# GRM: Generalized Regression Model for Clustering Linear Sequences

Hansheng Lei Venu Govindaraju *

**Abstract**

Linear relation is valuable in rule discovery of stocks, such as "if stock $X$ goes up 1, stock $Y$ will go down 3", etc. The traditional linear regression models the linear relation of two sequences perfectly. However, if user asks "please cluster the stocks in the NASDAQ market 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 model named GRM (Generalized Regression Model) to gracefully handle the problem of linear sequences clustering. GRM gives a measure, $GR^2$, to tell the degree of linearity of multiple sequences without having to compare each pair of them. Our experiments on the stocks in the NASDAQ market mined out many interesting clusters of linear stocks accurately and efficiently using the GRM clustering algorithm.

## 1 Introduction.

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. Let's classify them into 4 main categories:

**1.1 Lp norms [1, 2].** Given two sequences $X = [x_1, x_2, \cdots, x_N]$ and $Y = [y_1, y_2, \cdots, y_N]$, Lp norm is defined as $Lp(X,Y) = (\sum_{i=1}^{N} |x_i - y_i|^{\frac{1}{p}})$. 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 $X_1 = [1, 2, \cdots, 10]$ and $X_2 = [101, 102, \cdots, 110]$. $X_2$ is the result of shifting $X_1$ by 100, i.e., adding 100 to each element of $X_1$. The Lp distance between $X_1$ 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 $X_2 = \beta X_1$, where $\beta$ 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 $X_1$ is more similar to $X_2$ than to $X_3$. In conclusion, Lp norms as measure of (dis)similarity have two disadvantages:

- Cannot capture similarity in the case of shifting and scaling.
- Distance only has relative meaning of (dis)similarity.

It is well 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. A typical application of linearity is the rule discovery of stocks: cluster the stocks in the NASDAQ market into groups where sequences have strong linear relationship with each other.

**1.2 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.

**1.3 Time Warping [18, 19, 20].** It defines the distance between sequences $Xi = [x_1, x_2, \cdots, x_i]$ and $Yj = [y_1, y_2, \cdots, y_j]$ as $D(i,j) = |x_i - y_i| + min\{D(i -$

*Center for Unified Biometrics and Sensors, Computer Science and Engineering department, State University of New York at Buffalo, Amherst, NY 14260. Email: {hlei,govind@cse.buffalo.edu}

$1, j), D(i, j-1), D(i-1, j-1)\}$. 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.

**1.4 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 $\beta_0$ and $\beta_1$ so that $Distance(Y, \beta_0 + \beta_1 X)$ is minimized and this distance is below a given threshold. Paper [16] solved scaling factor $\beta_1$ and shifting offset $\beta_0$ 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.

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

The organization of this paper is as follows: Section §1 is introduction, section §2 provides a basic background of the traditional regression model. After that, section §3 describes GRM in detail and §4 shows examples of how to apply GRM to linearity measure and clustering of multiple sequences. Section §5 evaluates GRM using real stock prices in the NASDAQ market and discusses the experimental results. Finally, section §6 will draw conclusions.

## 2 Regression Model Background.

Linear regression analysis originated from statistics and has been widely used in econometrics [27, 28]. Let's use an example to introduce the basic idea of linear regression analysis. For an instance, to test the linear relation between consumption $Y$ and incoming $X$, we can establish the linear model as:

$$Y = \beta_0 + \beta_1 X + u \tag{2.1}$$

The variable $u$ is called the *error term*. The regression as (2.1) is termed as "*the regression of Y on X*". Given a set of sample data, $X = [x_1, x_2, \cdots, x_N]$ and $Y = [y_1, y_2, \cdots, y_N]$, $\beta_0$ and $\beta_1$ 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 $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$ as well as possible. We need to determine $\beta_0$ and $\beta_1$ such that $\sum_{i=1}^{N} u_i^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)$ is minimized, as shown in fig. 1 a). Using first order conditions[27, 28], we can solve $\beta_0$ and $\beta_1$ as follows :

$$\beta_0 = \overline{y} - \beta_1 \overline{x} \tag{2.2}$$

$$\beta_1 = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{N} (x_i - \overline{x})^2} \tag{2.3}$$

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

After obtaining $\beta_0$ and $\beta_0$, there will be a question: how do we measure how well the regression line fits these data? To answer this, the *R-squared* is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{i=1}^{N} (y_i - \overline{y})^2} \tag{2.4}$$

From (2.4) we can further derive:

$$R^2 = \frac{[\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})]^2}{\sum_{i=1}^{N} (x_i - \overline{x})^2 \sum_{i=1}^{N} (y_i - \overline{y})^2} \tag{2.5}$$

The value of $R^2$ is always between 0 and 1. The closer the value is to 1, the better the regression line fits the data points. $R^2$ is the measure for the *Goodness-of-Fit* in the traditional regression.

