related conditions. We checked the biological significance of our biclusters by finding the functionality on the Gene Ontology database [25]. Our biclusters were found to be significantly enriched with GO categories and had small p values ranging from e^{-11} to e^{-120}. We also used the web tool FuncAssociate [26] to evaluate the discovered biclusters. More than 90% of our biclusters were found to be statistically significant with adjusted p values < 0.001.

II. The Mutual Information

The mutual information between two random variables X and Y is a measure of information contained in X about Y or the information contained in Y about X. If given a value of X, it is easy to predict the value of Y then X contains good amount of information about Y. Clearly with this definition, if X and Y are independent the mutual information between them is zero and it is high if they are highly dependent or closely related to each other. Thus Kullback has defined mutual information between two random variables as a measure of divergence from the hypothesis that X and Y are independent.

A. The Kullback Divergence

Consider a system A with N_A possible states. An experiment performed on A puts the system in one of the states $a_1, a_2 \dots a_{N_A}$, each with its corresponding probability $p(a_i)$. The information gained by the system through a series of experiments is the amount of surprise one feels on reading the outcomes of the experiments. Thus if one hypothesize that probability distribution observed by the outcomes is $\{p^0\}$ and the actual densities are $\{p\}$, the *Kullback* divergence $K(p/p^0)$ between the two probability distributions is given by

$$K(p/p^0) = \sum_i p_i \log \frac{p_i}{p_i^0}$$

Kullback divergence can be interpreted as the information gained when the assumed probability distribution $\{p^0\}$ is replaced by the final distribution $\{p\}$. $K(p/p^0)$ is always greater than or equal to zero [27]. It equals zero if and only if $\{p^0\}$ and $\{p\}$ are same. In our case the assumed probability distribution $\{p^0\}$ is given by the hypothesis that two variables X and Y are statistically independent. Thus $p^0_{XY}(x_i, y_j)$ is given by

$$p^0_{XY}(x_i, y_j) = p_X(x_i) p_Y(y_j)$$

The final distribution $\{p\}$ is given by the observed joint probability densities $p_{XY}(x_i, y_j)$. Thus using Kullback divergence mutual information is defined as

$$I(X, Y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} p_{XY}(x_i, y_j) \log \frac{p_{XY}(x_i, y_j)}{p_X(x_i) p_Y(y_j)}$$

where X takes values $x_1, x_2 \dots x_{n_x}$, Y takes values $y_1, y_2 \dots y_{n_y}$, $p_{XY}(x_i, y_j)$ represents the joint probability

distribution of X, Y and, $p_X(x_i)$ and $p_Y(y_j)$ are the marginal distributions of X and Y respectively. The mutual information is zero if and only if X and Y are statistically independent i.e. vanishing mutual information does imply that the two variables are independent. This shows that mutual information provides a more general measure of dependencies in contrast to the commonly used measures of Euclidean distance and correlation coefficient which quantify only the linear relationships.

B. Estimating Mutual Information

Given the joint probability distribution and the marginal probability distributions, we can compute the mutual information between two variables. This requires explicit knowledge of the distributions. In general these probabilities are not known. Various methods are used to estimate the probability densities from the observed data. We have used Gaussian kernel as the KDE (Kernel Density Estimator) (as used by Steur et al in [12]) to estimate the mutual information. Consider a series (x_i, y_i) of N simultaneous observations of two continuous random variables X and Y. With Gaussian Kernel the probability density estimate is given as:

$$\hat{f}(x) = \frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} exp(\frac{-(x-x_i)^2}{2h^2}) \quad (1)$$

and the joint probability density function may be estimated as

$$\hat{f}(x,y) = \frac{1}{Nh^2 2\pi} \sum_{i=1}^{N} exp(\frac{-d_i(x,y)^2}{2h^2}) \quad (2)$$

$$\text{where } d_i(x,y) = \sqrt{(x-x_i)^2 + (y-y_i)^2} \text{ and,} \quad (3)$$

h is called the window width or the *smoothing* parameter. An appropriate value of h depends on the unknown density being estimated. The estimated mutual information can then be written as

$$I(X,Y) = \frac{1}{N} \sum_{j=1}^{N} \log \frac{\hat{f}(x_j,y_j)}{\hat{f}(x_j)\hat{f}(y_j)} \quad (4)$$

III. MIBiClus: Mutual Information based Biclustering Algorithm

In this section, we present an algorithm to find *biclusters* using mutual information for an expression matrix E_{mat} having N_g genes and N_c conditions. The algorithm takes a gene as a seed (g^*). It finds biclusters which are pairs of (G', C'), where G' is the subset of genes which are most closely related to the gene seed under the subset C' of conditions and C' is the subset of conditions under which the genes of G' are more closely related to each other as compared to other conditions.

