A Polygon Description Based Similarity Measurement of Stock Market Behavior

Por-Shen Lai, Student Member, IEEE, and Hsin-Chia Fu, Member, IEEE

Abstract—This paper proposes (1) a polygon distribution descriptor and (2) an EC-based similarity measurement for stock market behavior analysis. After learning stock market historical data, a polygon descriptor can capture the dependencies among stock market quantities, such as stock prices, volumes, EPS (earn per share) and so on. By applying the EC-based similarity measurement on polygon descriptors which were trained by stock market data during different periods, the similarity of corresponding stock market behavior can be analyzed. To demonstrate the representation capabilities of the proposed polygon descriptor, Taiwan stock market data from 1986 to 2006 are used. Experimental results show that the polygon descriptor captures the dependencies of stock market quantities, and the similarity measurement shows that the proposed methods capture the changes of market behavior as expected.

Index Terms—Similarity Measurement, Polygon, Data Distribution, Stock Market Behavior, Data Mining, Deforming Path, Evolutionary Computing.

I. INTRODUCTION

Investment activities are risky beyond doubt. For most investors, they are usually asking - is it the right time to buy? are there more stocks having similar behavior? Generally, these questions are answered depending on some theoretic rules and investors’ experiments. That is, investors investigate financial archives to find similar patterns as references to help them make decisions. There are so many records in financial archives that an automated search utility is necessary. However, as a prerequisite, the behavior of stock market has to be described by a numerical form. Besides, an appropriate similarity measurement is essential to search similar patterns in financial archives automatically.

Numerous researches[1] are proposed to model the dynamics beneath a single financial series. These researches estimate the dependencies between past and future data with different assumptions and characteristics, like linearity, mutual information, and so on. Instead of using the estimated instantaneous dynamics of system as the descriptor of stock market behavior, the estimated dynamics are used to do short term predictions. That is, they mainly focus on the time dependency of input series. Terasvirta et al.[2] have a re-examination on this type of methods. Without considering more input data and knowledges, It is very difficult for them to contribute the long-period investments.

On the other hand, several researches have been proposed to estimate the probability of business-cycle. Gregoir and Lengfart[3] use business survey data and HMM model to find the turning points at business cycle. They found that these multivariate data is useful for their applications. Bellone and Gautier[4] also used Gregoir’s model in a set of four financial series to supply an unrevised and reliable advanced qualitative probabilistic indicators. However, since these model are proposed to capture the dependencies between input and output as detailed as possible, they are so complex that reasoning the learned parameters is quiet difficult. Based on complicated computations with a great deal of parameters, you can get corresponding outputs by entering specific inputs. You receive a probability value but don’t know why. Without acquiring more explainable information, it’s very difficult for investors to have their knowledges and estimated results cooperated.

In the proposed paper, I’m targeting to design an automated method which computes the similarities among record sets in financial archives to help investors mining investment references. A stock market behavior descriptor which extracts and characterizes the dependencies among stock market quantities from data distribution are introduced. Since the data distribution reflects how stock market quantities interact with each other, the shape of data distribution is chosen as the descriptor of stock market behavior. Figure 1 is an example of the data distribution. Apparently, the very different data distribution shows that the behavior in 1996 and 2006 is different.

Besides, an EC (evolutionary-computation)-based comparison method is proposed to measure the difference between polygon descriptors. The difference between polygon descriptors is measured by the proposed minimal deforming distance. Based on a set of pre-defined operators, the minimal deforming distance is the minimal cost to transform a polygon descriptor to another. Based on the minimal deforming distance, the similarity of stock market behavior can be measured.

Section II briefly the polygon descriptor which describes the observation data distribution and the definition of deforming distance which is the distance between polygon descriptors. Section III introduces the EC-based method which calculates the deforming distance. The evaluations of the proposed methods are shown in section IV. Finally, the conclusions are given in section V.

