International Journal of Biological, Life and Agricultural Sciences
ISSN: 2415-6612
Vol:2, No:3, 2008

Clustering Protein Sequences with Tailored General Regression Model Technique

G. Lavanya Devi, Allam Appa Rao, A. Damodaram, GR Sridhar, and G. Jaya Suma

Abstract**—**Cluster analysis divides data into groups that are meaningful, useful, or both. Analysis of biological data is creating a new generation of epidemiologic, prognostic, diagnostic and treatment modalities. Clustering of protein sequences is one of the current research topics in the field of computer science. Linear relation is valuable in rule discovery for a given data, such as if value X goes up 1, value Y will go down 3", etc. The classical linear regression models the linear relation of two sequences perfectly. However, if we need to cluster a large repository of protein sequences into groups where sequences have strong linear relationship with each other, it is prohibitively expensive to compare sequences one by one. In this paper, we propose a new technique named General Regression Model Technique Clustering Algorithm (GRMTCA) to benignly handle the problem of linear sequences clustering. GRMT gives a measure, GR^*, to tell the degree of linearity of multiple sequences without having to compare each pair of them.

Keywords**—**Clustering, General Regression Model, Protein Sequences, Similarity Measure.

I. Introduction

CLUSTER analysis provides an abstraction from individual data objects to the clusters in which those data objects reside. Some clustering techniques characterize each cluster in terms of a cluster prototype which is a data object that is representative of other objects in the cluster. Cluster analysis is sometimes referred to as unsupervised classification. The main objective of this unsupervised classification is to find a natural grouping or meaningful partition by using distance or similarity function. Clustering is mainly used for dimensionality reduction, prototype selection, or abstraction for pattern classification, data reorganization and indexing and for detecting outliers and noisy patterns. Clustering techniques are applied in pattern classification schemes, bioinformatics, data mining, web mining, biometrics, document processing, remote sensed data analysis, biomedical data analysis etc., in which data size is very large. There are many types of clustering techniques namely hierarchical clustering, partitional clustering, exclusive clustering, non-exclusive clustering, and fuzzy clustering[29,30].Clustering is an active research topic in pattern reorganization, data mining, statistics and machine learning with diverse prominence.

Protein sequences have a remarkable ability to reproducibly fold into a three dimensional shape and this shape confers them to the ability to form a variety of critical for life: enzymatic catalysis, structural support, generation of motion, reception of signals between cells, and transduction of forces into chemical signals, to name a few [31]. Molecular biology has undergone an incredibly rapid development, currently yielding huge amounts of raw data that efficient computer algorithms are mandatory for data analysis. The number of unique entries in all protein sequence databases together exceeds now more than half a million. However biological evolution lets proteins fall into so called families, thus imposing a natural grouping. A protein family contains sequences that are evolutionarily related and or share a common three dimensional fold. Similar protein sequences probably have similar biochemical function and three dimensional structure. Protein sequence clustering helps in classifying a new sequence, retrieve a set of similar sequences for a given query sequence, predicting the protein structure of unknown sequence and finding the family and subfamily relationships of protein sequences.

Sequence analysis has attracted a lot of research interests with a wide range of applications. While matching, sub-matching, indexing, clustering, rule discovery, etc. are the basic research problems in this field [1] - [8], [23, 24], the core problem is how to define and measure similarity. Currently, there are several popular models used to define and measure (dis)similarity of two sequences.

The methods can be classified into four main categories:

#Lp norms [1, 2]

Given two sequences X =[x_1, x_2,.., x_N]and Y = [y_1, y_2,..... , y_N], Lp norm is defined as $Lp(X, Y) = (\sum_{i=1}^{N} |x_i - y_i|^{1/p}$ When p=2,it is the most commonly used Euclidean distance. Lp norms are straightforward and easy to calculate. But in many cases, the distance of two sequences cannot reflect the real (dis)similarity between them. A typical case is shifting. For example, suppose sequence X_1 =[1, 22,...,30]and X_2 = [301,302,…,330]. X_2 is the result of shifting X_1 by 300, i.e., adding 300 to each element of X_1. The Lp distance between X_1

Manuscript received March 31, 2008.