The algorithm proceeds in three steps. In the first step we find the set of genes which are most closely related to the input seed gene (g^*). For this we compute the pairwise mutual information of the seed gene with all other genes over all the conditions. For a gene g_i, we define the gene score s_i^g as the amount of mutual information betwen g^* and g_i. Genes having the gene score greater than the gene threshold t_g are selected.

In the second step, the algorithm identifies the experimental conditions under which the set of genes found in the first step show maximum dependence. For a condition c_j, we define the condition score s_j^c as the average contribution of c_j to the sum of pair wise mutual information between the reduced set of genes. Again only those conditions are selected whose score is greater than the condition threshold t_c.

In the third and the final step the algorithm selects from the whole expression data those genes which are most dependent on the gene seed under the reduced set of conditions identified in step two.

The procedure MIBiClus() summarizes our algorithm.

MIBiClus $(g^*, t_g, t_c, N_g, N_c)$
1) G_0 = Compute-G0(g^*, t_g, N_g, N_c)
2) C' = Compute-conditions(G_0, t_c, N_c).
3) G' = Compute-biclus-genes(C', t_g, N_g)

Fig. 1. Main Algorithm.

Compute-G0 (g^*, t_g, N_g, N_c)
1) Let $g^* = g_t$ for some t.
2) For $i = 1$ to N_g
   a) For $j = 1$ to N_c
      $m(i,j)$ = Compute-mi (i, t, N_c, j).
   b) compute the gene score $s_i^g = \Sigma_j m(i,j)$
3) $\mu = \frac{\Sigma_i s_i^g}{N_g}$
4) $\sigma^2 = \frac{\Sigma_i (s_i^g - \mu)^2}{N_g}$
5) $G_0 = \{g_i : \frac{(s_i^g - \mu)}{\sigma} > t_g\}$.

Fig. 2. Step 1: Compute the initial gene set.

Compute-conditions (G_0, t_c, N_c).
1) Compute the condition score $s_j^c = \Sigma_{i \in G_0} m(i,j)$
2) $\mu = \frac{\Sigma_j s_j^c}{N_c}$
3) $\sigma^2 = \frac{\Sigma_j (s_j^c - \mu)^2}{N_c}$
4) $C' = \{c_j : \frac{(s_j^c - \mu)}{\sigma} > t_c\}$.

Fig. 3. Step 2: Compute the relevant set of conditions.

Compute-biclus-genes (C', t_g, N_g)

1) For i = 1 to N_g
   compute the gene score $s_i^g = \Sigma_{j \in C'} m(i,j)$
2) $\mu = \frac{\Sigma_i (s_i^g)}{N_g}$
3) $\sigma^2 = \frac{\Sigma_i (s_i^g - \mu)^2}{N_g}$
4) $G' = \{g_i : \frac{(s_i^g - \mu)}{\sigma} > t_g\}$.

Fig. 4. Step 3: Compute the final genes of the bicluster.

Compute-mi (x, y, n, j)

1) $\hat{f}_{xj}$ = Compute-marg (x, n, j)
2) $\hat{f}_{yj}$ = Compute-marg (y, n, j)
3) $\hat{f}_{xyj}$ = Compute-joint (x, y, n, j)
4) Return $\frac{1}{n} \log(\frac{\hat{f}_{xyj}}{\hat{f}_{xj} \hat{f}_{yj}})$

Fig. 5. Compute the j^{th} component of the mutual information between genes x and y.

Compute-marg (x, n, j)

1) Compute d_{ji} = $(E_{mat}[x][j] - E_{mat}[x][i])^2$
2) Return $\frac{1}{nh\sqrt{(2\pi)}} \Sigma_i \exp^{\frac{-d_{ji}}{2h^2}}$

Fig. 6. Compute the j^{th} component of the estimator $\hat{f}_x$ for marginal probability density.

Compute-joint (x, y, n, j)

1) Compute d_{ij}^x = $(E_{mat}[x][j] - E_{mat}[x][i])^2$
2) Compute d_{ij}^y = $(E_{mat}[y][j] - E_{mat}[y][i])^2$
3) Compute d_{ij} = d_{ij}^x + d_{ij}^y
4) Return = $\frac{1}{nh^2 2\pi} \Sigma_i \exp^{\frac{-d_{ij}}{2h^2}}$

Fig. 7. Compute the j^{th} component of the estimator $\hat{f}_{xy}$ for joint probability density.

