Novel pruning based hierarchical agglomerative clustering for mining outliers in financial time series

D. Wang[1], P. J. Fortier[2] & H. E. Michel[2]
[1]*Western Asset Management, USA*
[2]*Department of Electrical and Computer Engineering, University of Massachusetts, Dartmouth, USA*

Abstract

Investors must make informed decisions using partial and imperfect information. As accuracy and completeness of information held by the investor rise, the probability for better decision making also rises. Similarity search based outlier detection in financial time series is key to making better decisions for many investment strategies and portfolio management techniques. This motivates people to utilize numerous data mining techniques to discover similarities from massive financial time series data pools. The research introduces a novel pruning based Hierarchical Agglomerative Clustering (HAC) algorithm to search for similarity among financial time series in high dimensional space using securities in the S&P500 index as experimental data. The algorithm is based on vertical and horizontal dimension reduction algorithms [11] and a unique similarity measurement definition [12] with the time value concept. This paper discloses a series of experiment results that illustrate the effectiveness of the algorithm.

Keywords: outlier, data mining, computational finance, financial time series, similarity search, high dimension, clustering.

1 Introduction

We propose a novel similarity search in high dimensional financial time series by using a pruning based HAC algorithm. The similarity search is performed after dimensionality reduction, which composes of an Attributes Selection (AS) algorithm [11] and a Piecewise Linear Representation (PLR) based Segmentation

WIT Transactions on Information and Communication Technologies, Vol 41, © 2008 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/CF080041

and Pruning (SP) algorithm as presented in Wang [11], and a unique similarity measurement definition [12] based on PLR represented financial time series objects.

The paper is organized as follows: Section 2 presents the problem statement to be addressed and summarizes the major contribution of this work toward academia. The background and related work of this topic is reviewed in Section 3. Section 4 states the business similarity. Similarity definition and measurement in Wang [12] is re-presented in Section 5. Section 6 describes the pruning based HAC algorithm to search similarity. The experiments and result analysis are presented in Section 7. The conclusion and future work are discussed in Section 8.

2 Problem statement and contribution

Investors usually have different investment and portfolio management strategies [10]. Some strategies include, cash surrogate passive strategy, passive strategy, outlier active strategy, and portfolio diversification [7] across segments etc. The key is to measure the similarity of the financial time series based on historical data before further mining. Existing similarity search algorithms [4, 6, 13] look for similar sequence based on given whole time series sequence or subsequence in non-high dimensional space and assume all historical data has equivalent weight for similarity. In real financial markets, the financial time series are driven by a high number of dimensions and historical data has time value. More recent data is more relevant to decision making based our research. Sophisticated unsupervised similarity search algorithms are needed to help investors compute similarity from hundreds or even thousands of securities. The goal is to develop a method that not only improves the efficiency in identifying similarity, but also provides a very high level of accuracy. If that is achieved, investors can expect to reduce the risk component in their investment decision making process. The developed similarity search algorithm contributes to the academic research fields in many ways.

- The first to study applying data objects time value function to high dimensional financial time series.
- Developed a novel pruning based HAC algorithm based on dimensional reduction algorithms.
- Developed a unique similarity measurement definition and calculation.
- Provided a similarity measurement foundation for developing potential models in financial markets.

3 Background and related work

3.1 Time series and sequence data mining research

A time series is a sequence of real numbers, representing the measurements of a real variable at equal time intervals. A time series database is a large collection of time series data such as that found in the NYSE, and NASDAQ stock

databases. Classical time series analysis can be categorized into identifying patterns [5] and forecasting. Similarity search is one of the major research topics within time series data and is used in subsequence similarity, clustering, indexing, and rule discovery. Existing similar sequence matching algorithms, map a data sequence of length n to a point in an n-dimensional space. Most of the algorithms define similarity between two data sequences using the Euclidean distance (ED) between two corresponding points. ED is easy to compute and scales to other problems such as clustering, and indexing. However, it doesn't allow for different baselines (i.e. stock A fluctuates at $80 and stock B at $10) and different scales (i.e. Stock A fluctuates between $80 and $120, stock B between $5 and $15). The challenge is to design solutions that attempt to strike a balance between accuracy and efficiency. Since search performance degrades exponentially as the dimensionality of the index structures increases, most of the existing algorithms reduce the dimensionality by mapping the n-dimensional points into the f-dimensional ones (f<n).