II. POLYGON DESCRIPTOR FOR DATA MODELING

In order to indicate the state of a dynamic system, a polygon descriptor is proposed to extract and characterize the dependencies among variates. That is, the similarity between two polygon descriptors reflects the dependency changes between data distributions. Section II-A briefs the method which models the data distribution by polygons. The deforming distance which measures the distance between polygon descriptors is defined in Section II-B and discussed in Section II-C.
A. Polygon Descriptor

In order to measure the similarity between two stock market data sets, appropriate discriminative feature components needed to be identified first. Since the stock market is usually fluctuant and the observed market data values are in a random and/or noisy manner, the feature has to be shift, scale, and rotate invariant. Modeling or approximating data distribution in linear or non-linear functions is difficult to represent the randomness of data. On the other hand, modeling data in complex mixture model can reserve most of characteristics of the data distribution. However, these models lack operational flexibility, such that shift, scale, and rotate invariant operations are very difficult to be included in these models. In the past, shape analysis\cite{5,6} methods have been proposed for shift, scale, and rotate invariant similarity measurement between images. However, most of these methods need obvious edges or boundaries to decide the shape of objects. Since the requirement of clear edges is too restricted for modeling economic data distribution, shape analysis method seems inappropriate to stock market application. Therefore, a polygon descriptor is proposed to represent the random and noisy stock market data distributions in a shift, scale, and rotate invariant form.

A polygon descriptor unions several convex polygons to represent a data distribution. A convex polygon can be represented by a reference point inside of the polygon and $N$ axes which show the normal direction of edges and the distance from reference point to each edges. Figure 2 shows an example polygon for representation of a 2D data distribution. This polygon example contains a reference point at $\langle 2, 2 \rangle$, and five axes which are the perpendiculars of edges, are represented in vector forms, $(0, 1)$, $(0.89, 0.45)$, $(0.71, -0.71)$, $(-0.71, -0.71)$, and $(-0.89, 0.45)$.

Based on the statistical characteristics of a data distribution, the polygon descriptor can be learned by an iterative process. The reference point $C$ is defined by Eq. 1 as the medoid of data points $P$.

$$C = \arg\min_{P_i \in P} \left( \sum_{P_j \in P} \text{dist}(P_i, P_j) \right), \quad (1)$$

where $P_i$ and $P_j$ are two data points in $P$ and $\text{dist}(P_i, P_j)$ is the distance between $P_i$ and $P_j$.

By using the reference point $C$ and randomly initiated axes $(A_i, i = 1, ..., N)$, the data points $P$ are clustered according to Eq. 2.

$$W = \arg\max_{W_j} A_j \cdot (P_i - C), \quad (2)$$

where $A_j$ is the axis of cluster $W_j$ and $j = 1, ..., N$.

Then each axis $A_j$ is refined by the data points in cluster $W_j$. Let $D$ be the dimension of data points, and divide the cluster into $D$ sub-clusters by hyper-planes passing through $C$. Then, compute the mean points of each sub-clusters, and generate the hyperplane that passing through all these $D$ mean points. Let the new direction of axis orthogonal to the hyperplane and the axes length equal to the distance from $C$ to the hyperplane. The clustering and refining process repeat until the axes converge.

The number of axes are decided during the learning process. Since the axes are orthogonal to the edges of distribution, axes which share the same boundary merge automatically. That is, by assigning excessed number of axes, extra axis merges to another axis automatically during the learning process.

B. Deforming on Polygons

Deforming distance is proposed to describe the difference between polygon descriptors. Based on the definition of a set
of operators, a deforming distance is the minimal total cost of the operators that transform one polygon descriptor to the other.

Let \( a_i \) be the angle between the \( i \)-th and \( i+1 \)-th axis of a polygon descriptor. A \( n \)-tuple value \( (a_0, a_1, ..., a_{n-1}) \) are used to represent the angle of all peaks. As shown as Figure 3, five deforming operators are defined as follows,

1) Flattening: \( \text{delete}(i) \) - remove \( a_i \) from list.
2) Raising: \( \text{insert}(i) \) - insert an angle into the list after \( a_i \).
3) Sharpening: \( \text{sharp}(i, \delta) \) - increase \( a_i \) by \( \delta \) and decrease \( a_{i+1} \) and \( a_{i+2} \) by \( \delta/2 \).
4) Rotating: \( \text{rotate}(\delta) \) - rotate the polygon by \( \delta \).
5) Extending: \( \text{extend}(i, \delta) \) - increase the \( i \)-th axis by \( \delta \).