IV. Experimental Results

In order to study the performance of our algorithm we used computer generated synthetic data. The main idea behind the synthetic data was to model nonlinear relationships between genes of the bicluster over a subset of conditions.

We created synthetic expression data for 100 genes and 100 conditions with two overlapping biclusters (refer to Figure 8). The first bicluster $M1$ consisted of genes g_1 to g_{55} and conditions c_1 to c_{55}. The first 15 genes (g_1 to g_{15}) of $M1$ had additive relation with the gene seed chosen as (g_{46}), the next 15 (g_{16} to g_{30}) of $M1$ had circular relation with the gene seed, the next 15 (g_{31} to g_{45}) had parabolic relation with the gene seed under the first 55 conditions.

Genes g_{47} to g_{50} had additive relation and genes g_{51} to g_{55} had parabolic relation with the gene seed g_{46} under all the condition c_1 to c_{100}.

$M2$ consisted of genes g_{45} to g_{100} and conditions c_{45} to c_{100}. $M2$ had additive relation on the 15 genes (g_{56} to g_{70}), circular relation in the next 15 genes (g_{71} to g_{85}) and parabolic relation in the last 15 genes (g_{86} to g_{100}) with the gene seed g_{46} under the last 55 conditions from c_{45} to c_{100}. The rest of the rows and columns ($D1$ and $D2$ in the figure) were given high constant value 10 to make them independent of the rows and columns in the biclusters.

We were able to identify $M1$ at $t_g = -0.5$, $t_c = -0.4$, $h = 0.2$ and choosing any of the genes from g_1 to g_{45} as the seed. By choosing the seed from g_{56} to g_{100} we were able to identify $M2$.

The distance based algorithms like PROCLUS and MSB are not able to find $M1$ and $M2$ as they clearly converge to $D1$ and $D2$. ISA also fails in detecting $M1$ and $M2$ because of high expression values in $D1$ and $D2$. Our previous algorithm also does not detect these biclusters because of the presence of the mixed relationships. However it identifies the six biclusters (with circular relationships only, with additive relationships only and parabolic relationships only) contained in $M1$ and $M2$.

We tested our algorithm on real dataset also. Gene expression data for Saccharomyces cerevisiae was downloaded from the site http://www.weizmann.ac.il. The dataset contained expression profiles of 6206 genes under 1011 conditions. We chose 200 gene seeds randomly. The algorithm was able to successfully extract the related groups of genes and a subset of conditions for each of these groups. Frequency of each bicluster was computed by comparing it against all other biclusters for overlap. A bicluster showing less than 40% overlap with all other biclusters was considered infrequent and was filtered out. We checked the biological significance of the remaining biclusters by finding their functionality on the Gene Ontology database [25]. Our biclusters were found to be significantly enriched with GO categories and had small p values. At t_g = 2.0, t_c = 0.1 and $h = 0.5$ we found biclusters having p values in the range of e^{-11} to e^{-120}. Three biclusters out of 200 showing high frequency of occurrence (> 15) had p values of the order of e^{-65} to e^{-120}. Also, these biclusters had significant overlap with that of ISA. We also used the web tool FuncAssociate [26] to evaluate the biclusters. More than 90% of our biclusters were found to be statistically significant with adjusted p values < 0.001.

V. Conclusion

We have presented an algorithm which improves upon our previous algorithm for computing biclusters using mutual information. As the mutual information captures more general relationships as compared to traditional similarity measures

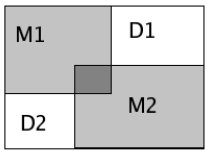

Fig. 8. Expression matrix for synthetic data showing two overlapping biclusters.

like Euclidean distance and correlation coefficient, our algorithm will be able to discover more and better biclusters in more complex data sets.

We applied our algorithm to the Yeast expression data and the results are promising. In future, we intend to apply it on more complex data sets like expression matrix of higher organisms.

ACKNOWLEDGMENT

We thank Jan Ihmels and Naama Barkai, Weizmann Institute of Science, Israel for allowing us to download the Yeast expression data from their website.