Wu, Salzberg and Zhang [13] proposed a subsequence based similarity search method particularly for stock market prediction analysis. However, the paper employs a b% indicator, a popular indicator derived from Bollinger Bands, and utterly relies on zigzag price movement to do prediction and therefore is not a high dimensionality approach. In reality, most sophisticated investors will consider not only the price, but also other dimensions (EPS, daily trading volume, 52 weeks low and highs, etc.) in making investment decisions.

3.2 Hierarchical agglomerative clustering

Hierarchical clustering is a deterministic algorithm. The hierarchical based algorithms use a distance matrix as clustering criteria and do not require the number of clusters k as an input, but require a termination condition. Two well-known heuristic methods are k-means and k-medoids algorithms [6].

In our study we use HAC to build a hierarchical classification of objects through a series of binary merges. The algorithm starts by merging the individual objects followed by merging these initial clusters. A slice across the hierarchy can then be taken at any level to provide the desired number of clusters. We experienced different rules for agglomeration, i.e. merge two clusters with the smallest distance between two inter-cluster elements.

4 Business similarity

The business definition of similarity is the foundation of our algorithm and implementation. Financial Analysts believe that individual stocks with a similar price change, a similar slope change, a similar length of change period and with other attributes such as similar trading volume, solid earning per share and price per earning are considered to be similar objects. Financial Analysts believe the reflect characteristics at the current time.

Price is the most important attribute in financial markets and is the foundation of the similarity definition in this study. The properties of subsequence of the

price attributes movement over time are typically of more concern by financial analysts.

For example, in the Figure 1, S1 and S3, S2 and S4 have same amplitudes with different time differences. S1 and S2, S3 and S4 have the same time difference with different amplitude differences. Financial analysts consider S1 and S3, S2 and S4 more similar while S1 and S2, S3 and S4 are less similar. So we apply different weights for normalized amplitude change and time difference. Based on similarity principles above from the business world, we proposed our similarity definition and measurements [12] in section 5.

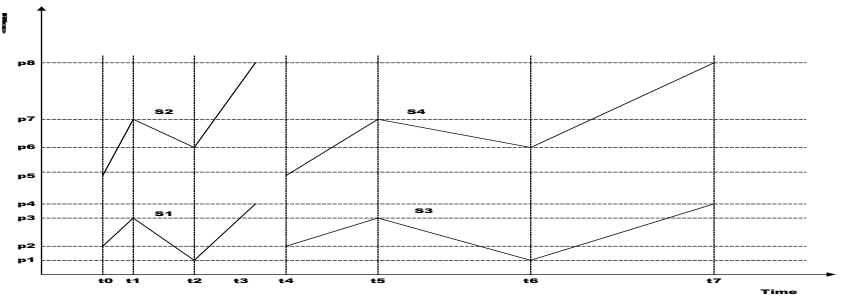


Figure 1: Price dimension similarity amplitude vs. time difference.

5 Similarity definition and measurement

Various indexing based similarity searching techniques have been explored for large databases. The biggest challenge in our research was how to implement similarity search in high dimensional data space across all time series data objects. The general idea is outlined below. Based on characteristics analysis of time series and data transformation, the following measurements are proposed: First, the zigzag shape of the Price dimension must be similar. We use a method similar to that proposed in Wu et al [13] based on permutation of the changing points and Euclidean distance between the objects based on the PLR. Second, the trend is always very important in a time series and should be part of the similarity search. We consider the Price attribute trend only for approving the concept. Third, for the remaining selected attributes, we calculate weighted distance based on the normalized values at changing points' time stamps. This is based on the correlation coefficient value of the attributes calculated during the Attribute Selection algorithm [11] against the Price dimension. Fourth, a time value function is applied to all data points. Based on the above analysis and observation, we proposed a similarity definition between two time series objects in Wang [12]:

Definition 1: Similarity measurement
Given two time series object S and S',

$S= ((t_1, \overrightarrow{D_1}\ ,\ \overrightarrow{T_1}\ ), (t_2, \overrightarrow{D_2}\ ,\ \overrightarrow{T_2}\ )\ldots(t_n, \overrightarrow{D_n}\ ,\ \overrightarrow{T_n}\ ), t_0, t_c),$

S'= (($t'_1, \overrightarrow{D'_1}$, $\overrightarrow{T'_1}$), ($t'_2, \overrightarrow{D'_2}$, $\overrightarrow{T'_2}$)…($t'_n, \overrightarrow{D'_n}$, $\overrightarrow{T'_n}$), t'_0, t'_c), where $\overrightarrow{D_n}$ = (D_{1n}, D_{2n}, D_{3n}, …, D_{kn}), $\overrightarrow{T_n}$ = (T_{1n}, T_{2n}, T_{3n}, …, T_{kn}) and k the number of final dimensions. S and S' are similar if S and S's satisfy the following properties:

- S and S' have the same number of segments and permutation of the Price dimension. This is the pre-condition for any of the following properties.
- Distance(S, S') < ε_1, where ε_1 is the given error threshold and Distance(S, S') = $\sum_{i=1}^{k}\sum_{t=0}^{n}\gamma_i * f(t) * d(S.A_{it}, S'.A_{it})$, γ_i is the aggregated correlation coefficient value of i[th] dimension A_i . *f(t)* is the Time Value Function. The distance refers to the amplitude change difference of Price and the other selected dimensions.
- Distance(T,T') < ε_2 , where ε_2 is given an error threshold and Distance(T,T') = $\sum_{t=1}^{n} f(t) * d(S.T, S'.T')$. T in this study is only for the Price dimension. *f(t)* is the Time Value Function and $f(t) = 1 - \lambda^{t_c - t}$ in this study where λ is an experiment value i.e. λ =0.95.

The similarity calculation uses backward order to calculate the value of each segment based on a time value function. According to the time value function, older data points within a selected time window contribute less to the final similarity value. The segment is the base unit for whole calculation. The similarity value is the average of the sum of the similarity of each segment. Each pair of segment similarities is calculated based on the price dimension, trend of the price dimension and other values from the final selected attributes with the time value function.

Equation 1: Segment similarity between two objects: [12]

$$
\begin{aligned}
&Sim\ (seg_1, seg'_1) \\
&= (1-\lambda^{(h-\frac{t_1+t_2}{2})}) * \left\{ \left| \begin{bmatrix} \alpha * |P_1 - P_2| + \beta * |T_{1-2}| + \alpha * \sum_{j=1}^{m} (\gamma_j * |A_{j1} - A_{j2}|) \end{bmatrix} \right. \right. \\
&\quad \left. \left. - \begin{bmatrix} \alpha * |P'_1 - P'_2| + \beta * |T'_{1-2}| + \alpha * \sum_{j=1}^{m} (\gamma_j * |A'_{j1} - A'_{j2}|) \end{bmatrix} \right| \right\} \\
&= (1-\lambda^{(h-\frac{t_1+t_2}{2})}) * \left[\alpha * \left(\left\| |P_1 - P_2| - |P'_1 - P'_2| \right\| + \sum_{j=1}^{m} \left(\gamma_j * \left(\left\| |A_{j1} - A_{j2}| - |A'_{j1} - A'_{j2}| \right\| \right) \right) \right) + \beta * \left\| |T_{1-2}| - |T'_{1-2}| \right\| \right]
\end{aligned}
$$

where $(1-\lambda^{(h-\frac{t_1+t_2}{2})})$ is the time value function and h is the number of business days within the user specified time period. t_1 is the date for ending changing point and t_2 is the date for starting changing point of seg_1, where seg_1 stands for segment 1. P_1 and P_2 stand for the ending and starting plast value. α and β

stand for the relative contribution scale of the attributes amplitude and trend to the similarity respectively. $T_{1\sim2}$ stands for the trend degree value for seg_1 with end point as 1 and start point as 2.