G. Lavanya Devi is with Department of Computer Science and Engineering, Gitam Institute of Technology, Gitam University, Visakhapatnam, Andhra Pradesh, India (phone: 91-0891-2840270, e-mail : lavanyadevig@yahoo.co.in).

Allam Appa Rao, PhD, is with Andhra University College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India (phone: 91-0891-2844999, e-mail: apparaoallam@gmail.com)

A Damodaram, PhD, is with Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India (phone: 9440843366, e-mail: damodarama@gmail.com).

GR Sridhar, PhD, is with Endocrine and Diabetes Centre, Visakhapatnam, Andhra Pradesh, India. Also he is adjunct professor in Bioinformtics at Andhra University College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India (phone: 91-0891-2844999; e-mail: sridharvizag@gmail.com).

G. Jaya Suma is with Department of Computer Science and Engineering, Gitam Institute of Technology, Gitam University, Visakhapatnam, Andhra Pradesh, India (phone: 91-0891-2840270; e-mail: jayasuma@gitam.edu).

and X_2 is large, but actually they should be considered to be similar in many applications [10, 16, 17]. Another case is scaling. For example, let $X_2 = \beta X1$, where β is a scaling factor. In some applications, we also need to consider X_2 to be similar to X_1. Obviously, Lp norms cannot capture these types of similarity. Furthermore, Lp distance only has relative meaning when used to measure (dis)similarity. By "relative", we mean that a distance alone between two sequences X_1 and X_2,e.g., $Distance(X_1,X_2) = 95.5$, cannot give us any information about how (dis)similar X_1 and X_2 are. Only when we have another distance to compare, e.g., $Distance(X_1,X_3) = 100.5 > 95.5$, we can tell that $X1$ is more similar to X_2 than to X_3. In conclusion, Lp norms as measure of (dis)similarity have two drawbacks:

• Cannot capture similarity in the case of shifting and scaling.

• Distance only has relative meaning of (dis)similarity.

It is known that the mean-deviation normalization can discard the shifting and scaling factors. The mean-deviation normalization is defined as $Normal(X) = (X - mean(X))/std(X)$. However, it can not tell what the shifting and scaling factors are. Those factors are exactly what we need to mine the linearity of sequences.

#Transforms [3, 21, 22]

Popularly used transforms in sequences are the Fourier Transform and Wavelet Transform. Both transforms can concentrate most of the energy to a small region in the frequency domain. With energy concentrated to some a small region, processes can be carried out in this small region involving only few coefficients, thus dimension is reduced and time is saved. From this point of view, the transforms are used actually for feature extraction. However, after features are extracted, some type of measure is unavoidable. If Lp norm distance is used, it inherits the disadvantages stated above.

#Time Warping [18, 19, 20]

It defines the distance between sequences $X_i = [x_1, x_2, \dots, x_i]$and $Y_j = [y_1, y_2, \dots, y_j]$as $D(i, j) = |x_i - y_i| + min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}$. This distance can be solved using dynamic programming. It has a great advantage that it can tolerate some local non-alignment of time phrase so that the two sequences do not have to be of the same length. It is more robust and flexible than Lp norms. But it is also sensitive to shifting and scaling. And the warping distance only has relative meaning, just like the Lp norms.

#Linear relation [10, 16, 17]

Linear transform is $Y = \beta_0 + \beta_1 X$. Sequence X is defined to be similar to Y if we can determine such β_0 and β_1 so that $Distance(Y, \beta_0 + \beta_1 X)$ is minimized and this distance is below a given threshold. Paper [16] solved scaling factor β_1 and shifting offset β_0 from a geometrical point of view. Although $Distance(Y, \beta_0+\beta_1 X)$ is invariant to shifting and scaling, the distance still only has relative meaning.[8]

In this paper, we propose a new model, named GRMT (General Regression Model Technique) to measure the degree of the linear relation of multiple sequences at one time. In addition, based on GRMT, we develop techniques to cluster massive linear sequences accurately and efficiently.