![Fig. 3. The operators for deforming. 1) Flattening - remove an angle; 2) Raising - insert an angle; 3) Sharpening - modify an angle; 4) Rotating - rotate the descriptor; 5) Extending - stretch an axis.](image)

Figure 4 shows an example of such deforming process. A triangle is transformed into a diamond shape by one raising, one rotating, and two sharpening.

To simplify the process of minimal deforming distance estimation, a polygon descriptor is represented as a sequence of angles (the angle between neighboring axes) and several restrictions are applied to the deforming process.

1) The extending operator is applied at last.
2) The raising operator only inserts zero degree angle and the flattening operator only delete zero degree angles.
3) The raising operator should be applied before sharpening operator.
4) The flattening operator should be applied after sharpening operator.
5) The total cost is the total changed degrees of all operators.

C. Estimating the Deforming Distance

Searching the optimal path is related to many well-known algorithms, such as shortest-path[7][8], traveling salesman problem[9], string-to-string correction problems[10][11], and so on. In this section, the method that reduce the deforming distance estimation problem to a string-to-string correction like problem is introduced. By dividing the solution space into several sub-spaces that the minimal deforming distance in a sub-space can be calculated directly, the global minimal deforming distance can be estimated like a string-to-string correction problems.

The calculation of the minimal deforming distance for each sub-space is introduced in this section and the string-to-string correction problem is discussed in section III.

Let \( S = (s_0, ..., s_{n-1}) \) and \( T = (t_0, ..., t_{m-1}) \) represent the angle sequences of source and target polygon descriptor respectively, where \( \sum_{i=0}^{n-1} s_i = \sum_{j=0}^{m-1} t_j \). Assume a match assignment \( M = \{(l'_0, l'_0), ..., (l'_{k-1}, l'_{k-1})\} \), where \( 0 \leq l' \leq n-1, 0 \leq t' \leq m-1, \) and \( l' \) and \( t' \) are the label of the specified angle in \( S \) and \( T \) respectively, is given (See Section III). An element \((l'^i, t'^i)\) in a match assignment means that the angle \( l'^i \) is corresponding to \( s_{l'^i} \). Since the cost of raising, flattening, rotating, and extending are determined when a match assignment is given, the optimal solution in the corresponding sub-space is only involved with the sharpening operators. Based on a given match assignment \( M \), the shift amounts \( (D_0, ..., D_n) \) of every angles to convert from \( S \) to \( T \) can be estimated as follows.

1) Depending on \( S \) and \( T \), create a map like Figure 5.
   a) The table is partitioned into \( n \) rows and the height of \( (i+1) \)-th rows is \( s_i \).
   b) The table is partitioned into \( m \times 2 \) columns and the width of \( (i+1) \)-th column is \( t_i \).
   c) Put labels at the left-top cell corner of the \( [l'_i + n - l'_i % n + l'_i + 1] \)-th column and \( [l'_i + 1] \)-th rows for \( i = 1, ..., k \).
   d) Draw a line with slope equal to \(-1\) that the sum of their vertical distance from all labels to the line equal to zero.

2) If \( l'_i = l'_{i+1} \), a zero angle is inserted after the \( (i+1) \)-th angle. Let \( S' = (s_{l'_1}, ..., s_{n'}) \) represent the modified source angle sequence.
3) If \( l'_i = l'_{i+1} \), a zero angle is inserted after the \( (i+1) \)-th angle. Let \( T' = (t_{l'_1}, ..., t_{m'-1}) \) represent the modified target angle sequence.
4) Let \( D_{(i+n-1)%n} \) be the vertical distance from the \( (i+1) \)-th label to the line. If the label is above the line, let \( D \) be negative.

Using the shift amounts \( D \), the degree \( \delta \) of sharpening operator for each angle can be calculated. Let \( \delta_i \) be the change amount for the sharpening operator at the \( i+1 \)-th angle. Based on the definition of sharpening operator, the following relation holds

\[
D_i = \delta_i - \delta_{(i+1)\%n},
\]

where \( i \in \{0, 1, 2, ..., n-1\} \).