6 Hierarchical agglomerative clustering

6.1 Pruning based HAC algorithm

We customized the traditional HAC algorithm through application of a pruning condition. After our PLR based Segmentation and Pruning algorithm [11], the initial data set is divided into groups based on the number of segments. Moreover, the groups are divided into subgroups based on a permutation. The data objects with the same permutation are grouped into sub groups. We consider these objects as having more similar characteristics. First, the similarity values only need to be computed within subgroups. Second, the clusters formation can not be across sub groups. With these two conditions, we prune the cluster formation based on the traditional HAC algorithm. The following is our customized HAC algorithm pseudo code.

```
/**************customized HAC algorithm pseudo-code*****************/
 Begin;
Read in all PLRed stock file names;
Initialize HAC instance hac;
//step 1: grouping by number of segments
   hac.groupBySegNumber();
// step 2: Inside each group, filter stock by permutation algorithm
   hac.groupByPermutation(){
          Perform permutation inside each segment group;
          For every two stocks inside each segment group
                    sort two stocks' low points by plast attribute;
                    if all low point index are same;
                              continue to high points permutation;
                    Else
                              Skip these two stocks because they can not be permuted;
                    sort two stocks' high points by plast attribute;
                    If all high point index are same;
                              Put in one sub group under the segment group;
                    Else
                              Skip these two stocks because they can not be permuted;
}
//step 3: HAC clustering based on segment grouping and permutation sub grouping
Load each sub group within segment group as individual group;
Create HACMatrix for each individual group;
Perform HAC clustering based on HACMatrix(){
          //Compute Similarity
           For every two stocks inside HACMatrix group{
                    Compute similarity value;
                    Store the similarity value along with two stocks index in an Array;
                    Compute similarity calculation;
          }
          //customized clustering based on minimum distance between clusters
           Find minimum stocks' similarity value;
          Produce a cluster by joining the stocks with this minimum similarity value;
          Update HACMatrix;
          Recompute similarity value between new grouped stocks and other stocks;
}
/**************customized HAC algorithm pseudo-code*****************/
```

As depicted in the pseudo code above, first objects are grouped by the number of segments. Second further group objects to produce subgroups inside each group through the permutation process. The permutation process divides all changing points of any two objects into upper changing points and lower changing points. If the index is the same for each point, the two objects belong to

the same subgroup. The third step performs HAC based on the produced subgroups. The algorithm creates a HAC matrix to store the similarity values for each individual subgroup. After the cluster is generated at each step, the algorithm updates the matrix and re-computes the similarity values between the new cluster and other clusters or objects.

7 Experiment and result analysis

7.1 Raw data and attribute set

The Standard & Poor 500 Composite (S&P 500) stocks from BLOOMBERG© were selected as our raw data. We obtained 15 years of historical data with 95 attributes ranging from the pricing attributes to fundamental financial attributes. In order to adapt the environmental effect, also collected were 11 key economic indicators for the United States over those 15 years from DATASTREAM.

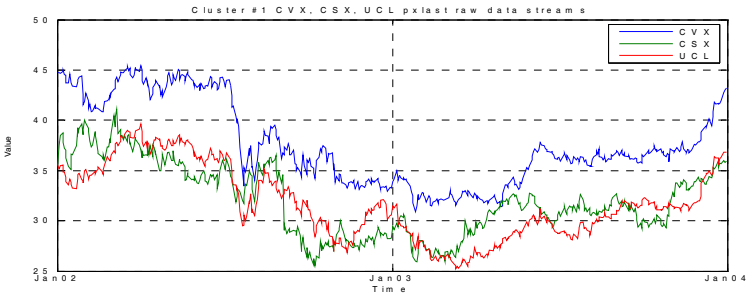
7.2 Attribute selection and construction

The selection of the initial attribute set was conducted using inputs from financial market domain experts. Three categories of attributes: market attributes, financial attributes, and environmental attributes were selected. Besides these directly accessible attributes, others were created to better measure stock performance. For instance, $\ln \frac{\Pr ice_t}{\Pr ice_{t-1}} = \gamma_{t-1}$: which reflects the ratio of return between the current day and last day.