The organization of this paper is as follows: Section1 is introduction; Section 2 provides a basic background of the classical regression model. Section 3 describes GRMT in detail and section 4 shows applications and examples of how to apply GRMT clustering algorithm to linearity measure and clustering of multiple sequences. Finally section 5 will draw conclusions.

II. Background of Regression Model

Linear regression analysis originated from statistics and has been widely used in econometrics [27, 28]. For an instance, to test the linear relation between consumption Y and incoming X, we can establish the linear model as:

$$Y = \beta_0 + \beta_1 X + u \tag{1}$$

The variable u is called the *error term*. The regression as (1) is termed as"*the regression of Y on X*". Given a set of sample data, $X = [x_1, x_2, \dots, x_N]$and $Y=[y_1, y_2, \dots, y_N]$, β_0 and β_1 can be estimated in the sense of minimum-sum-of-squared-error. That is, we seek to find a line, called regression line, in the $Y-X$ space, to fit the points $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ as well as possible. We need to determine β_0 and β_1 such that

$$\sum_{i=1}^{N} u_i^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i) \text{ is minimized.}$$

Using first order conditions [27, 28], we can solve β_0 and β_1 as follows:

$$\beta_0 = \sum_{i=1}^{N} \bar{y} - \beta_1 \bar{x} \tag{2}$$

$$\beta_1 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{3}$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\frac{1}{N}\sum_{i=1}^{N} y_i$, the average of sequence Y and X respectively.

After obtaining β_0 and β_1, we have to measure how well the regression line fits these data. To answer this, the R^* is defined as:

$$R^* = 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{4}$$

The value of R^* is always between 0 and 1. The closer the value is to 1, the better the regression line fits the data points. R^* is the measure for the *Goodness-of-Fit* in the traditional regression. The regression model as (1) is called *Simple Regression Model*, since it involves only one independent variable X and one dependent variable Y . We can add more independent variables to the model as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_K X_K + u \tag{5}$$

This is called *Multiple Regression Model*. $\beta_0, \beta_1, \dots, \beta_K$ can be estimated similarly using first order conditions.

III. Classical Regression Model

We observed that the Classical Regression Model is excellent in testing the linear relation of two sequences. R^* is a good measure for linear relation. For an instance, $R^*(X_1,X_2) = 0.95$ is statistically strong evidence that the two sequences are highly linear related to each other, thus they are very similar (if we think similarity should be invariant to shifting and scaling). We do not have to compare $R^*(X_1,X_2) > R^*(X_1,X_3)$ and say X_1 is similar to X_2 rather than X_3. Therefore, the meaning of R^* for similarity is not relative, unlike distance-based measures.

When we need to test only two sequences, the Simple Regression Model is suitable. However, when more than two sequences are involved in some applications such as clustering, the Classical Regression Model has to run regression between each pair of sequences. The performance cannot be efficient. One might be tempted to think that we can use the Multiple Regression Model. Unfortunately, there exists a critical problem in the Multiple Regression Model. We cannot use R^* in the multiple regression model to test whether multiple sequences are similar to each other or not, because it only means the linear relation between Y and the *linear combination* of $X_1,X_2,\ldots,X_K$. Moreover, R^* in the multiple regression is sensitive to the order of sequences. If we randomly choose X_i to substitute Y as dependent variable and let Y be independent variable, then the regression becomes

$X_i = \beta_0 + \beta_1 X_1 + \ldots\ldots + \beta_i Y + \ldots\ldots + \beta_K X_K + u$. The R^* here will be different from that of (5), because they have different meanings.

From a geometrical point of view, equation (5) describes a hyper-plane instead of a line in $(K+1)$-dimensional space. To test the similarity among multiple sequences, we need a line in the space instead of a hyper-plane. [11, 12, 13, 14, 15]

Generalizing the idea of Classical Regression Model to multiple sequences, we propose the General Regression Model Technique (GRMT).

