Clustering with semi metrics

by

Neeraj Koul

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Major Professor: Vasant Honavar

Iowa State University

Ames, Iowa

2001

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Neeraj Koul

has met the thesis requirements of Iowa State University

Major Professor

For the Major Program

For the Graduate College

TABLE OF CONTENTS

ABSTRACT

With the advent of Human Genome Project and other genome sequencing efforts, we are now faced with the challenge of developing not only new methods of data analysis but also improving the already existing methods of data analysis so that they can be better used to take advantage of the data. Here we revisit clustering as a tool for large-scale gene expression(or any other data) analysis. Distance measures are an integral part of any clustering algorithm as a means of capturing similarity between objects. We define a *Universal Distance Measure (UDM)* that is flexible enough to describe a broad class of distance measures. UDM provides a principled meansof translating a user specified notion of "sameness" (under a specified set of transformations) into a well-defined distance measure. We also investigate the process of replacing the two closest objects by a single object. This process is an important step in certain class of clustering algorithms and we call this process as *Methodolgy of Joining (MOJ)* . We investigate some properties of MOJ and establish that the MOJ choice has a critical effect on the results of clustering.

CHAPTER 1. INTRODUCTION

Molecular biologists are currently engaged in some of the most impressive data collection projects. Recent genome-sequencing projects are generating an enormous amount of data related to the function and the structure of biological molecules and sequences. Other complementary high-throughput technologies, such as DNA micro-arrays, are rapidly generating large amounts of data that are too overwhelming for conventional approaches to biological data analysis. The interpretation of this wealth of data is critical to our understanding of life at the molecular level. extraction and representation of biological knowledge from data call for sophisticated computational tools for data analysis.

Cluster analysis is a commonly used tool for analysis of large data sets [JD88] [GLW86]. Recently it has been used with some success in predicting relations between genes [ESBB98] [DIB97]. The basic premise is that genes with similar expression pattern are co-expressed together [SSZ+98]. So, if a gene of unknown function clusters with some gene of known function we may have a clue to its function. Most of the present clustering algorithms have been borrowed from other domains (statistics applied to economics, market research) and applied as such. There is a need for clustering algorithms that can discover more subtle relationships from data. This assumes added significance as we look for more complex relations among genes (besides co-expression). The clustering methodology needs to be made more refined to capture these complex relations. Hence, we develop a methodology which is able to capture user-defined similarities. This is done by defining the distance measure in a novel way. Further, we believe one of the most overlooked areas of research in clustering is how clusters work as a group. We have performed experiments to show that this has a very strong bearing on the results of the clustering. Our methodolgy tries to capture this aspect by formalizing something that we call *Methodology of Joining*. The rest of the thesis is organized as follows. Chapter two

provides a introduction to clustering concepts. Chapter three discusses distance measures and presents universal distance measure which provides a flexible means of capturing the notion of "distance" (or conversely similarity) between objects. Chapter four describes methodology of joining. Chapter five includes summary and scope for future work.

CHAPTER 2. CLUSTERING

2.1 Types of Clustering

The primary goal of cluster analysis, one wishes to partition objects into groups based on given features of each object, so that groups are homogeneous and well separated. Figure 2.2 shows a group of objects (each of which is a point) broken into eleven clusters. Each of the objects in a cluster should be similar to one another and different from elements of other clusters. In context of gene expression analysis the term objects to a vector representing the expression pattrern of a particular gene. Clustering is one of the widely used techniques to derive information from gene expression data. The fundamental thing that is required for any clustering algorithm is the distance/similarity measure. The distance measure allows to check how close two entities are and progressively group the closest entities into a cluster. Clustering can be divided into two major groups

Clustering with representative patterns : In this case the similarity measure is defined only between two objects. At each step of the clustering we need to find a representative of the two objects so that similarity measure is still defined for the group. The major point here is that the two objects are replaced by a single object. The process of replacing the two objects by a single object is called *Methodology of Joining.* This class of clustering uses hierarchical clustering algorithms

Clustering without representative patterns : Here the similarity measure is defined across sets. In this case the closest objects are not replaced by a representative but instead bot the objects become part of the cluster. Since the similarity measure is defined across sets it is still possible to find the two nearest clusters. This class of clustering uses non-hierarchical clustering algorithms

2.2 Similarity Measure

Similarity measure is the fundamental entity in any clustering algorithm as it allows us a way to measure how similar two entities are and group together the most similar elements. The more similar two entities are, the smaller the distance between them. In other words similarity and distance are inversely related and can be used interchangeably. Presently three different classes of distance measures can be recognized.

Euclidean metrics: These measure true straight line distances in euclidean space. Given that objects are characterized as vectors in n-dimensional space, i.e. X, Y in R^n. Then

the euclidean distance $d(X,Y) = sqrt\sum_{i=1}^{n}(x_i - y_i)^2$

Non euclidean metrics: These encapsulate our intuitive perception of space. These are distances that are not straight-line, but which obey certain rules.

It is non-negative, $d(a,b) > 0$.

The distance of an object from itself is zero $d(a,a) = 0$.