Using the shift amount \( D \) generated from the map, the \( \delta \)
A string-to-string correction problem can be solved by the following linear equations.

\[
\begin{bmatrix}
1 & -1 & 0 & \ldots & 0 & 0 \\
0 & 1 & -1 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & -1 \\
-1 & 0 & 0 & \ldots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\delta_0 \\
\delta_1 \\
\vdots \\
\delta_n-1
\end{bmatrix}
= \begin{bmatrix}
D_0 \\
D_1 \\
\vdots \\
D_n-2
\end{bmatrix}
\]

Assume \(\delta_0 = \alpha\), then

\[
\delta_{n-1} = D_{n-1} + \alpha = \alpha - \sum_{i=0}^{n-2} D_i,
\]

and

\[
\begin{bmatrix}
-1 & 0 & \ldots & 0 & 0 \\
1 & -1 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & -1
\end{bmatrix}
\begin{bmatrix}
\delta_1 \\
\vdots \\
\delta_{n-1}
\end{bmatrix}
= \begin{bmatrix}
D_0 - \alpha \\
D_1 \\
\vdots \\
D_{n-2}
\end{bmatrix},
\]

That means

\[
\begin{bmatrix}
\delta_0 \\
\delta_1 \\
\delta_2 \\
\vdots \\
\delta_{n-2} \\
\delta_{n-1}
\end{bmatrix}
= \begin{bmatrix}
\alpha \\
\alpha - D_0 \\
\alpha - D_0 - D_1 \\
\vdots \\
\alpha - \sum_{i=0}^{n-3} D_i \\
\alpha - \sum_{i=0}^{n-2} D_i
\end{bmatrix}
\]

The solution can be written in the following general form.

\[
\delta_0 = \alpha, \quad \text{and} \quad \delta_j = \alpha - \sum_{i=0}^{j-1} D_i
\]

The deforming distance is minimal, when \(\alpha\) equal to the average of \(\{0, D_0, \sum_{i=0}^{1} D_1, \ldots, \sum_{i=0}^{n-2} D_1\}\).

### III. EVOLVING FOR THE MINIMAL DEFORMING DISTANCE

An EC-based method is proposed in this section to find a match assignment \(M\) having the minimal deforming distance. Consider string-to-string correction problem. Let \(A = \{a_1, a_2, \ldots, a_n\}\) and \(B = \{b_1, b_2, \ldots, b_m\}\) are two strings of characters. Three types of edit steps: 1) insert - insert a character into the string, 2) delete - delete a character from the string, and 3) replace - replace one character with a different character, are used to correct the string \(A\) to \(B\).

Let \(A(i)\) and \(B(i)\) be the string which contains the first \(i\) characters of \(A\) and \(B\), respectively. Since the minimal edit cost from \(A(i+1)\) to \(B(i+1)\) can be estimated based on the value from \(A(i)\) to \(B(i)\), \(A(i+1)\) to \(B(i)\), and \(A(i)\) to \(B(i+1)\). String-to-string correction problem is generally solved by dynamic programming[12]. As shown in figure 6, a table that span by the string \(A\) and \(B\) is established. The cell value \(C(i,j)\) which represents the minimal cost from \(A(i)\) to \(B(j)\) can be computed as follows.

\[
C(i,j) = \min \left\{ \begin{array}{ll}
C(i-1,j) + 1 & , a_i \neq b_j \\
C(i,j-1) + 1 & , a_i = b_j
\end{array} \right.
\]

with \(C(i,0) = i\) for all \(i\), \(0 \leq i \leq n\), and \(C(0,j) = j\) for all \(j, 0 \leq j \leq m\). If \(C(i,j)\) is based on \(C(i-1,j)\), that means a deletion is applied here. If \(C(i,j)\) is based on \(C(i,j-1)\), that means an insertion is applied here. If \(C(i,j)\) is based on \(C(i-1,j-1)\), that means it’s a match or a replacement is applied here.

![Fig. 6. Solving the string correction problem by dynamic programming.](image)

Each arrow represents a dynamic programming operation. The DP-table is filled from the left-top to right-bottom corner.

By judging flattening operations as deletions, raising operations as insertions, and sharpening operations as replacements, a match assignment can be represented as a path on dynamic programming table of string-to-string correction problem. Figure 7 is an example of the graphical representation of a match assignment. In this example, there are one raising after \(s_0\), and two flattening at \(s_2\) and \(s_m\).