REFERENCES

[1] Kevin Y. Yip, David W. Cheung, and Micheal K. Ng:HARP: A Practical Projected Clustering Algorithm. IEEE Transactions On Knowledge And Data Engineering, volume 16, IEEE Computer Society, 2004.
[2] Charu C. Aggarwal, Cecilia Procopiuc, Joel L. Wolf, Philip S. Yu, and Jong Soo Park: Fast Algorithms For Projected Clustering. ACM SIGMOD Int'l Conf. Management of Data, pages 61–72, (1999).
[3] Charu C. Aggarwal and Philip S. Yu:Finding generalized projected clusters in high dimensional spaces. ACM SIGMOD Int'l Conf. Management of Data, pages 70–81, 2000.
[4] Y. Cheng and G. M. Church: Biclustering Of Gene Expression Data. System Molecular Biology, volume 8, pages 1–93. System Molecular Biology, 2000.
[5] H. Wang, W. Wang, J. Yang, and P. S. Yu:Clustering by Pattern Similarity in Large Data Sets. Bull, Math. Biol. 46, pages 515 –527. ACM Press, 1984.
[6] Gad Getz, Erel Levine, and Eytan Domany: Coupled Two-Way Clustering Analysis Of Gene Microarray Data. Cell Biology, volume 97, pages 12079–12084. PNAS, 2000.
[7] Yuval Kluger, Rones Basri, Joseph T. Chang, and Mark Gerstein: Spectral Biclustering Of Microarray Data: Coclustering Genes And Conditions. Genome Research, Cold Spring Harbor Laboratory Press, 2003.
[8] J. Ihmels, G. Friedlander, S. Bergmann, Y. Ziv O. Sarig, and N. Barkai: Revealing Modular Organization In The Yeast Transcription Network. Nature Genetics, volume 31, pages 1–370. Nature Publishing Group, 2002.
[9] S. Bergmann, J. Ihmels, and N. Barkai:Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review, volume 67, pages 1–18. American Physical Society, 2003.
[10] M. Kloster, C. Tang, and N.S. Wingreen:Finding regulatory modules through large-scale gene-expression data analysis. Bioinformatics, volume 21, pages 1172–1179, 2005.
[11] Xiaowen Liu and Lusheng Wang: Computing the maximum similarity bi-clusters of gene expression data. bioinformatics, volume 23, pages 50–56, 2007.
[12] R.Steuer and J.Kurths and C.O.Daub and J.Weiseand and J.Selbig:The mutual information:Detecting and evaluating depencies between variables. Bioinformatics,volume 18 Suppl 2, pages S231–S240, 2002.
[13] O.Maimon I. Priness and I.Ben-Gal: Evaluation of gene expression clustering via mutual information distance measure. BMC Bioinformatics, 2007.
[14] A.J. Butte and I.S. Kohane: Mutual Information relevance networks, Functional Genomic clustering using pairwise entropy measurements. PSB,volume 5, pages 415–426, 2000.
[15] X.Zhou, X. Wang, E.R. Dougherty, D.Russ and E. Suh: Gene Clustering Based on Clusterwide Mutual Information. Jouranl of Computational Biology, volume 11, Number 1, pages 147–161, 2004.
[16] N.SLonim, G.S.Atwal,G.Tkacik and W.Bialek: Information based clustering. PNAS, volume 102, pages 18297–18302, 2005.
[17] MacQueen: Some methods for classification and analysis of multivariate observations. Proc.of the 5th Berkeley Symposium om Mathematical Statistics and Probability, 1965.
[18] T. Kohonen: Self Organizing Maps, Springer, 1997.
[19] R. Sharan and R. Shamir: CLICK: a clustering algorithm with applications to gene expression analysis Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 2000.
[20] N. Slonim: The information Bottleneck: Theory and applications. In Ph.D. Thesis Tel Aviv University, Computer Science Department, 2002.
[21] Neelima Gupta and Seema Aggarwal: MIB: Using Mutual Information for Biclustering .
[22] G.S.Michaels,D.B.Carr,M.Askenazi,S.Fuhrman,X.Wen and R.Somogyi: Cluster Analysis and Data Visualization of Large Scale Gene Expression Data. PSB, volume 3, pages 42–53, 1998.
[23] S.Mallela I.S.Dhillon and D.S.Modha: Information Theoritic CO-Clustering. Proceedings of SIGKDD'03, 2003.
[24] N.Slonim and N.Tishby: Document clustering using word clusters via the information bottleneck method. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, p.208-215, July 24-28, 2000, Athens, Greece
[25] Saccharomyces Genome Database, http://www.yeastgenome.org/.
[26] Funcassociate: The Gene Set Functionator, http://llama.med.harvard.edu/cgi/func1/funcassociate.
[27] Simon Haykin.: Neural Networks-A comprehensive Foundation 2nd Ed., Prentice Hall of India, 2007.