It is symmetric, $d(a,b) = d(b,a)$

The triangle inequality holds, i.e. $d(a,c) <= d(a,b) + d(b,c)$. In other words when considering three objects the distance between any two of them can not exceed the sum of the distances between the other two. The Manhattan or City Block metric is an example of this type .

Semi metrics: These obey all the rules of non euclidean metrics. However, in the case of semi metrics d(X,Y) = 0 does not necessarily imply X = Y. The Mutual Information is an example of Semi Metric.

2.2.1 Why is Choice of Distance Measure Important?

The choice of distance measure is important since it basically captures your intuitive notion of similarity. Two objects, which may be close with respect to one distance measure, may not be close with respect to another distance measure. With reference to Fig. 2.1 *Point a* and *Point b* are closest using euclidean distances but if we take correlation coefficient into consideration *Point a* and *Point c* are the closest. Consider the *series d, e and f* in Fig. 2.1. The distance

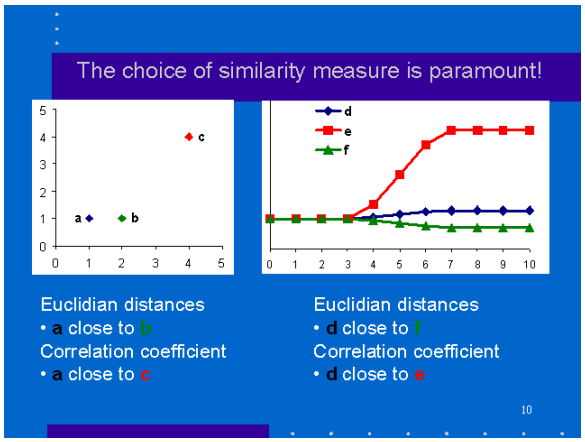

Figure 2.1 Importance of Similarity Measures

between any two series is defined as the sum of distances between its components. With respect euclidean distance *series d* is closest to *series f* but with respect to correlation coefficient *series e* and *series f* are the closest.

Thus the type of distance measure to be used is dependent on the type of similarities we want to capture. To capture only positive correlations, euclidean distance would be adequate. On the other hand, to capture both positive and negative correlations it is required to use pearson's correlation, chi-square or a measure with similar behavior.

2.3 Clustering Algorithm

Once having established how to compute similarity (distance measure) the next step is to choose the clustering algorithm, i.e. the rules which govern how distances are measured between clusters. There are many methods available, the criteria used differ and hence different classifications may be obtained for the same data, even using the same distance measure. The clustering algorithms can be divided into two main classes

2.3.1 Non-Hierarchical

These methods include those techniques in which a desired number of clusters is assumed at the start. Points are allocated among clusters so that a particular clustering criterion is

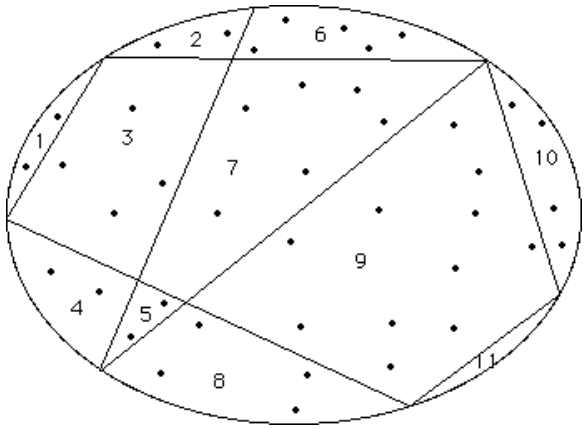

Figure 2.2 A set of points broken into eleven clusters

optimized. A few variants are described below.

Average Linkage Clustering: The distance between clusters is calculated using "average "values. The most common method used is Unweighted Pair-Groups Method Average(UPGMA). First the distance between each point in a cluster and all other points in another cluster is calculated and the sum of all these distances is divided by the total number of points in both the clusters to get the average distance between the clusters. The two clusters with the lowest average distance are joined together to form the new cluster.

Complete Linkage Clustering: This is also called Maximum or Furthest-Neighbor method. The distance between two groups is equal to the maximum distance between a member of cluster i and a member of cluster j.

Single Linkage Clustering: (Minimum or Nearest neighbor Method): The distance between two clusters is the minimum distance between members of the two clusters.

Within Group Clustering: This is similar to UPGMA except clusters are fused so that within cluster variance is minimized. This tends to produce tighter clusters than the UPGMA method.

Ward's Method: Cluster membership is assessed by calculating the total sum of squared deviations from the mean of a cluster. The criterion for fusion is that it should produce the smallest possible increase in the error sum of squares.

An implementation of non-hierarchical method is the K-means clustering algorithm [HW79].