The regression model as (2.1) 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:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K + u \tag{2.6}$$

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

## 3 Generalized Regression Model.

**3.1 Why not the traditional Regression Model.** We observed that the Simple Regression Model is excellent in testing the linear relation of two sequences. $R^2$ is a good measure for linear relation. For an instance, $R^2(X_1, X_2) = 0.95$ 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 $R^2(X_1, X_2) > R^2(X_1, X_3)$ and say $X_1$ is similar to $X_2$ rather than $X_3$. Therefore, the meaning of $R^2$ for similarity is not relative, unlike distance-based measures. Furthermore, according to equation (2.5), we know that no matter sequence $X_2$ regresses on $X_1$ or vice versa, the $R^2$ is the same, i.e., $R^2$ is invariant to the regression order of sequences. This makes it feasible for $R^2$ to be a measure of similarity.

Fig. 2 shows two pairs of sequences and corresponding $R^2$ values. Sequence $X_1$ and $X_2$ are similar with

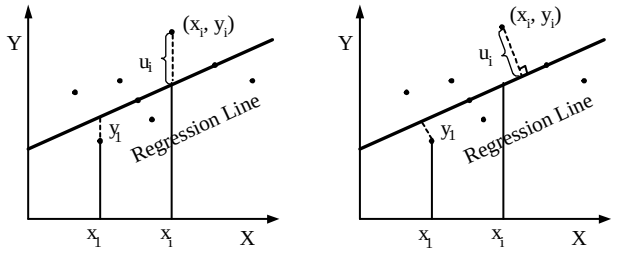



Figure 1: a) In the traditional Simple Regression Model, sequences $Y$ and $X$ compose a 2-dimensional space. $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$ are points in the space. The regression line is the line that fits these points in the sense of minimum-sum-of-squared-error. b) In GRM, two sequences also compose a 2-dimensional space. The error term $u_i$ is defined as the vertical distance from the point to the regression line.

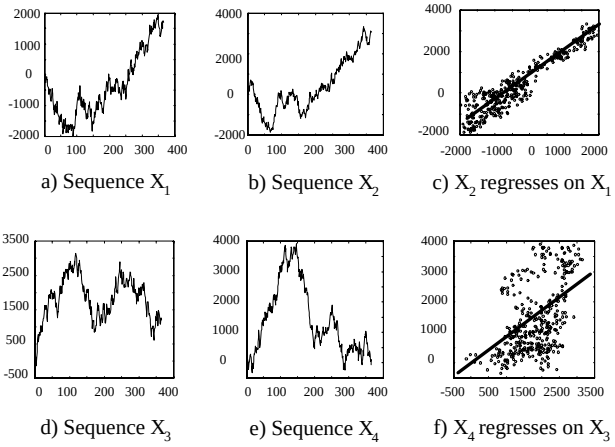



Figure 2: Two pairs of sequences. $R^2(X_1, X_2) = 0.91$ and $R^2(X_3, X_4) = 0.31$. The values of $R^2$ fit our observation that $X_1$ and $X_2$ are similar, while $X_3$ and $X_4$ are not similar.

$R^2(X_1, X_2) = 0.91$, while $X_3$ and $X_4$ are not similar since $R^2(X_3, X_4) = 0.31$. Points in fig. 2 c) are distributed along the regression line, while points in fig. 2 f) are scattered everywhere. 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 Simple 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, the measure. We cannot use $R^2$ 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 $X_1$, $X_2$, $\cdots$, $X_K$. Moreover, $R^2$ 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 $X_i = \beta_0 + \beta_1 X_1 + \cdots + \beta_i Y + \cdots + \beta_K X_K + u$. The $R^2$ here will be different from that of (2.6), because they have different meanings.

From a geometrical point of view, equation (2.6) 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.

Generalizing the idea of Simple Regression Model to multiple sequences, we propose the GRM (Generalized Regression Model).

**3.2 GRM: Generalized Regression Model.** Given $K(K \geq 2)$ sequences $X_1, X_2, \cdots, X_K$ and

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

We first organize them into $N$ points in the $K$-dimensional space:

$$p_1 = \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{K1} \end{pmatrix}, p_2 = \begin{pmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{K2} \end{pmatrix}, \cdots, p_N = \begin{pmatrix} x_{1N} \\ x_{2N} \\ \vdots \\ x_{KN} \end{pmatrix}$$

Then, we seek to find a line in the $K$-dimensional space that fits the $N$ points in the sense of minimum-sum-of-squared-error.

In the traditional regression, the error term is defined as:

$$u_i = y_i - (\beta_0 + \beta_1 x_{1i} + \cdots + \beta_K x_{Ki}) \tag{3.7}$$

It is the distance between $y_i$ and the regression hyper-plane in direction of axis $Y$. This makes sequence $Y$ unique from any $X_i (i = 1, 2, \cdots, K)$. Fig. 1 a) shows the $u_i$ of the traditional regression in the 2-dimensional space. In GRM, we define the error term $u_i$ as the *vertical* distance from point $(x_{1i}, x_{2i}, \cdots, x_{Ki})$ to the regression line. Please note that there is no $Y$ here anymore, because no sequence is special among its community. Fig. 1 b) shows the new defined $u_i$ in the case of two-dimensional space.

