

Cluster Subspace Identification via Conditional Entropy Calculations

James C. Diggans¹ and Jeffrey L. Solka²

1 - George Mason University, 2 - Dahlgren Division of the Naval Surface Warfare Center

Abstract

Methods of high-level data exploration capable of robustness in the face of noise found within microarray data are few and far between. Solutions making use of all original features to derive cluster structure can be misleading while those that rely on a trivial feature selection can miss important characteristics. We present a method adopted from previous work in the field of geography (Guo et al, 2003) relying upon conditional entropy between pairs of dimensions to uncover underlying, native cluster structure within a dataset. Applied to an artificially clustered data set, this method performed well though some sensitivity to multiplicative noise was in evidence. When applied to gene expression data, the method produced a clear representation of the underlying data structure.

Introduction

Our standard microarray dataset consists of n_s observations, samples, or experiments in n_g genes or dimensions. So our data matrix is n_s rows by n_g columns. n_g is typically much larger than n_s . Microarray data sets are, by their nature, high-dimensional data sets complicating the analysis of results and restricting one's ability to perform initial, high-level data exploration (i.e. it's rather difficult to visually inspect 8,000 dimensions). Even if this were tenable, useful clusters that exist across all available dimensions are exceedingly rare. There is no shortage of methods available for feature selection in an attempt to prune down the number of genes to a 'best' set for visualizing the cluster structure within the data. This, however, presupposes that there is a single 'best' collection of genes useful for visualizing all subspace clusters which is not often the case (Getz *et al*, 2000). A more useful means for detecting subspace cluster structure would make use of only those genes useful in assessing the cluster structure for a particular sub-set of dimensions. It is this attribute which makes conditional entropy such a useful tool for high-level data analysis.

If entropy is the amount of information provided by the outcome of a random variable, then conditional entropy can be defined as the amount of information about the outcome of one random variable provided by the outcome of a second random variable. We can make use of this measure of shared information in exploring a series of dimensions and observations on those dimensions (Cheng, C. *et al*, 1999). For any given data set about which, perhaps, we know very little, we can make use of conditional entropy to discover clusters of dimensions. This information can then be used to inform and to tailor downstream analyses to the structure inherent in the data set to be analyzed.

Guo et al. 21003 use this technique to discover latent, unexpected clusters on dimensional subspaces. Their model data sets from the field consist of a limited

number of dimensions (e.g. several different types of geological measurements) and a very large number of observations on those dimensions (e.g. in excess of ten thousand). Guo *et al* use this technique to discover latent, unexpected cluster structure between their measurement dimensions.

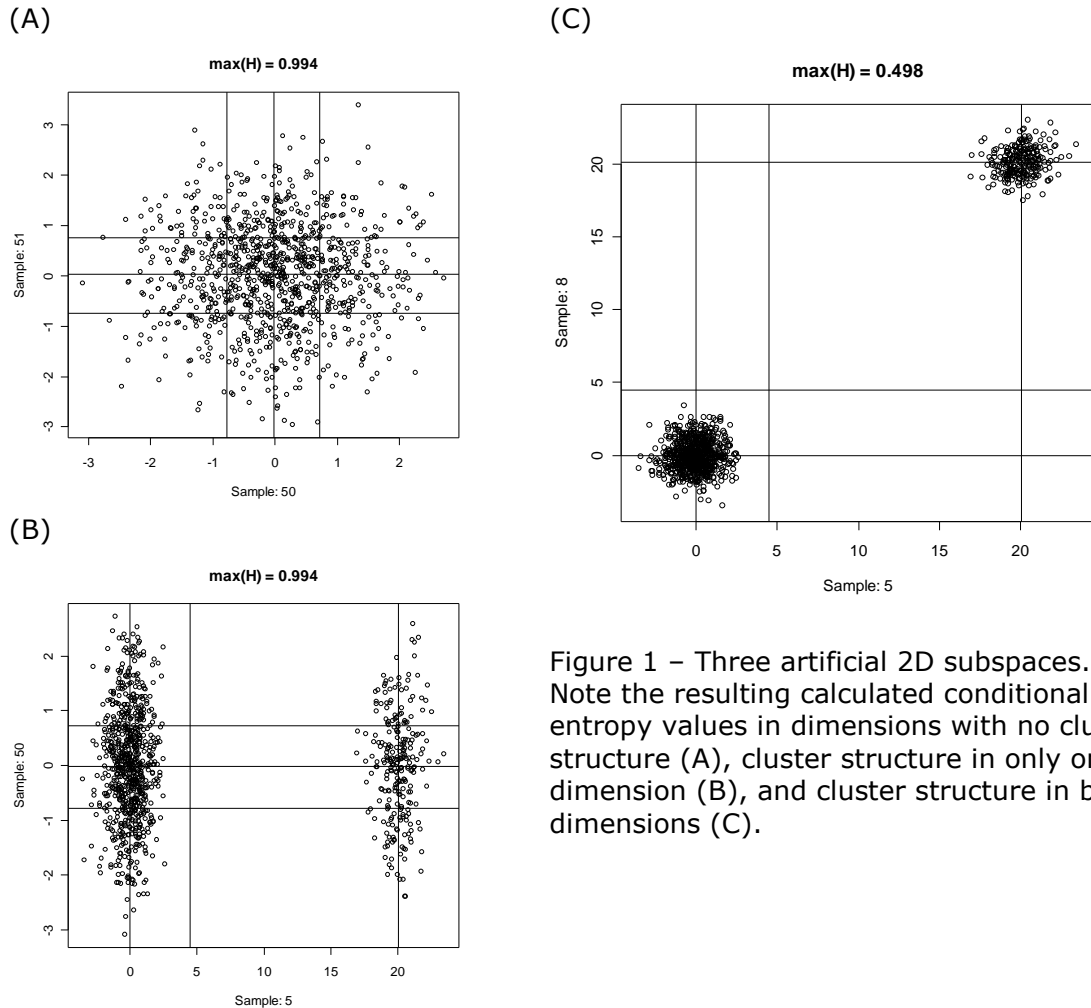


Figure 1 – Three artificial 2D subspaces. Note the resulting calculated conditional entropy values in dimensions with no cluster structure (A), cluster structure in only one dimension (B), and cluster structure in both dimensions (C).