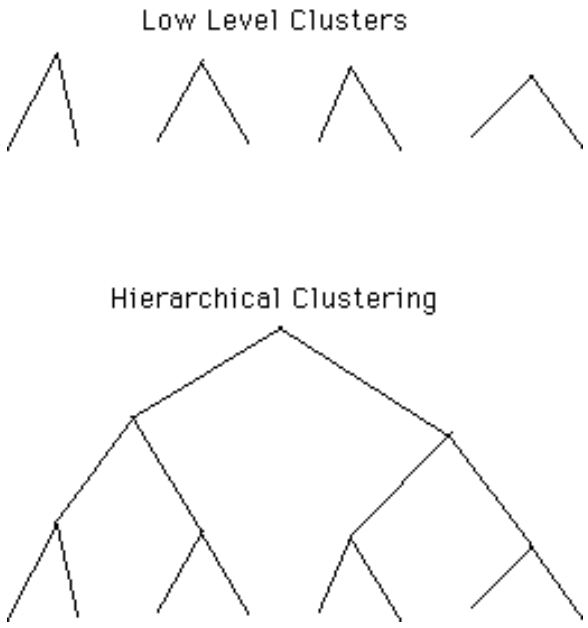

Figure 2.3 Hierarchical Clusters

This nonhierarchical method initially takes the number of components of the population equal to the final required number of clusters. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart. Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroid's position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

2.3.2 Hierarchical

These methods include those techniques where the input data are not partitioned into the desired number of classes in a single step. Instead, a series of successive fusions of data are performed until the final number of clusters is obtained. An example of Hierarchical is Minimum Spanning Tree Clustering Algorithm [Roh73].

CHAPTER 3. UNIVERSAL DISTANCE MEASURE

3.1 Motivation for a New Distance Measure

Any clustering algorithm basically groups together entities which are similar. For this purpose it uses a distance measure. The smaller the distance between the two entities the more similar they are. In this sense there is a inverse relationship between the distance measure and similarity and hence any distance measure can be used as a similarity measure .In this chapter when we refer to similarity measures it means distance measures used as a similarity measure.

Most of the similarity measures used in bioinformatics have been borrowed from statistics applied to domains such as economics,demographics or market research. The most common similarity measures are euclidean distance, chi-square and pearson's coefficient. There are potential pitfalls if such similarity measures are applied per se without understanding how they need to be modified before applying to this domain.

- If we are trying to capture both co-expressed and negatively correlated genes it does not suffice to use euclidean distance as it captures only positive correlation.

- The statistical similarity measures are not intuitive. It takes significant amount of statistical expertise to know what kind of similarity a given measure captures. (e.g., chi-square captures both positive and negative correlations)

- The present similarity measures are not powerful enough to capture complex (e.g. two genes are similar if there gene expressions are the same but shifted in time) or user defined similarities.

- Even when we choose the correct similarity measure we may need to modify it to be applicable to this domain. Consider the pearson's correlation coefficient. For any two

genes X and Y observed over a series of N conditions ,the pearson's correlation coefficient can be computed as follows

$$S(X,Y) = 1/N \sum_{i=1}^{N} (X_i - X_{offse}/\phi_x)(Y_i - Y_{offset}/\phi_y)$$

where

$$\phi_G = \sqrt{\sum_{i=1}^{N} (G_i - G_{offset})^2/N}$$

G_i represents data for gene G in condition i. ϕ_G is the standard deviation of G

In normal statistics X_{offset} represents the mean of the observations of X observed over N conditions. However, if we are trying to capture positive/negative correlations X_{offset} and Y_{offset} should be set to the boundary where genes X and Y are turned ON and OFF respectively.

3.2 Universal Distance Measure

We propose to define a new distance measure D(X,Y) called the universal distance measure. This universal distance measure is defined in terms of a basic distance measure (say euclidean distance). It incorporates user defined similarities. The user specifies the precise conditions under which two objects are the "same" or have zero distance between them. Computing distance between two objects X and Y involves computing all pair of distances between objects same as X and objects same as Y. The minimum of these distances is defined to be the distance between X and Y. Mathematically,

$D(X,Y) = min(d(A,B)|A \in \Gamma(X) and B \in \Gamma(y))$ where $\Gamma(X)$ and $\Gamma(Y)$ represent sets whose members are transformations of X and Y as a result of user defined similarities. Essentially $\Gamma(X)$ is a set that contains all objects that are same as X. It also includes X. The cardinality of $\Gamma(X)$ is a measure of the power of the metric. As an example consider object A with expression profile [1, 1, 1] and object B have expression profile [-0.2, -0.2, -0.2]. Let

the user specify that an object and its negative image (inverse) are the same and the range of values is [-1, 1]. Using UDM the distance between object A and object B is the mimimum of the distance between the following four pairs of objects; A and B i.e. [1, 1, 1] and [-0.2, -0,2, -0,2], A and inverse of B i.e. [1, 1, 1] and [0.2, 0.2, 0.2], inverse of A and B i.e. [-1, -1, -1] and [-0.2, -0.2, -0.2], and inverse of A and inverse of B i.e. [-1, -1, -1] and [0.2, 0.2, 0.2].

3.2.1 Incorporating User View

The user specifies what kind of things are same (say apples are same as oranges) and $\Gamma(X)$ enumerates all objects that are same as X (including X itself).

So, $\Gamma(X) = \{x_1, x_2, x_3, ...\}$ such that $x_i \equiv x_j \, \forall i, j$

The user specifies the set $\Gamma(X)$ implictly or explicitly. The user may provide a function that computes this set or may define same objects in terms of a formula such as X = -X which implies that an object and its negative image are the same. In this case $\Gamma(X) = \{X, -X)\}$. Such a definition will capture all positive and negative correlations. An another example may be $X = X^r$ (an object and its rotation image are the same). This way of defining similarity is very flexible and has the power to capture any abstract notion of similarity. Thus, $\Gamma(apples) = \{apples, oranges\}$ captures the abstract notion of apples being same as oranges.