GRMT: Generalized Regression Model Technique
Given $K(K \geq 2)$ sequences $X_1,X_2,\ldots\ldots,X_K$ and

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots x_{1N} \\ x_{21} & x_{22} & \cdots x_{2N} \\ \vdots & \vdots & \vdots \\ x_{K1} & x_{K2} & \cdots x_{KN} \end{pmatrix}$$

We first organize them into N points in the K dimensional space.In the traditional regression, the error term is defined as:

$$u_i = y_i - (\beta_0 + \beta_1 x_{1i} + \ldots\ldots + \beta_K x_{Ki}) \quad (6)$$

It is the distance between y_i and the regression hyper-plane in direction of axis Y. This makes sequence Y unique from any X_i ($i = 1, 2, \ldots, K$). In GRMT, we define the error term u_i as the *vertical* distance from point $(x_{1i}, x_{2i}, \ldots, x_{Ki})$ to the regression line. Please note that there is no Y here anymore, because no sequence is special among its community. To guarantee the regression line exists uniquely, we need following two assumptions:

No sequence is constant. It guarantees the scatter matrix has eigenvector.

#N points determine a line uniquely.In real applications, it is highly unlikely that a random sequence is constant or all K sequences are exactly the same. Therefore, the assumptions will not limit the applications of GRMT. Similar to the traditional regression, after determining the regression line, we need a measure for Goodness-of-Fit. We define:

$$GR^* = 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{j=1}^{K}\sum_{i=1}^{N}\left(x_{ji} - \overline{x_j}\right)^2} \quad (7)$$

IV. Applications of GRMT

The procedure of applying GRMT to measure the linear relation of multiple sequences is described by algorithm GRMT1.

GRMT1: Testing linearity of multiple sequences

• Organize the given K sequences with length N into N points $p_1, p_2, \ldots, p_K$ in K-dimensional space as shown in section3.2.

• Determine the regression line. First, calculate the average $m = \frac{1}{N}\sum_{i=1}^{N} p_i$ calculate the scatter matrix $S = \sum_{i=1}^{N} (p_i - m)(p_i - m)^t$.

Then, determine the maximum eigen value λ and corresponding eigenvector e of S.

• Calculate GR^* according to property 1 of GR^*.

• Draw conclusion. Suppose we only accept linearity with confidence no less than C (say, $C = 85\%$). If $GR^* \geq C$, we can conclude that the K sequences are linear to each other with confidence GR^*.

Suppose we want to test two sequences X_1 and X_2 and X_1 = [5, 1, 8, 17, 27, 10]; X_2 = [17, 10, 25, 34, 12, 31],

First, organize the two sequences into 6 points: (5, 17), (1, 10), (8, 25), (17, 34), (27,12), (10,31).

Second, determine the regression line. Average m = $\frac{1}{N}\sum_{i=1}^{N} p_i$ = [11.33, 21.50]^t. Maximum eigen value λ = 1942.3 and corresponding eigenvector e = [0.4657, 0.8849]^t.

Third, calculate GR^*. We can conclude that X1 is related toX2. Also we find their relation as $\frac{X1 - 11.33}{0.4657} = \frac{X2 - 21.50}{0.8849}$

GRMT1 is intended to test whether multiple sequences are linear to each other or not. Consider an example for testing 3 sequences at a time. Suppose we have three sequences:X_1 = [6, 9, 13, 16, 12, 11, 16, 20, 19, 23]; X_2 = [8, 13, 13, 17, 13, 18, 16, 13, 17, 19]; X_3 = [5, 9, 12, 14, 17, 18, 17, 15, 13, 13].

Following the same procedure, we can calculate GR^* =0.7301. This confidence is not much high, thus we can conclude that some sequences are not very linear to others. This example demonstrates that GR^* is a good measure again.

We have tested many sequences and found GR^* as linearity measure agrees with our observation.