When applying this technique to exploration of gene expression data we face an altogether different challenge. Viewing a gene expression data set in the same light as the data found in the Guo *et al* work, we would consider the genes as our 'dimensions' (i.e. our measurements) and the samples run over the microarray to be 'observations' on those dimensions. Determining cluster structure using conditional entropy when there are far more dimensions than observations is simply untenable: this technique is not currently useful in detecting cluster structure within genes present on a microarray. If, however, we flip what we consider to be 'dimensions' – now our samples – and what we consider to be 'observations' – now our genes, we can apply this data exploration technique to microarray data sets. So our data matrix consists of n_g , number of genes or rows, by n_s , number of samples or observations.

System and Methods

Our high-level goal in this exploratory analysis is to determine, given any pair of 2D subspaces, which 2D subspace can be considered more 'clustered' than the other. Guo *et al* suggest several requirements that should be met by any generally useful measure of cluster tendency:

- 'It does not assume any particular cluster shape;
- It does not assume any particular cluster size (in terms of both the coverage and the range on each bound dimension);
- It does not assume any distribution of data points;
- It is robust with various dataset sizes;
- It is robust with outliers and extreme values;
- It is tolerant to a high portion of noise.'

Given these requirements, and before we can calculate the conditional entropy between any two pairs of dimensions, we must first discretize each 2D subspace so that we may measure the relative difference in localized density (as a measure of cluster tendency) for each 2D subspace to be analyzed. This can be accomplished, trivially, by breaking each dimension up into n equally-sized intervals. This approach, however, is very sensitive to various data distributions, violating one of the requirements on our measure of cluster tendency. Guo *et al* propose instead a more flexible discretization method called 'nested means' discretization (NM).

When applying NM to a data set, we can imagine a scatterplot (see Figure 2) for each pair of dimensions and all observations on those dimensions in the plot. For each dimension, the mean of all observations is calculated and a boundary drawn. Means are then calculated for each of the two ranges created by the initial drawn boundary. This proceeds until each dimension has been divided up into the 'right' number of discrete intervals. This notion of the 'right' number of intervals is dependant on the number of observations in the data set and is intended to ensure that, on average, each cell (an area bounded by two pairs – one from each dimension – of overlapping interval boundaries) contains 35 observations (Cheng *et al*, 1999).

Nested Means Discretization

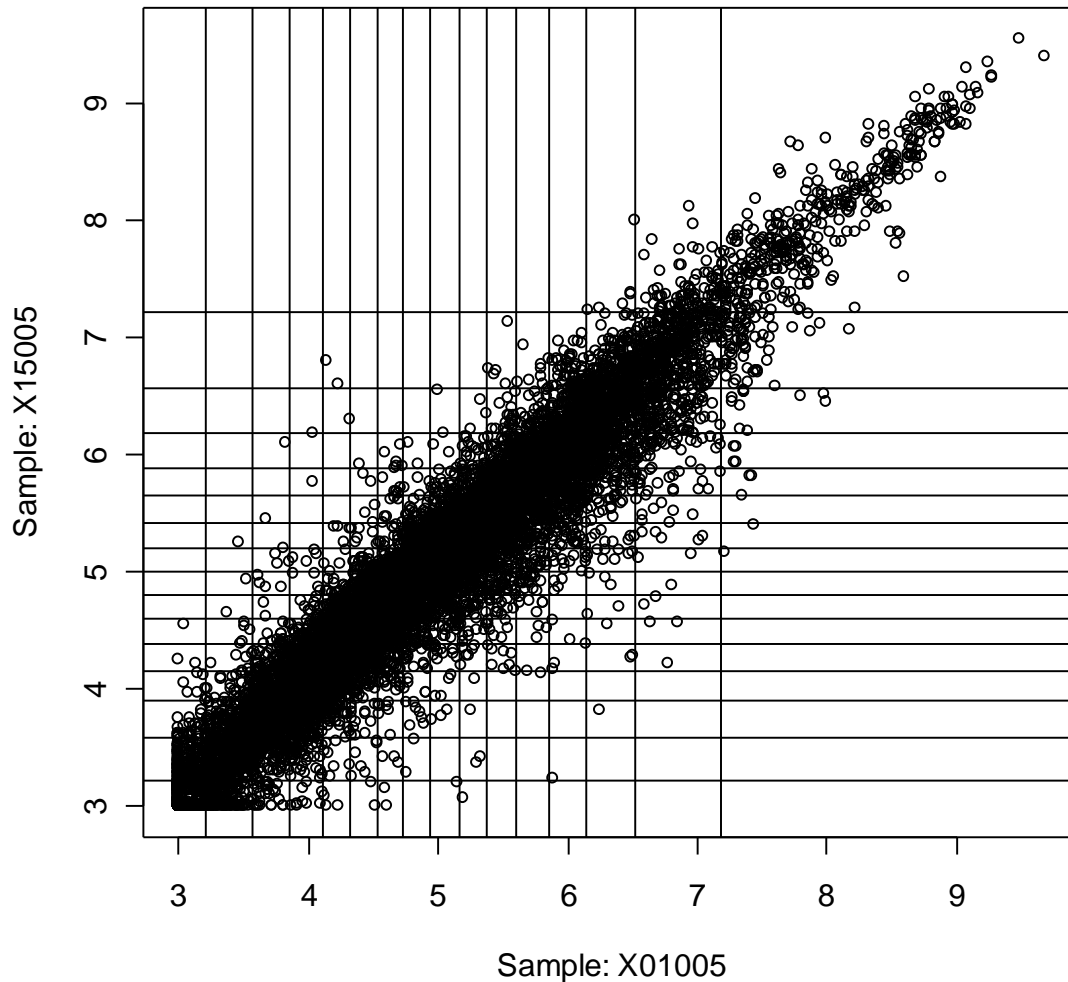


Figure 2 - Nested means discretization for two samples from a microarray data set using logged data. Note the uneven interval size in both tails of the large cluster between the two dimensions. This helps account for outliers in a way that would reduce the utility of an equal-interval discretization.