By definition, $d(X, \gamma(x)) = 0 \, \forall \, \gamma(x) \; where \; \gamma(x) \in \Gamma(X)$. This specifies that by definition the distance between two objects which are same is zero.

3.2.2 UDM is a Semi Metric

The UDM is defined in terms of some basic distance measure and it conserves the properties of the underlying distance measure. As long as the basic measure is a metric, the UDM is a semi metric

Property 1: $D(X, X) = 0$;

This property essentially states that using UDM conserves the property that the distance of an object from itself zero.

Proof: $D(X, X) = min\{d(\gamma(x), \gamma(x))\} = 0$.

Property 2: $D(X, Y) = D(Y, X)$

This property states the the distances measured using UDM are symmetric. The distance from point A to point B is same as the distance from point B to point A.

Proof: $D(X,Y) = min\{d(\gamma(x),\gamma(y))\}$

$= min\{d(\gamma(y),\gamma(x))\}$ as d(a,b) = d(b,a) (distance measures are commutative)

$= D(Y,X)$

Property 3 : $D(X,Y) + D(Y,Z) \geq D(X,Z)$

This property states that distances measured using UDM obey triangle inequality.

Proof: $D(X,Y) + D(Y,Z) = min\{d(\gamma(x),\gamma(y))\} + min\{d(\gamma(y),\gamma(z))\}$

or $D(X,Y) + D(Y,Z) = min\{d(\gamma(x),\gamma(y)) + d(\gamma(y),\gamma(z))\}$ as $d(a,b) \geq 0$ (taking min. out as distances are +ve)

or $D(X,Y) + D(Y,Z) \geq min\{d(\gamma(x),\gamma(z))\}$ (as d obeys triangle inequality)

or $D(X,Y) + D(Y,Z) \geq (X,Z)$

Thus UDM is a semi metric if the underlying distance measure is also a metric. Besides the above properties the UDM also obeys the following properties:

Property 4: If $D(X,Y)$ *and* $D(Z,W) = 0$ *then*

$D(X,Z) = D(Y,W)$ *and*

$D(X,W) = D(Y,W)$

This property states that if two objects are same, then the distance measured from these two objects to a third object is same.

Proof: $D(X,Z) \leq D(X,Y) + D(Y,Z)$ (from triangle inequality)

or $D(X,Z) \leq D(Y,Z)$ as $D(X,Y) = 0$

$D(X,Z) \leq D(Y,Z)$ .... (1)

Similarly, $D(Y,Z) \leq D(X,Z)$ ....(2)

From (1) and (2) we have $D(X,Z) = D(Y,W)$

Similarly, we can prove that $D(X,W) = D(Y,W)$

Corallary : An immediate implication of the above result is that the set enumerated by $\Gamma(X)$ includes the transitive closure of the objects which are at a distance zero from each other.

Property 5: $D(X,Y) = D(\gamma(X),Y) = D(X,\gamma(Y))$

Proof: $D(\gamma(X),Y) = min\{d(\Gamma(\gamma(X)),\Gamma(Y))\}$

or $D(\gamma(X), Y) = min\{d(\Gamma(X), \Gamma(Y))\}$ as $\Gamma(\gamma(X)) = \Gamma(X)$

or $D(\gamma(X), Y) = D(X, Y)$

Similarly, $D(X, \gamma(Y)) = D(X, Y)$

Property 6: $D(X, Y) = min\{d(X, \gamma(Y))\} = min\{d(\gamma(X), Y)\}$

Finding distance between two objects X and Y using UDM involves finding all the pair of distances between objects same as X and objects same as Y.The minimum of these distances is the distance between X and Y. This property states that finding distances between X and all objects same as Y or vice-versa and then taking the minimum will lead to the same result as computed using UDM. This property has a critical effect on the complexity of computing the distance. It reduces the complexity by half on the logrithmic scale.

Proof: $D(X, Y) = min\{d(\gamma(X), \gamma(Y))\} \forall \gamma(X)\ and\ \gamma(Y)$

or $D(X, Y) = min\{d(x_1, \gamma(Y), d(x_2, \gamma(Y), \ldots, d(x_n, \gamma(Y)\}$

or $D(X, Y) = min\{d(x_1, X)+d(X, \gamma(Y)), d(x_2, X)+d(X, \gamma(Y)), \ldots, d(x_n, X)+d(X, \gamma(Y))\}$ as $d(x_i, \gamma(y)) = d(x_i, X) + d(X, \gamma(Y))$

or $D(X, Y) = min\{d(X, \gamma(y))\}$ as $(d(x_i, X) = 0)$

Similarly $D(X, Y) = min\{d(\gamma_x, Y)\}$

This property is important because it states that transformations on one of the elements are sufficient to capture the entire relation.