Because the value of \(C(i,j)\) is not related to the value of \(C(i-1,j-1)\), \(C(i,j-1)\), or \(C(i-1,j)\) and the \(S\) and \(T\) are circular connected, dynamic programming is not capable to estimate the deforming distance. However the edit paths of string-to-string correction problem are useful to generate all possible match assignments.

Figure 8 is the solution space generated from an example of deforming from a 3 angles polygon descriptor to a 4 angles polygon descriptor. The red line in each block shows the match assignment, and the brightness of the block represents the minimal deforming distance (see. Section II-C) of a match assignment. From this example, we noted that there are more than one local minimal (0.025 and 0.016) exist. To find the...
A. Individuals

An individual is a match assignment. A match assignment is capable to be represented as an edit path in the string-to-string correction problem (see Figure 7).

As shown as Figure 9, a match assignment is represented as a sequence of locations on the table used by Dynamic Programming. An example match \( \{(0,0),(0,1),(1,1),(2,2)\} \) means that \( s_0 \) is matched to \( t_0 \) and \( t_1 \). \( s_0 \) and \( s_1 \) are matched to \( t_1 \). And \( s_2 \) are matched to \( t_2 \).

B. Fitness Function

Giving a match assignment, the minimal deforming distance can be estimated by the methods introduced in Section II-C. The inverse of estimated minimal distance is used to represent the degree of fitness.

C. Initialization

As the example shown in Figure 10, the dashed path in center is called the center match assignment. The solid path in the left-top corner is the left-most match assignment, and the dotted path in the right-bottom corner is called the right-most match assignment. These three match assignments are used as the initial populations.

D. Mutation

To be a valid match assignment, a path \( P = (p_0^s, p_0^t), ..., (p_n^s, p_n^t) \) should obey the following rules.

1) \( p_i = p_{i-1} \) or \( p_i = p_{i+1} + 1 \).
2) \( (p_0^s, p_0^t) = (0,0) \).

3) \( (p_k^s, p_k^t) = (n,m) \), where \( k \) is the number of path steps, \( n \) and \( m \) is the number of angle for source and target polygons, respectively.

Based on these restriction, the possible movements among legal match assignments are listed as follows.

- When two elements \( (i,j) \) and \( (i+1,j+1) \) are in a row, the element \( (i,j+1) \) or \( (i+1,j) \) can be inserted into them. For example: \( (1,1),(2,2) \rightarrow (1,1),(1,2),(2,2) \) or \( (1,1),(2,1),(2,2) \).
- When three elements \( (i,j), (i+1,j) \), and \( (i+1,j+1) \) are in a row, the middle element can be removed. For example: \( (1,1),(1,2),(2,2) \rightarrow (1,1),(2,2) \).
- When three elements \( (i,j), (i,j+1) \), and \( (i+1,j+1) \) are in a row, the middle element can be removed. For example: \( (1,1),(1,2),(2,2) \rightarrow (1,1),(2,2) \).
E. Parent Selection Mechanism

For every iteration, the individuals having more chance to successfully produce new generation are selected as parents. The chance is measured by the number of neighboring individuals minus the failed mutation count.

F. Survivor Selection Mechanism

Two kinds of individuals are selected as the survivors: 1) the individuals at local optimal, and 2) the individuals having high chance to produce new generations. In order to track the status of individuals, two counts are recorded for each individual during the evolving process: 1) the number of failed mutations, and 2) the number of duplicated mutations, where a failed mutation is the mutation which generates a child having worse fitness score, and a duplicated mutation is the mutation which generates a child already in population. Higher failed mutation count means that an individual has higher chance to be a local optimal. Higher duplicated mutations count means that an individual has lower chance to produce new generations because there are already too many individuals in its neighboring areas.

G. Termination Condition

When most of the individuals in population are survived because they are judged as the local optimal, the evolving process terminated.

IV. EVALUATION

For evolutionary computing, spending more times for evolution usually provides higher chance to find the global optimal. In Section IV-A, randomly generated data are used to evaluate the efficiency of the EC-based similarity measurement. Section IV-B is the real-world data evaluation.