In addition to the goal of having 35 observations per cell, the ideal number of intervals, r , should also be some value of 2^k where k is a positive integer. This ensures that our discretization is balanced (i.e. that we never have to take only one side of a newly-halved sub-interval). Guo et al. 2003 suggest a rule to determine the proper number of intervals (which we have made use of in our application), namely, that $n/r^2 \approx 35$ along with $r = 2^k$. In the example given in Figure 1, we have 12,625 observations on our two sample dimensions. Using Guo's formula, then, $r = 16$ because $16 * 16 = 256$ and $256 * 35 = 8960$ (which is the closest value obtainable based on r being an even power of two without going over our upper bound of 12,625 genes). Once the subspace has been discretized, we add up the number of

observations found in each cell and use that number to represent the cell in downstream calculations.

Now that we have successfully discretized our 2D sub-space in a way that is robust to various distributions in the underlying data, we are ready to compute the conditional entropy between each pair of dimensions as a measure of cluster tendency within those dimensions. We use a calculation of local entropy in each column or row as a means to a final, weighted conditional entropy. Given a vector of cells, χ , in the matrix, we define $d(x)$ to be the density of a given cell in χ by dividing the number of points found in x by the total number of points across all cells in the vector χ . We can then calculate the entropy of the vector as:

$$H(C) = -\sum_{x \in \chi} [d(x) \log_e d(x)] / \log_e |\chi|$$

where $|\chi|$ denotes the number of entries in the initial vector. The calculation of conditional entropy given a matrix of values is found in (Pyle, 1999):

1. To first calculate $H(Y|X)$, calculate the sum of each column of data in the discretized 2D subspace.
2. Calculate the weight of each column as the column sum divided by the total number of observations, n .
3. Calculate the entropy for each column. In the toy example given in Table 1, taken from Guo *et al*, the entropy of column X2 is calculated as:

$$H(X2) = -[(1/36)*\log(1/36)+(9/36)*\log(9/36)+...+(2/36)*\log(2/36)]/\log(6)$$

which is 0.847; 36 represents the total number of points in column X2 while 6 is the total number of cells in the column (so $r = 6$; note that the toy example does not average 35 observations per cell as it is solely for the purposes of demonstration).

4. Conditional entropy, $H(Y|X)$ is then the weighted sum of all column entropies. In Table 1, this is 0.700.

Calculation of conditional entropy using rows instead of columns in Table 1 results in determination of the conditional entropy $H(X|Y)$ instead of $H(Y|X)$.

	X1	X2	X3	X4	X5	X6		Sum	Wt	H
Y1	0	1	3	0	0	0		4	.03	.314
Y2	1	9	1	0	1	2		14	.09	.629
Y3	7	14	3	7	6	0		37	.25	.835
Y4	7	6	13	19	12	5		62	.41	.939
Y5	0	4	14	5	1	1		25	.17	.668
Y6	1	2	3	2	0	0		8	.05	.737
										H(X Y) 0.812
Sum	16	36	37	33	20	8				
Wt	.11	.24	.25	.22	.13	.05		H(Y X) 0.700		max(H) 0.812
H	.597	.847	.806	.615	.540	.502				

Table 1 - A toy example of the calculation of conditional entropy for both $H(X|Y)$ (across the rows) and $H(Y|X)$ (across the columns).

If both $H(X|Y)$ and $H(Y|X)$ are small, the subspace comprised of the two dimensions shows a great deal of cluster tendency. If one of the values is small and one is large, the subspace clusters well only in one dimension (a situation in which we have little interest). For this reason, we use the larger of the two conditional entropy values to represent the overall conditional entropy of the 2D subspace. Guo points out that, since this measure will be used only to compare cluster tendency between 2D subspaces, we can consider the measure robust to noise as long as the noise is present in all dimensions.

Performing this calculation across all 2D subspaces from our initial collection of dimensions results in what can be considered a distance matrix consisting of d columns and d rows populated by our calculated conditional entropy values. This interpretation facilitates further analysis using graph-theoretic methods. We can visualize the relationship between all dimensions as a complete graph assigning to each edge the conditional entropy value between its two vertex dimensions. This graph can then help us both to visualize the relationships between dimensions and to identify groups of dimensions with a high-degree of cluster tendency. In the latter case, we depend upon the idea that if an L -dimensional subspace (where $2 < L < d$ for d , the total number of dimensions in our data set) has good cluster tendency, all possible 2D subspaces of our original L -dimensional subspace will have low conditional entropy.

Use of a data image to visualize high-level cluster structure in the conditional entropy matrix is a useful approach. It depends, however, upon an 'optimal ordering' of the dimensions in the distance matrix. Guo *et al.* 2003 suggest the construction of a minimum spanning tree (MST) from the complete graph along with a UPGMA-like tree-building approach to come up with this final, optimal ordering of the samples. An MST is a graph with $n-1$ total edges (where n is the number of nodes in our graph) in which edges have been chosen to minimize the total weight of the graph.

We make use of Kruskal's algorithm in generating a minimum spanning tree. From this tree, the optimal dimensional ordering is produced by constructing a UPGMA (unweighted, pair-group method with arithmetic mean; the simplest method of tree construction building a tree in a stepwise manner selecting topological relationships in order of decreasing similarity) tree using the remaining conditional entropy edge weights in the MST.

To identify groups of dimensions with a high degree of cluster tendency, we can use any of a number of clustering algorithms on the CE distance matrix. Applying a graph-theoretic cluster-detection method like clique partitioning to the fully-connected graph is a useful approach. A maximum edge-weight parameter is used to remove those edges too long to be considered of 'use' to the clustering effort. Clique determination can then proceed given an efficient algorithm.