3.3 UDM Merits

UDM has certain advantages which make it a better distance measure. It incorporates the users view of similarity and therefore provides a flexible means of capturing similarity between objects. This unique way of incorporating similarity makes UDM context sensitive in the sense that its behaviour changes as the users notion of similarity changes. This allows UDM to be applicable across domains. UDM is also able to capture much more complex relations than is possible with other distance measures. As an example $\Gamma(X) = \{X, X^r\}$ specifies that an image and its rotation are the same and will capture all relations of this type. Theoretically, it is possible to incorporate any abstract notion of similarity into UDM. This makes UDM very powerful. An another distinct advantage of UDM is that it is very intuitive. It takes

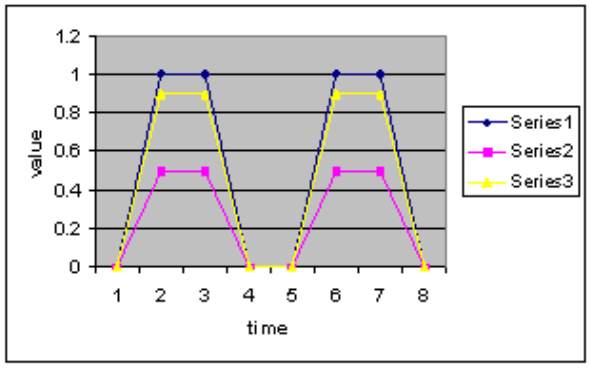

Figure 3.1 Pearsons Vs. UDM

a significant amount of statistical insight to understand what kind of similarities a distance measure captures. A user without significant basis in statistics will find it difficult to look at the formulas of chi-square, mutual information or pearson's and be able to guess intuitively what kind of similarities they capture. However, UDM is very intuitive as it incorporates user view. When a user specifies that a object and its negative image are the same the corresponding UDM is represented as UDM (X = -X). It is intuitive that such as UDM is designed to captures positive and negative correlations.

3.4 UDM(X= -X) vs. Pearson's.

Both pearson's and UDM(X=-X) capture positive and negative correlations. However, they are not exactly equivalent to each other. A clustering algorithm using these two distance measures on the same input data may lead to different results. In this section we explain some of the reasons for this difference.

Consider the expression pattern of three series (genes) which have the same direction but different levels of expression as shown in Fig. 3.1 . According to pearson's metric all three series are perfectly correlated and hence it is not able to distinguish between them. This seems to be a limitation of pearson's as *series 1* very closely matches *series 2* while *series 3* is relatively far from it in terms of expression level(both in positive and negative sense). On

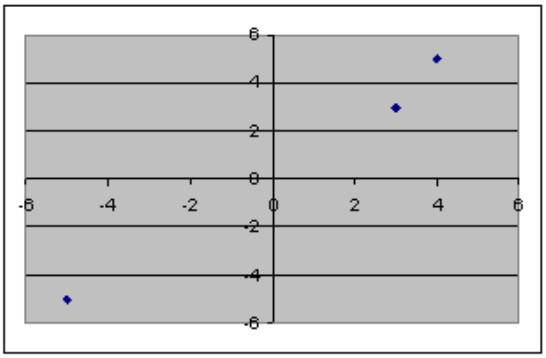

Figure 3.2 Pearson Vs. UDM

the other hand UDM(X= -X) is powerful enough to see this difference and picks out the two closest Series.

As an another illustration consider the following three points viz. A(-5,-5),B(3,3) and C(4,5) as shown in Fig. 3.2. Pearson's picks *point A* and *point B* as the two closest points while UDM picks *point A* and *point C* as the closet. Such differences lead to variation in the clusters generated by using these two distance measures.

CHAPTER 4. METHODOLOGY OF JOINING

For clustering with representative patterns it is required to find representatives for the two closest objects that are clustered together. In other words the two profiles are replaced by a single profile that is a representative of the two profiles. The particular way of joining of two profiles to produce a new representative profile is called *Methodology of Joining (MOJ).* MOJ has a significant bearing on the results of the experiment. We have performed simulations which show that even with same distance measure use of different MOJ results in very different clusters. The important requirement for MOJ is that it should capture the way how two clusters/profiles work as a group. The most popularly used MOJ has its basis in the fact that the earlier algorithms for clustering used euclidean distance. The merging of the two profiles was carried as follows. Let X (w_1)and Y (w_2) be the two expression profiles which need to be replaced by a representative profile where w_1 and w_2 are the weights associated with X and Y respectively.

Then $MOJ(X,Y) = X : Y(w_1 + w_2) = X * w_1 + Y * w_2/(w_1 + w_2)$

Where '*' implies multiplication and ':' is the MOJ operator.

As the distance measures were changed for capturing more complex relations (e.g. to chi-square, mutual information), the corresponding MOJ remained unchanged. The MOJ needs to be tied in with the distance measure used as MOJ effects the distances between profiles.

4.1 A Proper MOJ

Even while using a arbitrary MOJ we may achieve some insightful results. However, a proper (valid) MOJ is one in which satisfies the following two properties

Property 1 : For a MOJ to be valid it has to be order invariant

i.e. $X : Y = Y : X$

In other words replacing X by Y and vice-versa in the formula for MOJ should not make any difference. -XY is commutative but not inversable. This is because $(-X)(Y) = (Y)(-X) \neq (-Y)(X)$.