Proteins are strings of combination of the twenty amino acids. Each of the amino acid is given a significant weight. Also all the N sequences that are to be clustered may not have the same length. The sequence that has maximum length X_m is considered and all other (N-1) sequences are to be padded with a neutral value. Truncating the sequences to a fixed length may lead to loss of useful information. The procedure followed above prevents us from loosing such information.

When hundreds or thousands of random sequences are tested by algorithm GRMT1, one can foresee that GR^* cannot be close to 1 before really calculating it, because hundreds or thousands of random sequences are highly unlikely to be linear to each other. But we can make use of algorithm GRMT to obtain heuristic information for clustering sequences.

Given a set of sequences $S = \{X_i \mid i = 1, 2,\ldots,K\}$,algorithm GRMTCA (General Regression Model Technique Clustering Algorithm) works as follows.

GRMTCA: Clustering of massive sequences

• Apply Algorithm GRMT1 to test whether the given sequences are linear to each other or not. If yes, all the sequences can go into one cluster and we can stop, otherwise, go to next step.

• After GRMT1, we have eigenvector $[e_1, e_2,\ldots. , e_K]t$. Create a feature value sequence $F =(\sigma(X_1)/e_1, \sigma(X_2)/e_2,\ldots. , \sigma(X_K)/eK)$ and sort it in increasing order. After sorting, suppose $F = (f_1, f_2,\ldots..,f_K)$.

• Start from the first feature value $f1$ in F. Suppose the corresponding sequence is Xi. We only check the linearity of Xi with the sequences whose feature values in F are close to $f1$. Here" close" means $fj/f1 \leq \xi$ (According to our experience, ξ =0.95 is enough).We collect those sequences which have linearity with Xi with confidence $\geq C$ into cluster $CM1$. Delete all the sequences in this cluster from set S, then repeat the similar procedure to obtain next cluster until S becomes empty. The most time-consuming part in GRMT1 and GRMTCA is to calculate the maximum eigen value and corresponding eigen vector of scatter matrix S. Fast algorithm [25, 26] can do so with high efficiency.

V. Conclusion

We propose GRMT1 by generalizing the Classical Regression Model. GRMT1 gives a measure GR^*, which is a new measure for linearity of multiple sequences. The meaning of GR^* for linearity is not relative. Based on GR^*, algorithm GRMT1can test the linearity of multiple sequences at a time and GRMTCA can cluster massive sequences with high accuracy as well as high efficiency.

References

[1] R. Agrawal, C. Faloutsos and A. Swami, *Efficient Similarity Search in Sequence Databases*, Proceedings of the 4th Intl. Conf. on Foundations of Data Organizations and Algorithms (FODO) (1993), pp. 69–84.

[2] B. Yi and C. Faloutsos, *Fast Time Sequence Indexingfor Arbitrary Lp Norms*, The 26th International Conference on Very Large Databases(VLDB) (2000), pp. 385–394.

[3] D. Rafiei and A. Mendelzon, *Efficient Retrieval of Similar Time Sequences Using DFT*, Proceedings of the 5th International Conference on Foundations of Data Organizations and Algorithms (FODO) (1998), pp. 69–84.

[4] R. Agrawal, K. I. Lin, H. S. Sawhne and K. Shim, *Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases*, Proc. of the 2Ist VLDB Conference(1995), pp. 490–501.

[5] T. Bozkaya, N. Yazdani and Z.M. Ozsoyoglu, Matchingand Indexing Sequences of Different Lengths, Proc. of the 6th International Conference on Information and Knowledge Management(1997), pp. 128–135.

[6] E. Keogh, A fast and robust method for pattern matching in sequences database, WUSS (1997).

[7] E. Keogh and P. Smyth. A Probabilistic Approach to Fast Pattern Matching in Sequences Databases, The 3rd Intl. Conf. on Knowledge Discovery and DataMining(1997), pp. 24–30.

[8] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast Subsequence Matching in Time-Series Databases, International Proceedings of the ACM SIGMOD Conference on management of Data(1994), pp. 419–429.