The statistics language R was used in the development of methods to discretize microarray data sets and to calculate conditional entropy distance matrices. Additional R routines were written to calculate MST and fully-connected graphs. Perl and the Graphviz library from AT&T were used to visualize constructed graphs.

Results and Discussion

Analysis and validation of the method were carried out on an artificial data set. We constructed a matrix with 100 'samples' and 1,000 'genes'. We analyze this as a 1000 by 100 data matrix consisting of 1000 observations in 100 dimensional space. The matrix was populated with random data distributed as $N(0,1)$. The first one-fourth of the genes was permuted upwards by three in samples 5-8. The second one-fourth of the genes was permuted downwards by three in samples 10-14. The third one-fourth of genes was permuted upwards by five in samples 24-30. The last one-fourth of genes was permuted downwards by five in samples 56-66. The resulting data set, then, contains four distinct clusters of observations that only manifest themselves across a subset of the available dimensions.

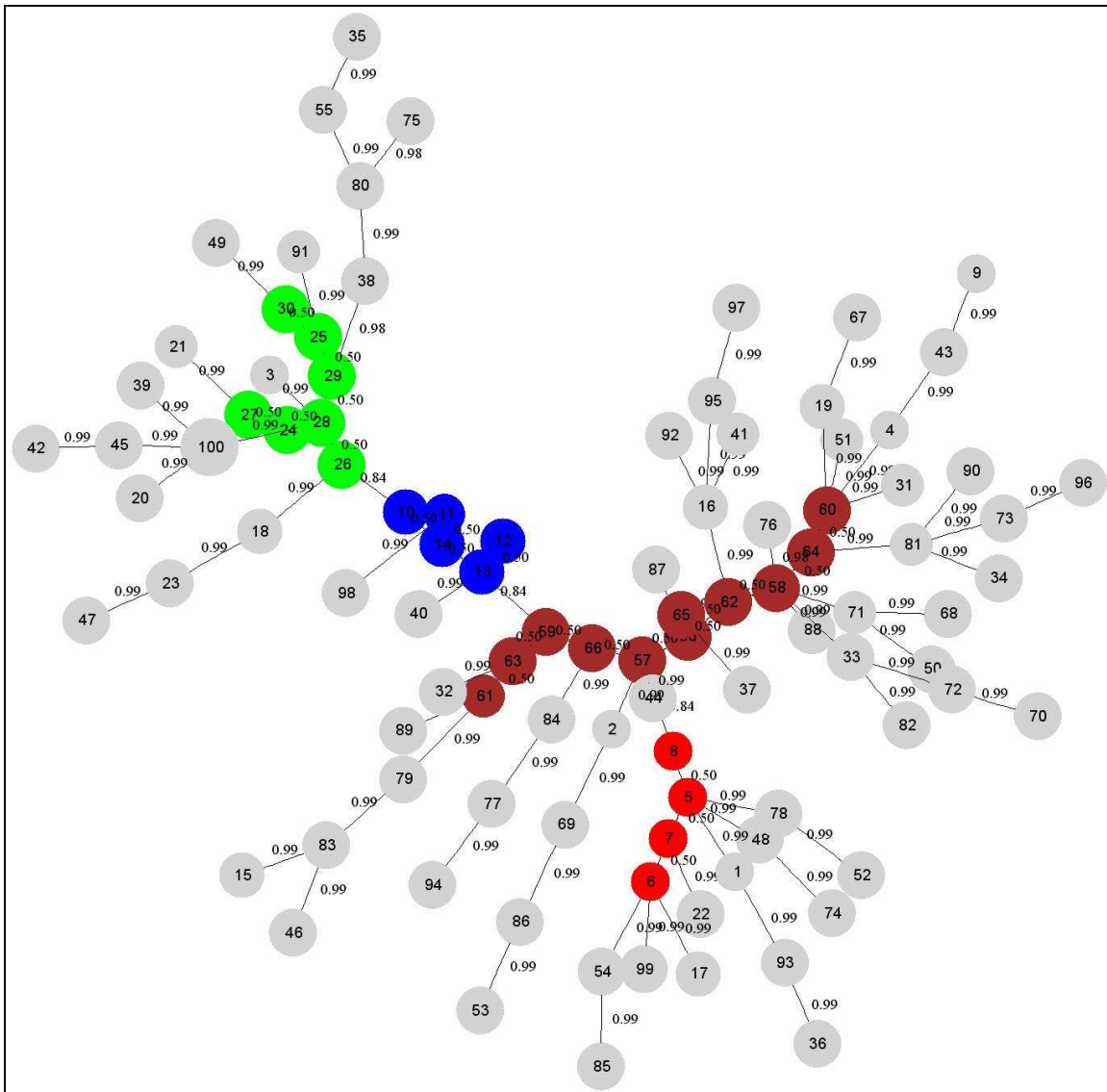


Figure 3 - Minimum spanning tree derived from the conditional entropy distance matrix built from the artificially-clustered data set. The member nodes in each of the four clusters are color-coded; samples (dimensions) not in any cluster are gray. Note that all gray/gray and gray/color edge weights have CE ≈ 0.99 .

As shown in Figure 3, the four clusters are clearly evident in the MST. The edge weights between members of each cluster are significantly lower than those edges between clusters. If the samples are ordered by building a UPGMA tree based on the edges found in the MST in Figure 3, the data image shown in Figure 4 results. The four clusters are clearly visible.