§7 Appendix gives the details of how to determine the regression line in $K$-dimensional space. Here, we assume we have found the regression line as follows:

$$\frac{p(1)-\overline{x_1}}{e_1}=\frac{p(2)-\overline{x_2}}{e_2}=\cdots=\frac{p(K)-\overline{x_K}}{e_K} \tag{3.8}$$

where $p(i)$ is the $i$-th element of point $p$, $[e_1,e_2,\cdots,e_K]^t$ is the eigenvector corresponding to the maximum eigenvalue of the scatter matrix(see §7 Appendix). $\overline{x_j}(j=1,2,\cdots,K)$ is the average of sequence $X_j$ (through out the rest of the paper, $\overline{x_j}$ always means the average of sequence $X_j$).

Expression (3.8) means that if any point $p$ in $K$-dimensional space satisfies equation (3.8), it must lie on the line. All the points that satisfy (3.8) compose a line in $K$-dimensional space. This line is the regression line. For the $N$ points $p_1,p_2,\cdots,p_N$, some may lie on the regression line, some may not. But the sum of squared-distance from $p_j(j=1,2,\cdots,N)$ to the regression line is minimized. To guarantee the regression line exists uniquely, we need following two assumptions:

- **Assumption 1.** $\sum_{i=1}^{N}(x_{ji}-\overline{x_j})\neq 0$. This assumption means no sequence is constant. It guarantees the scatter matrix has eigenvector.
- **Assumption 2.** There exists at least two points $p_i$, $p_j$ among the $N$ points such that $p_i\neq p_j$. This assumption guarantees the $N$ 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 GRM.

Similar to the traditional regression, after determining the regression line, we need a measure for Goodness-of-Fit. We define:

$$GR^2=1-\frac{\sum_{i=1}^{N}u_i^2}{\sum_{j=1}^{K}\sum_{i=1}^{N}(x_{ji}-\overline{x_j})^2} \tag{3.9}$$

If $GR^2$ is close to 1, then we know the regression line fits the $N$ points very well, which further means the $K$ sequences have a high degree of linear relationship with each other.

We need the following two lemmas before we prove an important theorem of GRM.

LEMMA 3.1. *Two sequences $X_1$ and $X_2$ are linear to each other $\Leftrightarrow$ Two sequences $X_1$ and $X_2$ have a linear relation as $\frac{x_{1i}-\overline{x_1}}{\sigma(X_1)}=\frac{x_{2i}-\overline{x_2}}{\sigma(X_2)}(i=1,2,\cdots,N)$, where $\sigma(X)$ denotes the standard deviation of sequence $X$.*

*Proof.* Two sequences $X_1$ and $X_2$ are linear to each other means:

$$x_{2i}=\beta_0+\beta_1 x_{1i}(i=1,2,\cdots,N) \tag{3.10}$$

So, we have:

$$\sum_{i=1}^{N}x_{2i}=N\beta_0+\sum_{i=1}^{N}x_{1i} \tag{3.11}$$

That is:

$$\overline{x_2}=\beta_0+\beta_1\overline{x_1} \tag{3.12}$$

Subtract (3.10) by (3.12), we get:

$$x_{2i}-\overline{x_2}=\beta_1(x_{1i}-\overline{x_1}) \tag{3.13}$$

On the other side, from (3.10) we know $\sigma(X_2)=\sigma(\beta_0+\beta_1X_1)=\beta_1\sigma(X_1)$, i.e., $\beta_1=\sigma(X_2)/\sigma(X_1)$. Thus, (3.13) can be rewritten as $\frac{x_{1i}-\overline{x_1}}{\sigma(X_1)}=\frac{x_{2i}-\overline{x_2}}{\sigma(X_2)}$. Δ

LEMMA 3.2. *$K$ sequences $X_1,X_2,\cdots,X_K$ are linear to each other $\Leftrightarrow$ $K$ sequences $X_1,X_2,\cdots,X_K$ have a linear relation as $\frac{x_{1i}-\overline{x_1}}{\sigma(X_1)}=\frac{x_{2i}-\overline{x_2}}{\sigma(X_2)}=\cdots=\frac{x_{Ki}-\overline{x_K}}{\sigma(X_K)}(i=1,2,\cdots,N)$, where $\sigma(X)$ denotes the standard deviation of sequence $X$.*

*Proof.* From lemma 3.1, this is obvious. Δ

Applying lemma 3.1 and 3.2, we can obtain the following theorem:

THEOREM 3.1. *$K$ sequences $X_1,X_2,\cdots,X_K$ are linear to each other $\Leftrightarrow$ Points $p_1,p_2,\cdots,p_N$ are all distributed on a line in $K$-dimensional space and this line is the regression line.*

*Proof.* From right to left is obvious. Let's prove from left to right.