Explanation: Otherwise the results of the clustering experiment are dependent on the order of the input file. Consider the two data sets D_1 and D_2. Assume that both D_1 and D_2 contain the expression profile of genes G1, G2 and G3. Let the order of data in D_1 be G1, G2 and G3 while in D_2 it be G2, G1 and G3. Let G1 and G2 have the two closest expression profiles. Assume for the sake of contradiction that $G1 : G2 \neq G2 : G1$. Let us construct two new data sets $D_1\prime$ and $D_2\prime$ from $D_1$and D_2 respectively by *appending* an expression profile for G4 to them respectively. The expression profile for G4 is such that G1:G2 is closest to G4 but G2:G1 is closest to G3 . Now the results of running the clustering algorithm on $D_1\prime$ is ((G1,G2)G4)G3) while running on $D_2\prime$ is ((G1,G2)G3)G4). This is a contradictory result since $D_1\prime$ and $D_2\prime$ are essentially the same data set but with different ordering. So, order invariance is a required property for a MOJ to be valid.

Property 2: Assume that each element of X takes values between the ranges 'a' and 'b' i.e. X,Y $\in [a, b]$. Then $\forall X, Y(X : Y) \in [a, b]$ where X:Y represents that object obtained after joining X and Y.

Explanation: If all the values of the components of a profile lie within some min. and max. value (for sake of interpretation), the representative profile should also lie within this range as otherwise we cannot interpret what the values means. This should hold for all possible values that the components of the profile take. This property can be relaxed if we know that for certain cases some of the possible values never occur.

Any MOJ which satisfies properties 1 and 2 stated above is said to be proper.

Consider $X(w1) : Y(w1) = -YX(w1, w2)$. This is not proper MOJ since it does not satisfy inversability. Similarly $X(w1) : Y(w1) = X + Y(w1 + w2)$ is not proper as it does not satisfy property2 for the case $(X : Y) \in [0, 1]$. However, $X : Y(w_1 + w_2) = X * w_1 + Y * w_2/(w_1 + w_2)$ is proper as it satisfies both the properties.

4.2 Desirable Qualities of a MOJ

In the above we discussed the properties required for a MOJ to be proper. Besides this there are some properties which lead to a good MOJ in the sense that the representative closely matches the two profiles from which it was generated.

4.2.1 Property of Similar Profiles

If $X = Y$ *then* $X : Y = X$

The basis for this profile comes from the fact that if two sets have the same profile then their representative can be either of the profiles. The more different the representative profile is from either of the two profiles the worse of the MOJ will be. This property can provide an insight into the quality of the MOJ.

A more general statement of the above property can be arrived at from the following reasoning. In case of exactly similar profiles the representative profile can be on of the two profiles or a profile equivalent to the two profiles. Since we use the representative profile in computing distances the equivalent profile will mean one which is at distance zero from either of the two profiles. So, if $X = Y$, then $X : Y = Z$ where $Z \in (\alpha | D(\alpha, X) = 0)$ where D is the distance measure used. An alternate way of stating this is that if $D(X : Y, Z) = D(\mathcal{F}(X : Y), Z)$ then for $X = Y, X : Y = \mathcal{F}(X : Y)$

4.2.2 Property of Associativity

MOJ $(X : Y) : Z = X : (Y : Z)$

This follows from the fact that both (X:Y):Z and X :(Y : Z) is a representative of the profiles of X, Y and Z taken together and hence should have the same or equivalent profiles.

4.3 Correspondence with Similarity Measure

MOJ should correspond to some nice intuitive notion (say avg., union of sets etc.) and should tie in very well with the distance measure used. For sake of illustration consider the case when we use euclidean distance and $MOJ(X, Y) = X : Y(w_1 + w_2) = X * w_1 + Y * w_2 / (w_1 + w_2)$

Here $D(X : Y, Z) = \alpha(D(\beta X, Y) + D(\gamma X, Z)$ where D calculates the euclidean distance and α, β, γ are constants defined in terms of the weights. Intuitively we are calculating the distance from center of mass of X,Y to Z. Hence the said MOJ ties in very well with the euclidean distance. However, for mutual information or chi-square the same MOJ does not have any intuitive correspondence.

4.4 Comments on MOJ

In this section we make some comments about the MOJ.

4.4.1 MOJ (:) is a Binary Operator.

MOJ is an operator that is defined for two expression profiles. However, it can be repeatedly applied at each step to join two or more profiles. Thus MOJ (X, Y, Z) does not make sense until it implies MOJ (X, MOJ (Y, Z)) or MOJ (Y, MOJ (Z, X)) or MOJ (Z, MOJ (X,Y)).This remark can become important during interpretation. Consider the following example.

Let $MOJ(X, Y) = X : Y + X + Y - XY$. This formula corresponds to union of set X and set Y (Venn Diagram). Now extending it to three sets we have $MOJ(X, Y, Z) = (X : Y) : Z = X + Y + Z - XY - YZ - ZX$. In this case it does not seem to corresponds of union of sets as $A \bigcup B \bigcup C = A + B + C - AB - BC - CA + 3ABC$. However, if we consider the fact that ':' is a binary operator and we first do X:Y and then join together X:Y and Z we can see that it indeed gives the union of sets (This can be verified using Venn diagram).