Artificial Data: MST-Clustered Conditional Entropy

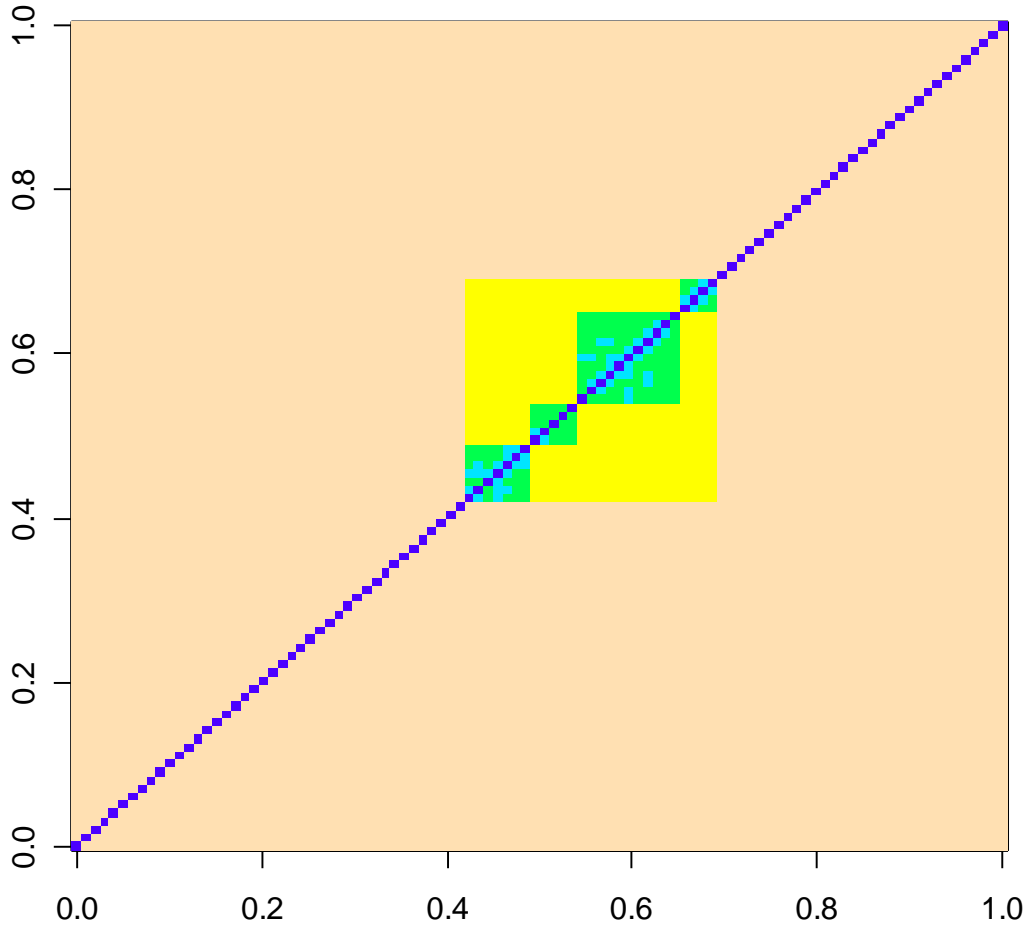


Figure 4 - Data image of the conditional entropy matrix derived from the artificial data set. The samples are sorted based on the hierarchical tree built from the MST edges. Note the four distinct clusters in the graph.

To examine the effects of noise on this technique, a second matrix of random values distributed as $N(0,1)$ of the same size was generated. Multiplying our clustered matrix by this random matrix generated a new test matrix in which our cluster structure has been obfuscated by noise.

Perturbed Data: MST-Clustered Conditional Entropy

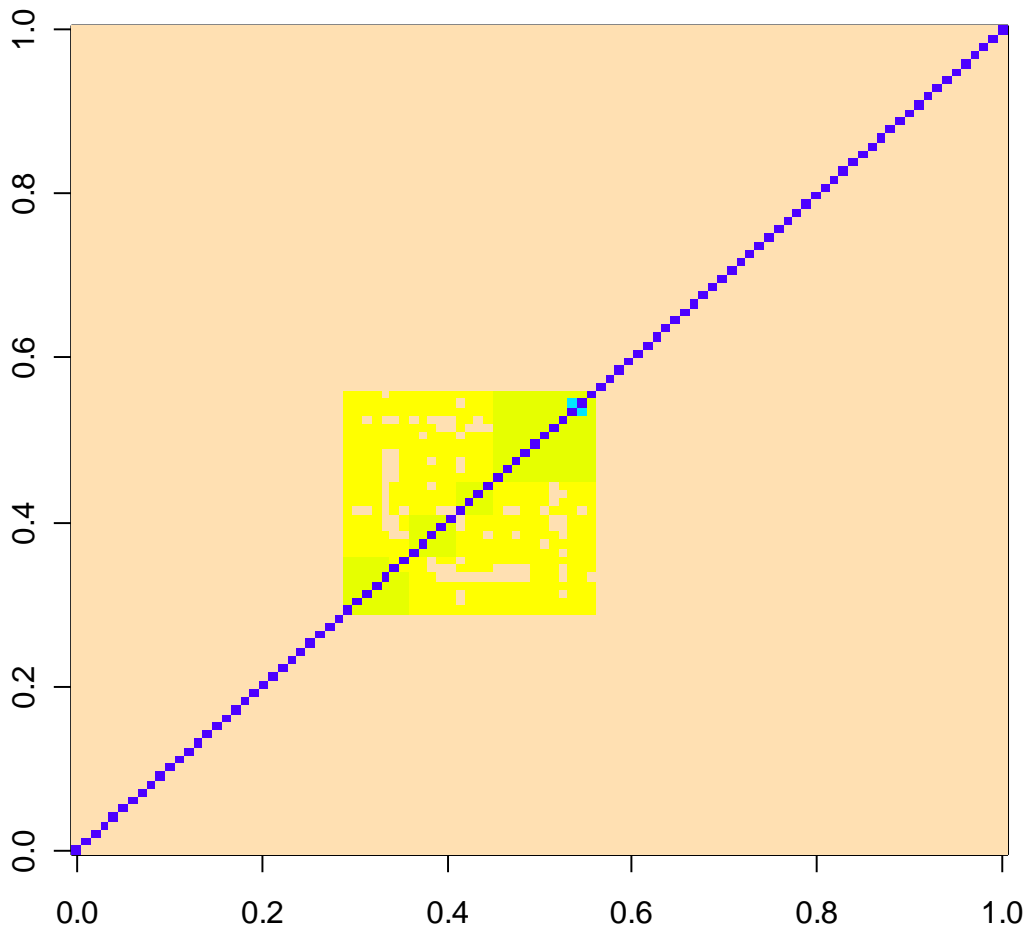


Figure 6 - Data image of the perturbed conditional entropy matrix derived from the artificial data set multiplied by an equivalent matrix of random $N(0,1)$ values. The samples are sorted based on the hierarchical tree built from the MST edges. Note the four distinct clusters in the graph.