A. Evaluation using randomly generated data

Stricter termination condition shorten the convergency time but produce worse solution. Whereas looser termination condition provides better solution but longer convergency time. In this section, random generated data are evaluated with different termination conditions.

A die-out threshold is used to decide when to remove an individual out of possible parent list. The die-out threshold is defined as follows,

\[
\text{die-out threshold} = \frac{f + r}{n - s},
\]

where \( n \) is the number of neighboring solutions, \( s \) is the number of successful mutations, \( f \) is the number of failed mutations, and \( r \) is the number of redundant mutations.

Figure 11 shows the relation between the probability of finding the global optimal and the die-out threshold. In this evaluation, the mutation count is almost linearly increased when enlarging the threshold. However, the probability of getting the global optimal is higher than 90% when the threshold larger than 1.5. When the threshold is larger than 5, the results is always the global optimal.

B. Evaluation using Taiwan Stock Market data

To validate the proposed similarity measurement method, the frequently discuss financial quantities, stock price and EPS (earn per share), are used to check if the proposed method generates reasonable results. That is, a group of investors believe that the stock price should base on how much interest, which is decided by EPS, they can get, and another group of investors think that the desire for future improvement is more important.

Although the polygon descriptors are designed for arbitrary dimension data, two dimension data are applied in the evaluation to easily render the resulted polygon descriptors. At first, 20 years stock price and EPS data are used to learn the polygon descriptors month by month. An example of the estimated polygon descriptor is drawn in Figure 12. The horizontal axis is stock price, the vertical axis is EPS and the points are observations. As shown in this example, a four-axes polygon descriptor (bold-black lines) is learned to represent the distribution of observations.

Then, the similarities of every two polygon descriptors are measured by the proposed minimal deforming distance method. As shown as figure 13, the resulted similarity matrix are measured by the proposed minimal deforming distance.

\[
y - x = \frac{1}{(y - x)^2}
\]

Figure 13 shows the relation between the similarity of two polygon descriptors and the die-out threshold. The thin-line shows the relation between the similarity and the die-out threshold.
Fig. 12. An example of the resulted polygon descriptor on EPS(vertical axis)-price(horizontal axis) data. Depending on the learned parameter of polygon descriptor, the contour of its corresponding polygon is plotted in bold lines.

Fig. 13. The similarities between every two months TW stock market data. The brightness of the pixel at the left-top corner represents the similarity of stock market behavior in Jan,1986 and Jan,1986. And the brightness of the pixel at the right-bottom corner represents the similarity of stock market behavior in March, 2006 and March, 2006.

The resulted similarities show that the monthly data are similar to each other from 2003 to 2006. It also show that the monthly data at the age of 90th are quiet similar to each others too. However, the two group of data are different. The corresponding polygon descriptor is in a more dependent shape (a leaning solid line) at 2003 and 2006 and a more independent shape (a straight block) at 90s. It reflects the change from the dreaming age at 90s to the realistic age after 2003. The estimated similarities match the observations made by investors.

Besides, since most of the similar entries are found after 2003, that means the investors should make their decision based on the experience after 2003 rather than the experience at 90s. The similarity of the data before 1993 is also better than the 90s.

However, more economical and financial series, such as volume, more historical data, and various comparison setup, such as stocks to stocks comparison, are needed to be imported for mining more investment references.

V. CONCLUSION

Since the data distribution reflects the interactions among stock market quantities, the shape of data distribution is proposed as the feature to represent the stock market behavior. The polygon descriptor is proposed to learn the best fitted parameters of polygon to serve as the numerical form of the shape of data distribution. Besides, based on the geometric meaning of shapes of distribution, the deforming distance is proposed as the measurement of similarity between polygon descriptors.

In addition, an evolutionary computing based method is proposed to estimate the deforming distance. Using the divide and conquer techniques, the problem is reduced to a string-to-string correction like problem. For every possible solution of the string-to-string correction like problem, the optimal solution can be calculated directly. Therefore, the deforming distance can be estimated efficiently.

Using the proposed similarity measurement of stock market behavior, various of applications can be developed to provide investment suggestions by mining references from financial archives. Take for example, the similar historical periods can be queried or the stock groups having the similar behavior can be found.

REFERENCES