4.4.2 Dependence of MOJ on Type of Values and Operators

The validity of MOJ is dependent on values of the expression profile and the operators defined on them. Let $MOJ(X, Y) = -X * Y + X$. Consider the case when X and Y are binary vectors and '-', '+" and '"*' represent logical NOT, OR and AND respectively. The given MOJ is proper in this case as it is proper and X:Y is again a binary vector. Consider another case where the elements of X and Y can take real values between zero and one. Let '-', '+' and '*' represent fuzzy NOT,OR and AND respectively. In this case the given MOJ is not proper as

it does not obey order invariance. This is because for the case X+Y $\geq$ 1, X: Y = -XY + X = -X + X while Y:X = -YX + Y = -Y +Y.

4.5 Experiments Conducted

The experiments were designed to test that with all other parameters remaining the same, MOJ has a critical effect on the results of the clustering algorithm. This set of experiments was carried on both synthetic and real data sets.

4.5.1 Effect of MOJ Choice

Ths set of experiments was designed to show that MOJ has a critical effect on the results of the clustering. Let X and Y represent the two profiles to be merged. Let their weights be w_1 and w_2 respectively. Let $X{:}Y$ represent the resulting merged profile. The operators '-', '+' and '*' have their normal meaning. The following MOJ were studied:

- *averageMergeWeight* $X : Y(w_1 + w_2) = X * w_1 + Y * w_2/(w_1 + w_2)$
- *minMergeWeight* $x_i : y_i(w_1 + w_2) = (min(x_i, \overline{x_i}) * w_i + min(y_i, \overline{y_i}) * w_2)/(w_1 + w_2)$. Here x_i and y_i represent the individual components of X and Y respectively.
- *maxMergeWeight* $x_i : y_i(w_1 + w_2) = (max(x_i, \overline{x_i}) * w_1 + max(y_i, \overline{y_i}) * w_2)/(w_1 + w_2$. Here x_i and y_i represent the individual components of X and Y respectively. This is same as minMergeWeight but with min. replaced by max.
- *unionMergeGeneral* $X : Y(w_1 + w_2) = X + Y - X * Y$. This does not take weights into consideration.
- *unionMergeMax* $x_i : y_i(w_1 + w_2) = a_i + b_i - ai * b_i \; where \; a_i = min(x_i, \overline{x_i}) \; and \; b_i = min(y_i, \overline{y_i})$. This does not take weights into consideration.x_i and y_i represent the indiv idual components of X and Y respectively.
- *unionMergeMin*

$x_i : y_i(w_1 + w_2) = a_i + b_i - ai * b_i \, where \, a_i = max(x_i, \overline{x_i}) \, and \, b_i = max(y_i, \overline{y_i})$. This does not take weights into consideration. x_i and y_i represent the individual components of X and Y respectively.

- *universalMerge* $X : Y(w_1 + w_2) = A * w_1 + B * w_2/(w_1 + w_2)$ where $A = min(X, \overline{X})$ and $B = min(Y, \overline{Y})$. The min. function here returns the vector(profile) having the lower magnitude(of first order)

The first data set (D_1) chosen was synthetic data. It was randomly generated and consisted of tens objects/genes each having sixteen observations. The observations had values through zero to one. To study the effect of MOJ, the distance measure was kept same and the MOJ was varied. The results showed that MOJ had a significant effect on the results even when the same distance measure was used. This behavior was consistently observed with almost every data set that was generated. As an illustration the results using MOJ averageMergeWeight and unionMergeGeneral are shown in the Fig. 5.1 and Fig. 5.2 respectively. The euclidean distance was used in both the cases. As can be seen from the diagram the genes cluster differently in these two cases. As an example when MOJ averageMergeWeight is used (Fig. 5. 1) gene 6d..6 clusters with gene 3d..3. On the other hand when MOJ unionMergerGeneral (Fig. 5.2) is used the gene 6d..6 clusters with the cluster consisting of genes 8d..8 and 2d..2. An important observation arrived at by studying the results was that there needs to be some sort of association between the MOJ and the distance measure used. When using pearson's metric the results obtained using MOJ minMergeWeight and maxMergeWeight are exactly same. This is expected as pearson's metric captures both positive and negative correlations and hence it does not matter whether MOJ takes the maximum value or its ineverse.

Similar results were obtained when the number of objects/genes was increased. To make sure that this phenomenon was also observed actual data,the experiments was run on a subset of data obtained from [SSZ+98]. The subset of data on which to run the experiments was randomly chosen. It was consistently found that different MOJ leads to different results by the clustering algorithm.