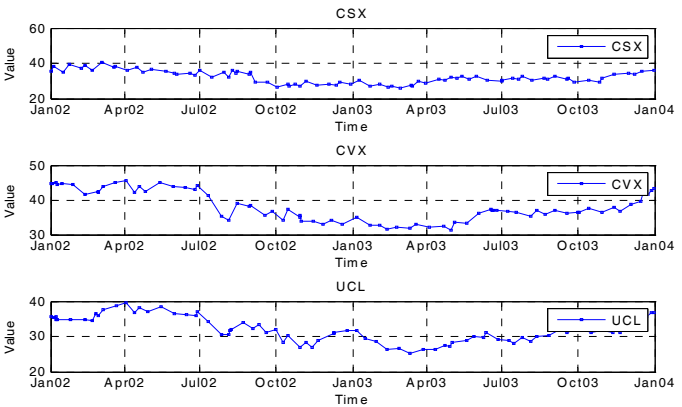
7.3 Sample clusters analysis

Figure 2 below represents groups of detected similar objects based on pxlast pruning thresholds equal to 12.5% with the data ranging from January 2, 2002 to December 30, 2003. We explain each figure in detail according to our framework but will only focus on Price amplitude change and trend degree.

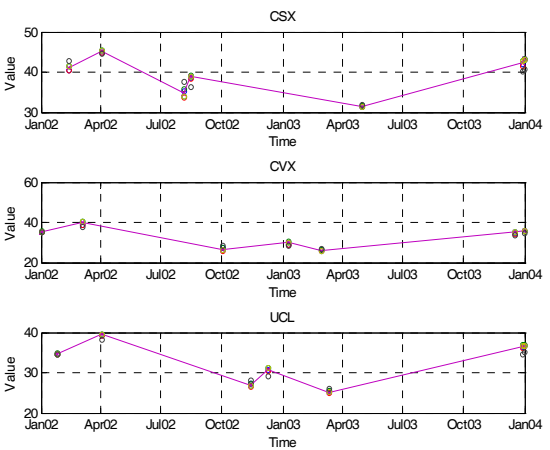
All three objects CVX, CSX, and UCL have 6 segments. All three objects have the same upper and lower points which ensure that the basic shapes of all three objects are the same. CVX and UCL joined as one cluster first. CSX is joined to the cluster in second step. As you can notice from figure above, the reason why CVX and UCL are joined as one cluster first is due to the time value function. The first segment (backward) of CVX and UCL is more similar since both segments are very short with similar amplitude change. The difference between these three objects is mostly seen in segment #3 and #4. Segment #3's objects have very close amplitude change which is 2 times the weighted trend change in our experiment. The amplitude change plays a more important role. Also, note that the trend change is not too far away.



(a) Raw data streams



(b) PLR segments representation of raw data stream before pruning



(c) Pruning result used to compute the similarity

Figure 2: Cluster #1 CVX, CSX, and UCL with 12.5% puning threshold.

Therefore they are considered to be very close by the algorithm. There are also other selected attributes denoted as "o" in the figure and their amplitude changes contribute to the final similarity value as well. You can see they are similar over the Price dimension in fact, especially the CVX and UCL stock objects.