According to lemma 3.2, if $X_1,X_2,\cdots,X_K$ are linear to each other, then this relation can be expressed as:

$$\frac{x_{1i}-\overline{x_1}}{\sigma(X_1)}=\frac{x_{2i}-\overline{x_2}}{\sigma(X_2)}=\cdots=\frac{x_{Ki}-\overline{x_K}}{\sigma(X_K)} \tag{3.14}$$

Recall that point $p_i=[x_{1i},x_{2i},\cdots,x_{Ki}]^t$. Let $p_i(k)=x_{ki}$, (3.14) can be rewritten as:

$$\frac{p_i(1)-\overline{x_1}}{\sigma(X_1)}=\frac{p_i(2)-\overline{x_2}}{\sigma(X_2)}=\cdots=\frac{p_i(K)-\overline{x_K}}{\sigma(X_K)} \tag{3.15}$$

Equation(3.15) means point $p_i(i=1,\cdots,N)$ is on the line. Therefore, we conclude that $p_1,p_2,\cdots,p_N$ are distributed on a line in $K$-dimensional space.

Now we need to show the line must be the regression line. The $N$ points $p_1,p_2,\cdots,p_N$ determine the line as (3.14) and it is unique according to assumption 2. On the other hand, our regression line guarantees that the sum of squared-error is minimized. If and only if the regression line is same as (3.14), the sum of squared-error is 0 (minimized). Δ

Now, let's derive some properties of $GR^2$:

1. $GR^2 = \frac{\lambda}{\sum_{i=1}^{N} \|p_i - m\|^2}$ and $1 \geq GR^2 \geq 0$.
2. $GR^2 = 1$ means the $K$ sequences have exact linear relationship to each other.
3. $GR^2$ is invariant to the order of $X_1, X_2, \cdots, X_K$, i.e., we can arbitrarily change the order of the $K$ sequences and the value of $GR^2$ does not change.

*Proof.* 1. According to §7 Appendix, we have:

$$\sum_{i=1}^{N} u_i^2 = -e^t S e + \sum_{i=1}^{N} \| p_i - m \|^2 \quad (3.16)$$

Thus,

$$\begin{aligned} GR^2 &= 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{j=1}^{K} \sum_{i=1}^{N} (x_{ji} - \overline{x_j})^2} \\ &= 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{i=1}^{N} \| p_i - m \|^2} \\ &= \frac{e^t S e}{\sum_{i=1}^{N} \| p_i - m \|^2} (\text{recall } (3.16)) \\ &= \frac{e^t \lambda e}{\sum_{i=1}^{N} \| p_i - m \|^2} \\ &= \frac{\lambda}{\sum_{i=1}^{N} \| p_i - m \|^2} (\text{recall that } \| e \|^2 = 1) \end{aligned}$$

From (3.16), we know $\sum_{i=1}^{N} u_i^2 = -e^t S e + \sum_{i=1}^{N} \| p_i - m \|^2 \geq 0$, therefore:

$$\sum_{i=1}^{N} \| p_i - m \|^2 \geq e^t S e = \lambda \geq 0 \quad (3.17)$$

so we conclude $1 \geq GR^2 \geq 0$.

2. If $GR^2 = 1$, then $\sum_{i=1}^{N} u_i^2 = 0$, which means the regression line fits the $N$ points perfectly. Therefore, according to theorem 3.1, the $K$ sequences have exact linear relationship to each other.
3. This is obvious. Δ

Because $GR^2$ has above important properties, we define $GR^2$ as the degree of linear relation and the similarity measure in the sense of linear relation.

## 4 Applications of GRM.

The procedure of applying GRM to measure the linear relation of multiple sequences is described by algorithm GRM.1.

**GRM.1: Testing linearity of multiple sequences**

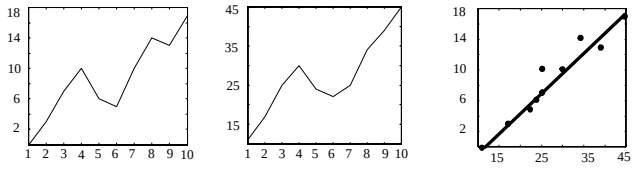



Figure 3: An example of applying GRM.1 to two sequences.

- Organize the given $K$ sequences with length $N$ into $N$ points $p_1, p_2, \cdots, p_K$ in $K$-dimensional space as shown in §3.2.
- Determine the regression line. First, calculate the average $m = \frac{1}{N} \sum_{i=1}^{N} p_i$. Second, calculate the scatter matrix $S = \sum_{i=1}^{N} (p_i - m)(p_i - m)^t$. Then, determine the maximum eigenvalue $\lambda$ and corresponding eigenvector $e$ of $S$.
- Calculate $GR^2$ according to property 1 of $GR^2$.
- Draw conclusion. Suppose we only accept linearity with confidence no less than $C$ (say, $C = 85\%$). If $GR^2 \geq C$, we can conclude that the $K$ sequences are linear to each other with confidence $GR^2$ and the linear relation is as (3.8).

### 4.1 Apply GRM.1 to two sequences: an example.

Suppose we want to test two sequences $X_1$ and $X_2$ and

$$X_1 = [\ 0,\ 3,\ 7,\ 10,\ 6,\ 5,\ 10,\ 14,\ 13,\ 17];$$

$$X_2 = [11,\ 17,\ 25,\ 30,\ 24,\ 22,\ 25,\ 34,\ 39,\ 45],$$

as shown in fig. 3 a) and b) respectively. Let's apply GRM.1 step by step.