Turning our attention to the performance of the technique when applied to real microarray data, we selected two public microarray data sets. The first is a data set from Golub *et al* designed to determine whether or not microarray data could distinguish between two related diseases, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML), by analysis of gene expression profiles in immune cells. The data set contains 7,129 genes and 72 samples. 47 of the samples are from patients with ALL while 25 are from patients with AML.

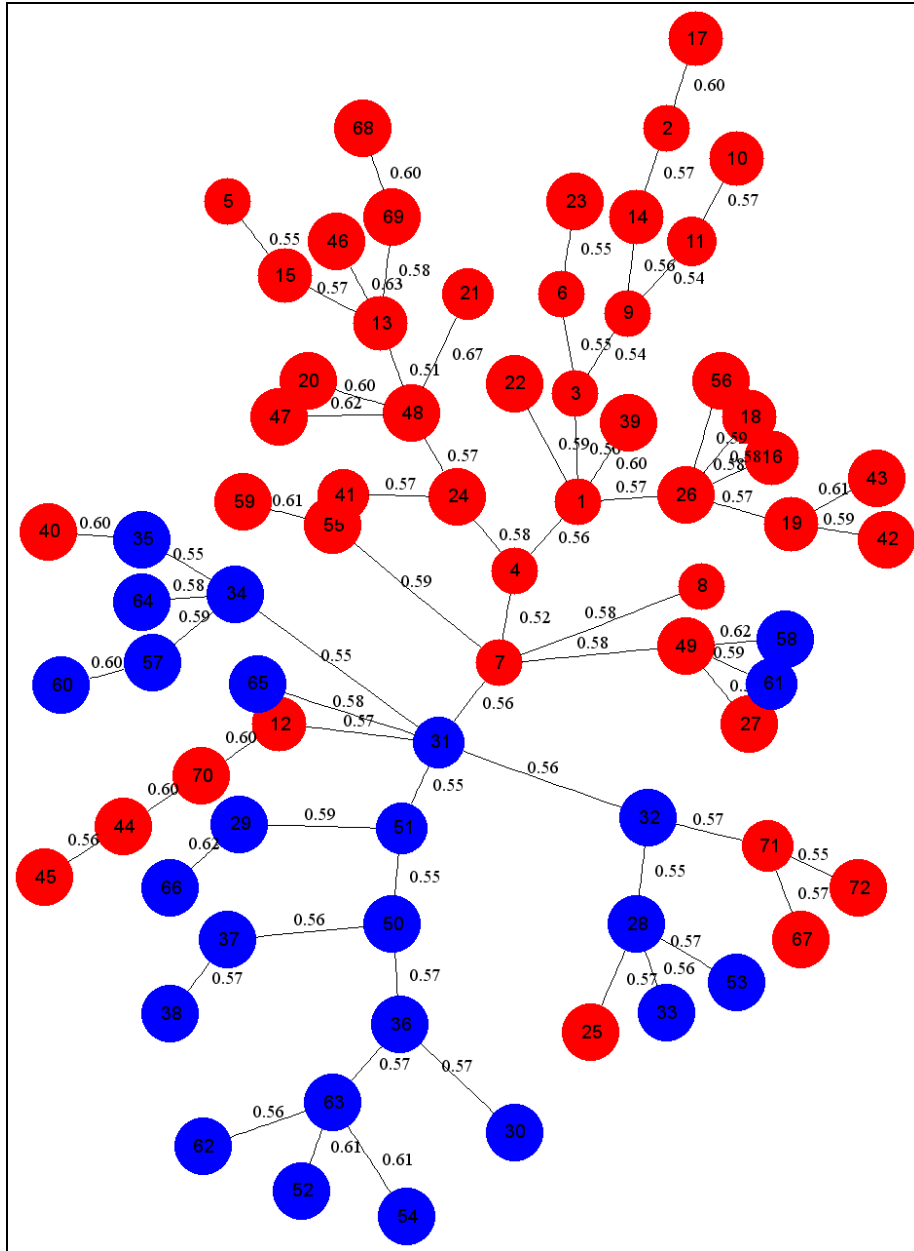


Figure 7 - Minimum spanning tree resulting from the conditional entropy distance matrix calculated on the Golub et al. 1999 data set. ALL samples are labeled in red; AML samples are labeled in blue.

Topologically, the resulting MST (see Figure 7) does show a high degree of separation between the AML and ALL samples. The edge weights within each sample type are, in general, shorter than those between sample types but exceptions exist, negating some potential benefit of automatic cluster identification. The sorted data image (see Figure 8) does not present any obvious cluster structure although some indication of structure can be said to exist in the image.