8 Conclusion

In this study, we introduced a pruning based Hierarchical Agglomerative Clustering algorithm over high dimensional financial time series. The pruning based Hierarchical Agglomerative Clustering algorithm is based on a piecewise linear represented multi-dimensional data object, produced by our Segmentation and Pruning algorithm over high dimensional financial data along with our Attribute Selection algorithm. Our experiments show that our vertical (Attribute Selection algorithm) and horizontal (PLR based Segmentation and Pruning algorithm) dimension reduction algorithms [11] are able to present the objects in concise and accurate form to be further processed by our HAC algorithm.

We also proposed a unique similarity measurement [12] which is composed of three parts, the Price dimension amplitude similarity, Price dimension trend similarity, and other selected attributes amplitude similarity. Our similarity measurement properties were derived from finance industry analysts, giving this research a realistic method for identifying similarity. Moreover, we introduced a new time value concept to the similarity measurement. We find that more recent data contributes more significantly to the similarity measurement. Consideration of an entire time series history is not meaningful to current decision making according to financial analysts in the industry.

The experiments illustrate that our algorithms are able to find similarity efficiently within a large set of high dimensional data.

In this study, we focused on static time series data. We can improve the algorithms presented in this research to dynamic (incrementally updated) time series data by dealing with the moving perspective of the data. In addition, our high dimensional dimension reduction algorithms can be utilized to predict the dependent variable data characteristics in next time interval based on classification.

References

[1] Adam Blazejewski, Richard Coggins. Application of Self-organizing Maps to Clustering of High-frequency Financial Data. The second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32, Pages: 85–90, 2004.

[2] Steve Craighead and Bruce Klemesrud. Stock Selection Based on Cluster and Outlier Analysis. Fifteenth International Symposium on Mathematical Theory of Networks and Systems University of Notre Dame, August 12-16, 2002.

[3] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, Rajeev Motwani. Mining the Stock Market: Which Measure Is Best? The sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages: 487–496, 2000.
[4] Dimitrios Gunopulos(UC, Riverside), Gautan Das(Microsoft Research). Tutorial PM-2 Time Series Similarity Measures. The sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages: 243–307, 2000.
[5] Jiawei Han, Guozhu Dong, Yiwen Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. The fifteenth International Conference on Data Engineering, pages 106–115, 1999.
[6] Jiawei Han and Micheline Kamber. Data mining: Concepts and Techniques. 2001 by Academic Press.
[7] Mehmed Kantardzic, Pedram Sadeghian, Chun Shen. The Time Diversification Monitoring of a Stock Portfolio: an Approach Based on the Fractal Dimension. The 2004 ACM symposium on Applied computing, Pages: 637–641, 2004.
[8] Boris Kovalerchuk and Evgenii Vityaev. Data Mining in Finance: Advances in Relational and Hybrid Methods. 2000 by Kluwer Academic Publisher.
[9] Jessica Lin, Eamonn Keogh, Wagner Truppel. Clustering of Streaming Time Series Is Meaningless. The 8th ACM SIGMOD workshop on Research Issues in Data Mining and Knowledge Discovery, Pages: 56–65, 2003.
[10] Robert R. Trippi, Jae K. Lee. Artificial Intelligence in Finance & Investing: State-Of-The-Art Technologies for Securities Selection and Portfolio Management. Irwin Professional Publishing; Revised Edition (January 1, 1996).
[11] Dajun Wang, Paul J. Fortier, Howard E. Michel, and Theophano Mitsa. T-outlier and a novel dimensionality reduction framework in high dimensional time series financial data. The Second International Conference on Computational Finance and its Applications, June 2006
[12] Dajun Wang, Paul J. Fortier, Howard E. Michel, and Theophano Mitsa. Hierarchical Agglomerative Clustering Based T-outlier Detection. The Sixth IEEE International Conference on Data Mining, Risk Mining Workshop Page(s):731–738
[13] Huanmei Wu, Betty Salzberg, and Donghui Zhang. Online Event-driven Subsequence Matching over Financial Data Streams. ACM SIGMOD International Conference on Management of Data, Pages: 23–34, 2004.
[14] Kenji Yamanishi and Jun-ichi Takeuchi. A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time series Data. The eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages: 676–681, 2002.