First, organize the two sequences into 10 points: (0, 11), (3, 17), (7, 25), (10, 30), (6, 24), (5, 22), (10, 25), (14, 34), (13, 39), (17, 45). If we draw the 10 points in the $X_1$-$X_2$ space, their distribution is as shown in fig. 3 c).

Second, determine the regression line. Average $m = \frac{1}{N} \sum_{i=1}^{N} p_i = [8.5, 27.2]^t$. Maximum eigenvalue $\lambda = 1161.9$ and corresponding eigenvector $e = [0.4553, 0.8904]^t$.

Third, calculate $GR^2 = 0.9896$. Since $GR^2$ is as high as 98.96%, we can conclude that $X_1$ is highly linear related to $X^2$. In addition, we find their linear relation is $\frac{X_1 - 8.5}{0.4553} = \frac{X_1 - 27.2}{0.8904}$.

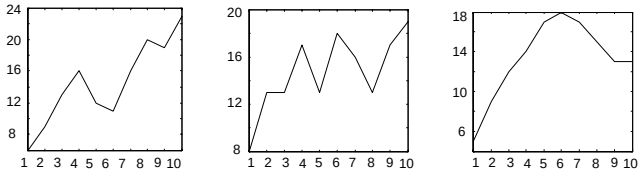



Figure 4: Three sequences with low similarity.

If we observe fig. 3 a) and b), we can see they are really similar (linear) to each other. So the conclusion based on $GR^2$ makes sense.

### 4.2 Apply GRM.1 to multiple sequences: an example.

GRM is intended to test whether multiple sequences are linear to each other or not. Let's give an example for testing 3 sequences at a time.

Suppose we have three sequences:

$$X_1 = [6,\ 9,\ 13,\ 16,\ 12,\ 11,\ 16,\ 20,\ 19,\ 23];$$

$$X_2 = [8,\ 13,\ 13,\ 17,\ 13,\ 18,\ 16,\ 13,\ 17,\ 19];$$

$$X_3 = [5,\ 9,\ 12,\ 14,\ 17,\ 18,\ 17,\ 15,\ 13,\ 13].$$

They are shown in fig. 4 a), b) and c) respectively. Following the same procedure, we can calculate $GR^2 = 0.7301$. This confidence is not much high, thus we can conclude that some sequences are not very linear to others. If we observe the three sequences in fig. 4, we can see that none of the three sequences are linear to one other. This example demonstrates that $GR^2$ is a good measure again. Actually, we have tested many sequences and found $GR^2$ as linearity measure agrees with our subjective observation.

### 4.3 Apply GRM to clustering of massive sequences.

When hundreds or thousands of random sequences are tested by algorithm GRM.1, one can foresee that $GR^2$ 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. If we know the result before carrying out the test, what is the use of GRM.1 in testing massive sequences? Fortunately, we can make use of algorithm GRM.1 to obtain heuristic information for clustering sequences.

From theorem 3.1, lemma 3.2 and the formula of the regression line as (3.8), we know that if the multiple sequences have linear relationship to each other, then the eigenvector is directly related to the standard deviations, i.e., $\frac{\sigma(X_1)}{e_1} = \frac{\sigma(X_2)}{e_2} = \cdots = \frac{\sigma(X_K)}{e_K}$. To infer reversely, $\frac{\sigma(X_i)}{e_i} \approx \frac{\sigma(X_j)}{e_j}$ is a strong hint that they *may* be linear; if $\frac{\sigma(X_i)}{e_i}$ differs greatly from $\frac{\sigma(X_j)}{e_j}$ , they can not be linear. This is very valuable for clustering. We call $\frac{\sigma(X_i)}{e_i}$ the feature value of sequence $X_i$.

Based on this, we derive an algorithm for clustering massive sequences. Given a set of sequences $S = \{X_i \mid i = 1, 2, \cdots, K\}$, algorithm GRM.2 works as follows.

**Algorithm GRM.2: Clustering of massive sequences**

- Apply Algorithm GRM.1 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 GRM.1, we have eigenvector $[e_1, e_2, \cdots, e_K]^t$. Create a feature value sequence $F = (\frac{\sigma(X_1)}{e_1}, \frac{\sigma(X_2)}{e_2}, \cdots, \frac{\sigma(X_K)}{e_K})$ and sort it in increasing order. After sorting, suppose $F = (f_1, f_2, \cdots, f_K)$.
- Start from the first feature value $f_1$ in $F$. Suppose the corresponding sequence is $X_i$. We only check the linearity of $X_i$ with the sequences whose feature values in $F$ are close to $f_1$. Here "close" means $f_j/f_1 \leq \xi$ (According to our experience, $\xi =$ is enough). We collect those sequences which have linearity with $X_i$ with confidence $\geq C$ into cluster $CM_1$. 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 GRM.1 and GRM.2 is to calculate the maximum eigenvalue and corresponding eigenvector of scatter matrix $S$. Fast algorithm [25] can do so with high efficiency.