Golub: MST-sorted Conditional Entropy

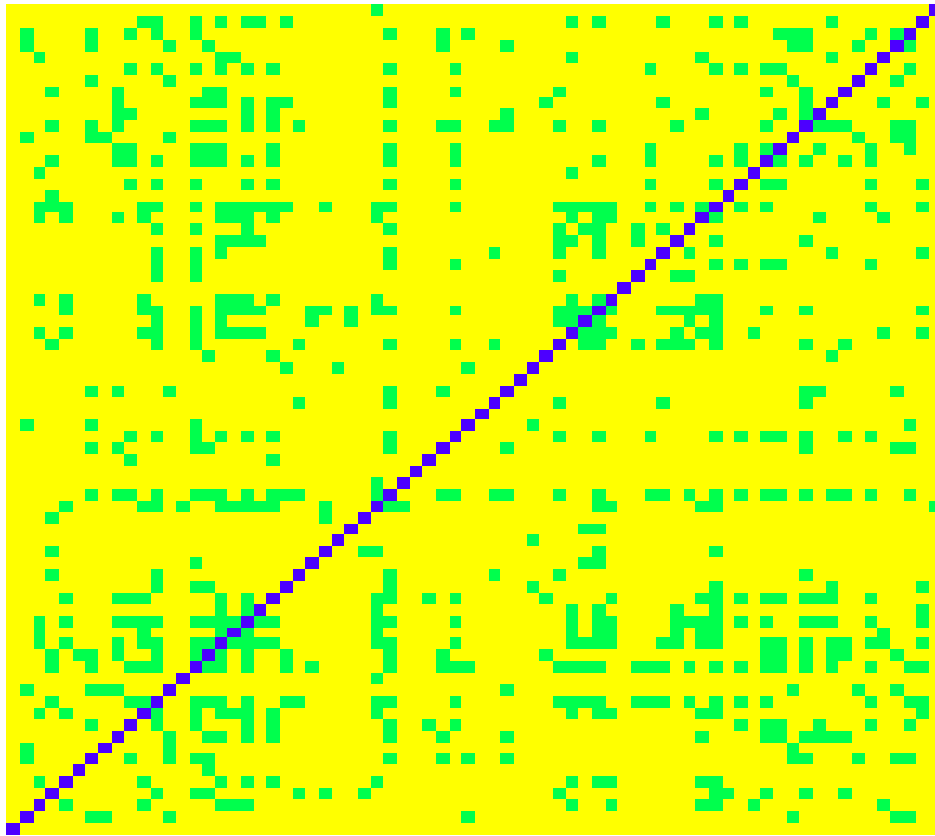


Figure 8 - Data image of the conditional entropy distance matrix from the Golub *et al* data set. The samples are ordered via the UPGMA tree built from the MST edges. While there is some evidence of two clusters, it is not particularly clear.

Conditional entropy was also calculated on an ALL data set contributed to the Bioconductor project by the Dana Farber Cancer Institute. This data set is from the Affymetrix™ U95Av2 chip on which there are 12,625 genes. There are 128 samples in this data set of which 95 are B-cells from acute lymphoblastic leukemia patients while 33 are T-cells from patients with the same disease.

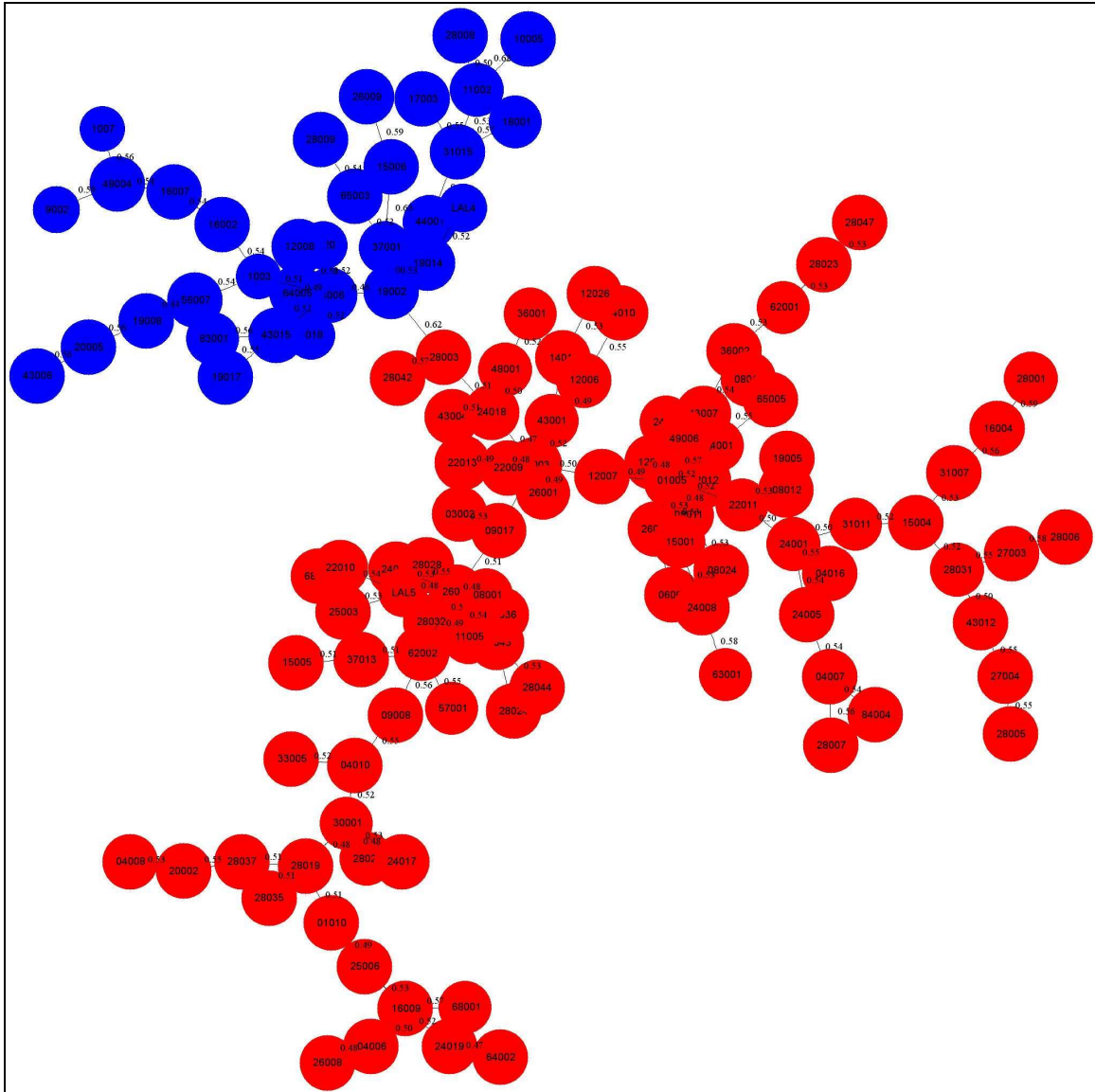


Figure 9 - Minimum spanning tree built from the conditional entropy distance matrix built using the Affymetrix™ U95Av2 ALL data set. The topological separation is clearly evident and indeed only a single edge links the ALL B-cell (in red) and T-cell (in blue) sample groupings.