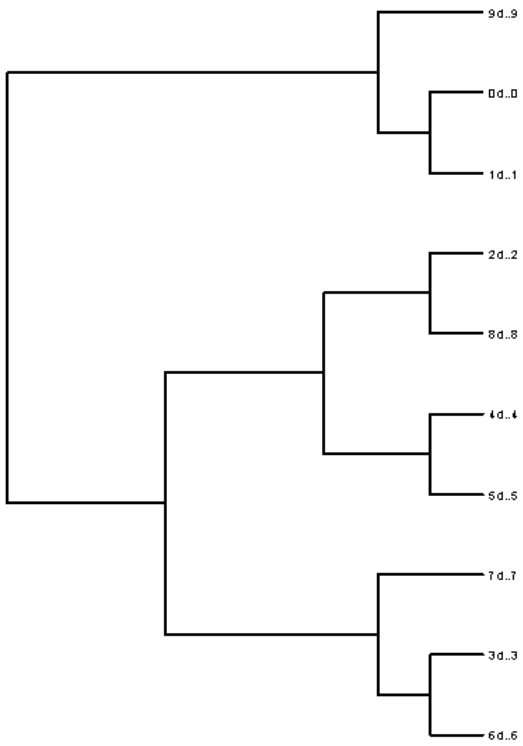

Figure 4.1 Euclidean and averageMergeWeight

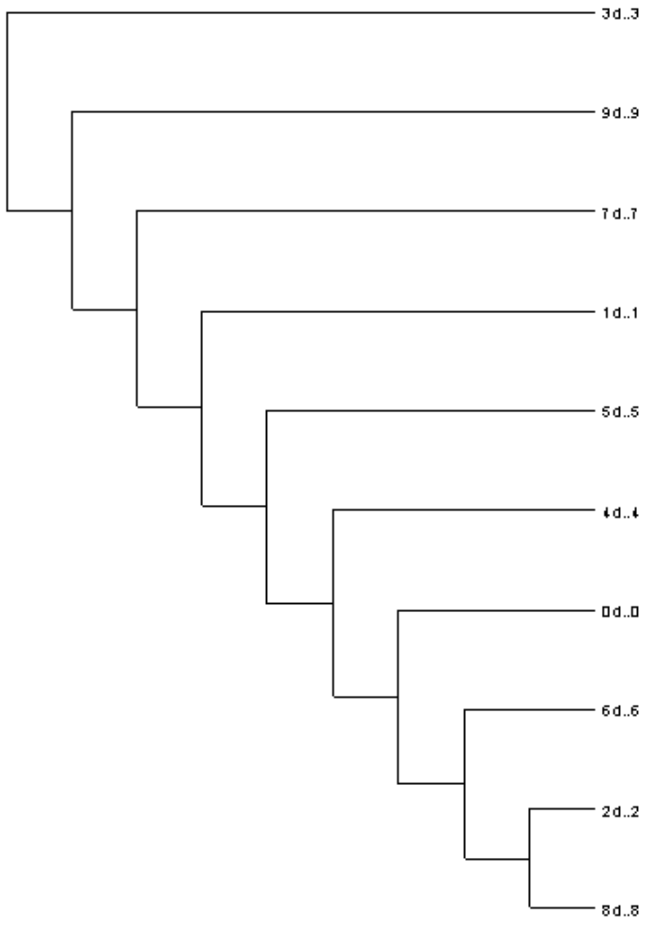

Figure 4.2 Euclidean and unionMergeGeneral

CHAPTER 5. Future Work

5.1 Summary

Universal Distance Measure provides a powerful of translating the users view of sameness into a well defined distance measure. UDM is flexible, intuitive and has the capabilty to capture any abstract notion of similarity. We established that UDM is a semi metric as long as the underlying distance measure is a metric. We also compared UDM under condition X =- X with pearson's metric and showed that UDM captures realtion which can different than those captured by pearson's metric. We also studied Methodology of Joining and stated the properties under whic a given MOJ is valid. We also established experimentally that the choice of MOJ has a critical effect on the results of the clustering.

5.2 Future Work

There is plenty of scope for future work. A few suggested areas are as follows:

- Establish those class of distance measures that can be expressed in a concise way using UDM.
- Expressing the well known distance measures in terms of UDM
- Study how MOJ interacts with UDM or any other distance measure so as to specify which MOJ to use for given distance measure.
- Cluster a publicly available data set using UDM and compare the results with those obtained from other clustering algorithms.
- Study the possibilt of using UDM in other domains such as pattern matching and robotics (finding nearest neighbor).

BIBLIOGRAPHY

[DIB97] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

[ESBB98] M. Eisen, P. Spellman, D. Botstein, and P. Brown. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Science USA*, 95:14863–14867, 1998.

[GLW86] A. Griffiths, H. Luckhurst, and P. Willett. Using interdocument similarity in document retrieval systems. *Journal of the American Society for Information Science*, 37(1):3–11, 1986.

[HW79] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[JD88] Jain and Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[Roh73] F. James Rohlf. Algorithm 76: Hierarchical clustering using the minimum spanning tree. *The Computer Journal*, 16(1):93–95, 1973.

[SSZ+98] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle -regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9:3273–97, 1998.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Vasant Honavar for his guidance, patience and support throughout this research and the writing of this thesis. I would also like to thank my committee members for their efforts and contributions to this work: Dr Drena Dobbs and Dr. Leslie Miller. I would additionally like to thank Adrian Silvescu for the inumerable discussions on various aspects of this research. This research was supported in part by a research assistantship funded by the grant from Carver Foundation to Dr. Vasant Honavar and a teaching assistantship funded by the Iowa State University Computer Science department.