## 5 Experiments.

Our experiments focus on two aspects: 1) Is the $GR^2$ really a good measure for linearity? If two or multiple sequences have high degree of linearity with each other, will $GR^2$ really be close to 1 or vice versa? 2) Can GRM.2 be used to mine linear stocks in the NASDAQ market? What is the accuracy and performance of GRM.2 in clustering sequences?

For the first concern, we need to test algorithm GRM.1. We generated 5000 random sequences of real number for experiments. Each sequence $X = [x_1, x_2, \cdots, x_N]$ is a random walk: $x_{t+1} = x_t + z_t, t = 1, 2, \cdots$, where $z_t$ is an independent, identically distributed (IID) random variable. Our experiments conducted on these sequences show that $GR^2$ is a good measure for linearity. Generally speaking, if $GR^2 \geq 0.90$,

the sequences are very linear to each other; if $0.80 \leq GR^2 \leq 9.0$, the sequences have high degree of linearity, while $GR^2 < 0.80$ means the linearity is low.

For the second concern, we need to test GRM.2. Experiments were conducted on the real stocks in the NASDAQ market. We will discuss our results in more detail here.

### 5.1 Experiments setup.

We downloaded all the NASDAQ stock prices from yahoo website ( http://table.finance.yahoo.com/) starting from 05-08-2001 and ending at 05-08-2003. It has over 4000 stocks. We used the daily closing prices of each stock as sequences. These sequences are not of the same length for various reasons, such as cash dividend, stock split, etc. We made each sequence length 365 by truncating long sequences and removing short sequences. Finally, we have 3140 sequences all with the same length 365. The 365 daily closing prices start from 05-08-2001, ending at some date (not necessarily at 05-08-2003, since there are no prices in weekends).

Our task is to mine out clusters of stocks with similar trends. To solve this problem, there are two choices basically. The first is GRM.2 and the second is to use the Simple Regression Model and brute-forcedly calculate linearity of each pair. For convenience, we name this method BFR (Brute-Force $R^2$). As we discussed in §2, the Simple Regression Model can also compare two sequences without worrying about scaling and shifting, because $R^2$ is invariable to both scaling and shifting. Intuitively, BFR is more accurate but slower, while GRM.2 is faster at the compensation of some loss of accuracy. Our experiments will compare the accuracy and performance of BFR and GRM.2. We do not have to compare GRM.2 with other methods in the field of sequence analysis, such as [16] - [20], because no method so far can test multiple sequences at a time and most of them are sensitive to scaling and shifting, to the best of our knowledge.

In experiments, we require the confidence of linearity of sequences in the same cluster to be no less than 85%. Recall that the linearity of is defined by $GR^2$ in GRM and $R^2$ in the Simple Regression Model.

Given a set of sequences $S = \{Xi \mid i = 1, 2, \cdots, K\}$, BFR works as follows:

1) Take an arbitrary sequence out from $S$, say $X_i$. Find all sequences from $S$ which has $R^2 \geq 85\%$ with $X_i$. Collect them into a temporary set. Do *post-selection* to make sure all sequences have confidence of linearity $\geq 85\%$ with each other by deleting some sequences from the set if necessary.

2) Save this temporary set as a cluster and delete all sequences in this set from S.

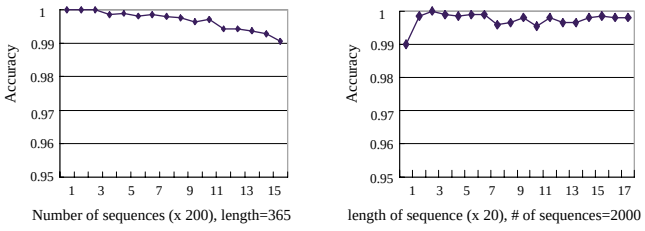



Figure 5: a) Relative accuracy of GRM.2 when the number of sequences varies. b) Relative accuracy of GRM.2 when the length of sequences varies.

3) Repeat 1) until $S$ become empty.

After we find clusters $CM_1, CM_2, \cdots, CM_l$ using GRM.2 and clusters $CR_1, CR_2, \cdots, CR_h$ using BFR, we can calculate the accuracy of GRM.2. The accuracy of the BFR is considered as 1 since it is brute-force. We calculate the relative accuracy of GRM.2 as follows.

$$Accuracy(CM, CR) = [\sum_i max_j(Sim(CM_i, CR_j))]/l,$$

where $Sim(CM_i, CR_j) = 2\frac{|CM_i \cap CR_j|}{|CM_i| + |CR_j|}$.

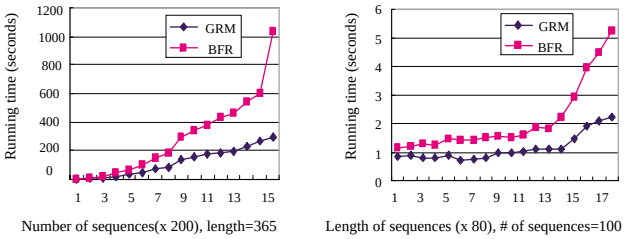
Our programs were written using MATLAB 6.0. Experiments were carried out in a PC with 800MHz CPU and 384MB of RAM.