[9] C. Chung, S. Lee, S. Chun, D. Kim and J. Lee, Similarity Search for Multidimensional Data Sequences, Proceedings of the 16th International Conf. on Data Engineering(2000), pp. 599–608.

[10] D. Goldin and P. Kanellakis, On similarity queries for time-series data: constraint specification and implementation, The 1st International Conference on the Principles and practice of Constraint Programming (1995), pp. 137–153.

[11] C. Perng, H. Wang, S. Zhang and D. Parker, Landmarks: a New Model for Similarity-based Pattern Querying in Sequences Databases, Proc. of the 16th International Conference on Data Engineering(2000)

[12] H. Jagadish, A. Mendelzon and T. Milo, Similarity-Based Queries, The Symposium on Principles of Database Systems (1995), pp. 36–45.

[13] D. Rafiei and A. Mendelzon, Similarity-Based Queries for Sequences Data, Proc. of the ACM SIGMOD Conference on Management of Data(1997), pp. 13–25.

[14] C. Li, P. Yu and V. Castelli, Similarity Search Algorithm for Databases of Long Sequences, The 12th International Conference on Data Engineering (1996), pp. 546–553.

[15] G. Das, D. Gunopulos and H. Mannila, Finding similar sequences, The 1st European Symposium on Principles of Data Mining and Knowledge Discovery(1997),pp. 88–100.

[16] K. Chu and M. Wong, Fast Time-Series Searching with Scaling and Shifting, The 18th ACM Symp. On Principles of Database Systems (PODS 1999), pp. 237–248.

[17] B. Bollobas, G. Das, D. Gunopulos and H. Mannila, Time-Series Similarity Problems and Well-Separated Geometric Sets, The 13th Annual ACM Symposium on Computational Geometry (1997), pp. 454–456.

[18] D. Berndt and J. Clifford, Using Dynamic Time Warping to Find Patterns in Sequences, Working Notes of the Knowledge Discovery in Databases Workshop(1994), pp. 359–370.

[19] B. Yi, H. Jagadish and C. Faloutsos, Efficient Retrieval of Similar Time Sequences Under Time Warping, Proc. of the 14th International Conference on Data Engineering (1998), pp. 23–27.

[20] S. Park, W. Chu, J. Yoon and C. Hsu, Efficient Similarity Searches for Time-Warped Subsequences in Sequence Databases, Proc. of the 16th International Conf. on Data Engineering (2000).

[21] Z. Struzik and A. Siebes, The Haar Wavelet Transform in the Sequences Similarity Paradigm, PKDD (1999).

[22] K. Chan and W. FU. Efficient Sequences Matching by Wavelets, The 15th international Conf. on Data Engineering (1999).

[23] G. Das, K. Lin, H. Mannila, G. Renganathan and P. Smyt, Rule Discovery from Sequences, Knowledge Discovery and Data Mining(1998), pp. 16–22.

[24] G. Das, D. Gunopulos, Sequences Similarity Measures, KDD-2000: Sequences Tutorial.

[25] I. Dhillon, A New O(n2) Algorithm for the Symetric Tridiagonal Eigenvalue/Eigenvector Problem, Ph.D. Thesis. University of. California, Berkerley, 1997.

[26] R. Duda, P. Hart and D. Stork, Pattern Classification. 2nd Edition, John Wiley & Sons, 2000.

[27] J. Wooldridge, Introductory Econometrics: a modern approach, South-Western College Publishing, 1999.

[28] F. Mosteller and J. Tukey, Data Analysis and Regression: A Second Course in Statistics, Addison-Wesley, 1977.

[29] M.R. Anderberg. Cluster Analysis for Applications. Academic Press, New York, December 1973.

[30] J. Han, M.Kamber, and A.Tung. Spatial Clustering Methods in Data Mining: A review. In H.J. Miller and J.Han, editors, Geographic Data

Mining and Knowledge Discovery, pages 188-217. Taylor and Francis, London, December 2001.
[31] Gusfield D. Algorithms on Strings, Trees and Sequences. New York: Cambridge University Press, 1997.