The resulting MST (see Figure 9) shows a perfect degree of separation between the B- and T-cell samples. The edge weights within each sample type are shorter than those between sample types. Automatic cluster identification (e.g. clique partitioning) would be very effective using this data set. The optimally-sorted data image (see Figure 10) provides solid evidence of the presence of two clusters.

MST-sorted ALL CE

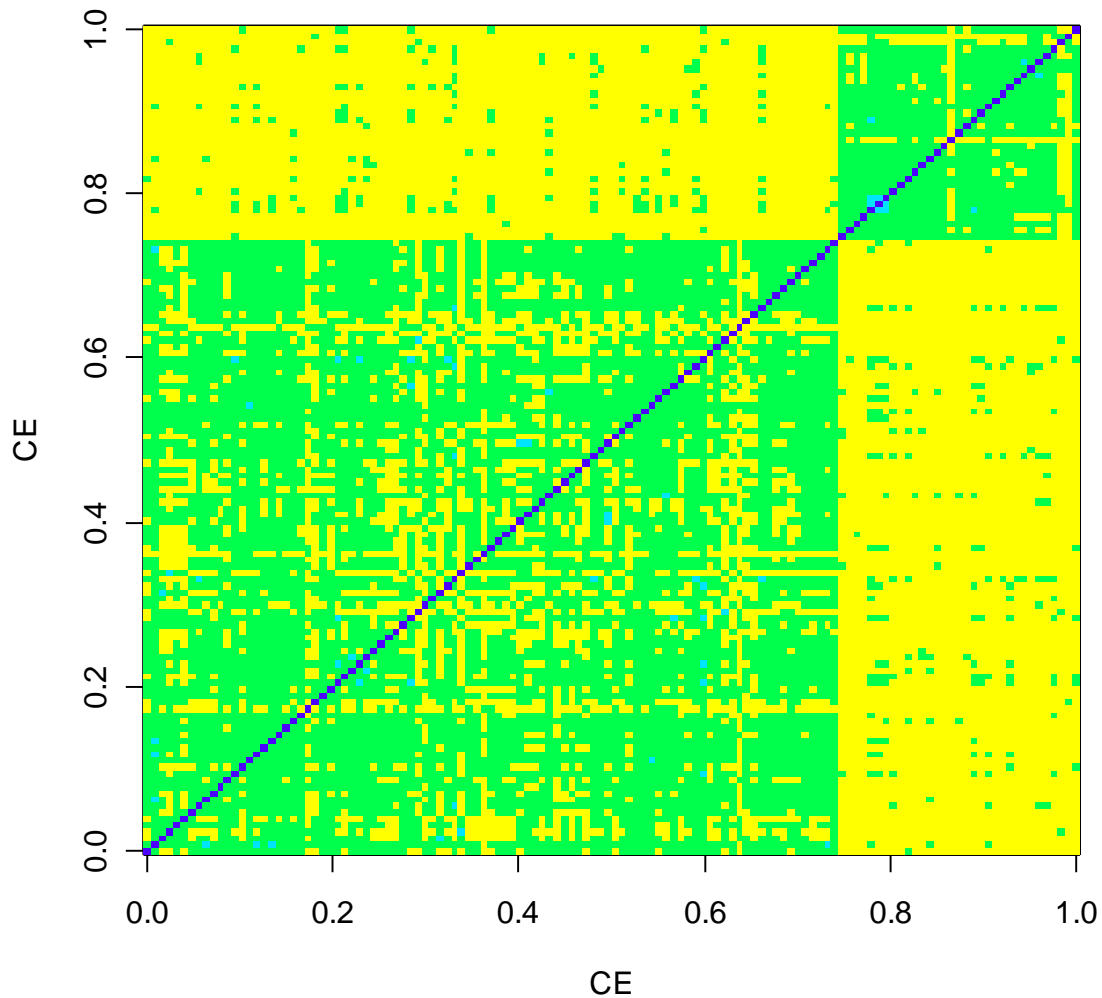


Figure 10 - Data image of the conditional entropy matrix built from the Affymetrix™ U95Av2 ALL data set. The optimal sort order reveals two clear clusters.

Conclusion

Conditional entropy can be a very useful technique for high-level data exploration. It worked very well on an artificial data set but was not quite as informative when applied either to a noise-filled artificial data set or to a rather noisy microarray data set. When applied to a cleaner, Affymetrix™-based data set, the method performed very well. The sensitivity to noise is a potential avenue for future work. Additional future direction may include a visualization tool for MST ordering and incorporation of a clique discovery method for automatic cluster detection based on the fully-connected graph.

Acknowledgements

One of the authors (JLS) would like to acknowledge the support of the Office of Naval Research (ONR) In-house laboratory Independent Research (ILIR) Program.

References

Cheng, C., A. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, USA. (1999)

Getz, G., Levine, E., Domany E. Coupled two-way clustering analysis of gene microarray data. PNAS. 97:22, 12079. (2000).

Golub, T.R. et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science, vol. 286, 531 (1999)

Guo, D. et al. Breaking Down Dimensionality: Effective and Efficient Feature Selection for High-Dimensional Clustering. Workshop on Clustering High-Dimensional Data. (2003)

Guo, D., D. Peuquet and M. Gahegan. Opening the Black Box: Interactive Hierarchical Clustering for Multivariate Spatial Patterns. The 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, VA, USA. (2002)

Pyle, D. Data preparation for data mining. San Francisco, California, Morgan Kaufmann Publishers. (1999).