### 5.2 Experimental results.

Fig. 5 a) shows the accuracy of GRM.2 relative to BFR while the number of sequences varies when the length of sequence is fixed at 365. The accuracy remains at above 0.99 (99%) when the number of sequences increases to over 3000. Fig. 5 b) shows that the relative accuracy of GRM.2 stays above 0.99 when the length of sequences varies with the number of sequences fixed at 2000.

Fig. 6 a) and b) show the running time of GRM.2 and BFR. We can see that GRM.2 is much faster than BFR, no matter when the number of sequences varies while holding the length fixed or vice versa. The faster speed is at the compensation of less than 1% accuracy of clustering. Note that any clustering algorithm cannot be 100% correct because the classification of some points are ambiguous. From this point of view, we can conclude that GRM.2 is better than BFR.

Fig. 7 shows a cluster of stocks we found out using GRM.2. The stocks of four companies, A (Agile Software Corporation), M (Microsoft Corporation), P (Parametric Technology Corporation) and T (TMP Worldwide Inc.), are of very similar trends. In addition to the clustering, we found they have following approximate linear relation: $\frac{A-10.96}{0.0138} = \frac{M-29.36}{0.0140} = \frac{P-6.37}{0.0115} =$




a) b)

Figure 6: Performance comparison between GRM.2 and BFR. a) Running time comparison when the number of sequences varies. b) Running time comparison when the length of sequences varies.

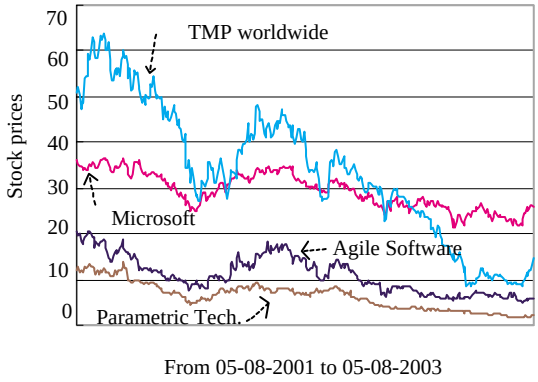



Figure 7: A cluster of stocks mined out by GRM.2.

$\frac{T-33.62}{0.0563}$

From the stocks shown in fig. 7, we can see that A, M and P fit each other very well with some shifting. T fits others with both scaling and shifting. Above relation is consistent with this our observation, since 0.0138, 0.014, 0.0115 is close to each other, which means the scaling factor between each pair of M, P and T is close to 1, while 0.0563 is about 4 times of 0.014, which means the scaling factor between T and M is about 4. Fig. 8 shows the regression of Microsoft stock on Agile Software stock. The distribution of the points in the 2-dimensional space confirms that the two stocks have strong linear relation.

Actually, we found many interesting clusters of stocks. In each cluster, every stock has similar trend to one another. Due to space limitation, we cannot show all of them here. The results of clustering are valuable for stocks buyers and sellers.

## 6 Conclusion.

We propose GRM by generalizing the traditional Simple Regression Model. GRM gives a measure $GR^2$, which is a new measure for linearity of multiple sequences. The meaning of $GR^2$ for linearity is not relative. Based on $GR^2$, algorithm GRM.1 can test the linearity of multiple sequences at a time and GRM.2 can cluster massive sequences with high accuracy as well as high efficiency.

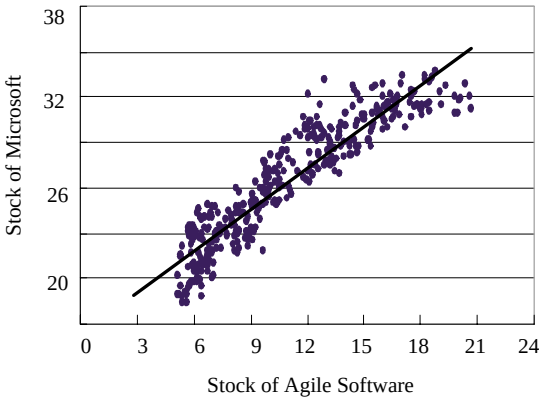



Figure 8: The regression of Microsoft stock on Agile Software stock.

**Acknowledgements.** The authors appreciate Dr. Jian Pei and Dr. Eamonn Keogh for their invaluable suggestions and advice on this paper. The authors also thank Mr. Mark R. Marino and Mr. Yu Cao for their comments.

## 7 Appendix

**Determine the Regression Line in $K$-dimensional space**

Suppose $p_1, p_2, \cdots, p_N$ are the $N$ points in $K$-dimensional space. First we can assume the regression line is $l$, which can be expressed as:

$$l = m + ae, \tag{7.18}$$

where $m$ is a point in the $K$-dimensiona space, $a$ is an arbitrary scalar, and $e$ is a unit vector in the direction of $l$.

When we project points $p_1, p_2, \cdots, p_N$ to line $l$, we have point $m+a_ie$ corresponds to $p_i (i = 1, \cdots, N)$. The squared error for point $p_i$ is:

$$u_i^2 = \| (m + a_ie) - p_i \|^2 \tag{7.19}$$

Thus, the sum of all the squared-error is to be:

$$\sum_{i=1}^{N} u_i^2 = \sum_{i=1}^{N} \| (m + a_ie) - p_i \|^2$$
$$= \sum_{i=1}^{N} \| a_ie - (p_i - m) \|^2$$

$$= \sum_{i=1}^{N} a_i^2 \parallel e \parallel^2 -2\sum_{i=1}^{N} a_i e^t(p_i - m) + \sum_{i=1}^{N} \parallel p_i - m \parallel^2$$

$$= \sum_{i=1}^{N} a_i^2 - 2\sum_{i=1}^{N} a_i e^t(p_i - m) + \sum_{i=1}^{N} \parallel p_i - m \parallel^2$$

Since $e$ is unit vector, $\parallel e \parallel^2 = 1$.

In GRM, the sum of squared error must be minimized. Note that $\sum_{i=1}^{N} u_i^2$ is a function of $m$, $a_i$ and $e$. Partially differentiating it with respect to $a_i$ and setting the derivative to be zero, we can obtain:

$$a_i = e^t(p_i - m) \tag{7.20}$$

Now, we should determine vector $e$ to minimize $\sum_{i=1}^{N} u_i^2$. Substituting (7.20) to it, we have:

$$\begin{aligned}
\sum_{i=1}^{N} u_i^2 &= \sum_{i=1}^{N} a_i^2 - 2\sum_{i=1}^{N} a_i a_i + \sum_{i=1}^{N} \parallel p_i - m \parallel^2 \\
&= -\sum_{i=1}^{N} a_i^2 + \sum_{i=1}^{N} \parallel p_i - m \parallel^2 \\
&= -\sum_{i=1}^{N} [e^t(p_i - m)]^2 + \sum_{i=1}^{N} \parallel p_i - m \parallel^2 \\
&= -\sum_{i=1}^{N} [e^t(p_i - m)(p_i - m)^t e] + \sum_{i=1}^{N} \parallel p_i - m \parallel^2 \\
&= -e^t S e + \sum_{i=1}^{N} \parallel p_i - m \parallel^2,
\end{aligned}$$

where $S = \sum_{i=1}^{N} (p_i - m)(p_i - m)^t$, called *scatter matrix*[26].

Obviously, the vector $e$ that minimizes above equation also maximizes $e^t S e$. We can use Lagrange multipliers to maximize $e^t S e$ subject to the constraint $\parallel e \parallel^2 = 1$. Let:

$$\mu = e^t S e - \lambda(e^t e - 1) \tag{7.21}$$

Differentiating $\mu$ with respect to $e$, we have:

$$\frac{\partial \mu}{\partial e} = 2Se - 2\lambda e \tag{7.22}$$

Therefore, to maximize $e^t S e$, $e$ must be the eigenvector of the scatter matrix $S$:

$$Se = \lambda e \tag{7.23}$$

Furthermore, note that:

$$e^t S e = \lambda e^t e = \lambda \tag{7.24}$$

Since $S$ generally has more than one eigenvectors, we should select the eigenvector $e$ which corresponds to the largest eigenvalue $\lambda$.

Finally, we need $m$ to complete the solution. $\sum_{i=1}^{N} \parallel p_i - m \parallel^2$ should be minimized since it is always non-negative. To minimize it, $m$ must be the average of $p_1, p_2, \cdots, p_N$. It is not difficult to prove this. Readers are referred to [26] for proof.

With $m$ as the average of the $N$ points and $e$ from (7.23), the regression line $l$ is determined. The line in form of (7.18) is not easy to understand. Let's write it in an easier way. Suppose $e = [e_1, e_2, \cdots, e_K]^t$ and $m = [\overline{x_1}, \overline{x_2}, \cdots, \overline{x_K}]^t$, $l$ can be expressed as: $\frac{p(1)-\overline{x_1}}{e_1} = \frac{p(2)-\overline{x_2}}{e_2} = \cdots = \frac{p(K)-\overline{x_K}}{e_K}$, where $p(i)$ is the $i$-th element of point $p$.

## References


[1] R. Agrawal, C. Faloutsos and A. Swami, *Efficient Similarity Search in Sequence Databases*, Proc.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 Indexing for Arbitrary Lp Norms*, The 26th Intl. Conf. on Very Large Databases(VLDB) (2000), pp. 385–394.

[3] D. Rafiei and A. Mendelzon, *Efficient Retrieval of Similar Time Sequences Using DFT*, Proc. of the 5th Intl. Conf. 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, *Matching and 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 Data Mining(1997), pp. 24–30.

[8] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, *Fast Subsequence Matching in Time-Series Databases*, In Proc. 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*, Proc. 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 Intl.Conf. 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 Conf. 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 Intl. Conf. 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 $O(n^2)$ Algorithm for the Symetric Tridiagonal igenvalue/Eigenvector Problem*, Ph.D. Thesis. Univ. of. Calif., 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.