Mining The Stock Market: Which Measure Is Best ?

[Extended Abstract]

Martin Gavrilov Dragomir Anguelov Piotr Indyk Rajeev Motwani

Department of Computer Science
Stanford University
Stanford, CA 94306-9010

{martinga, drago, indyk, rajeev}@cs.stanford.edu

ABSTRACT

In recent years, there has been a lot of interest in the database community in mining time series data. Surprisingly, little work has been done on verifying which measures are most suitable for mining of a given class of data sets. Such work is of crucial importance, since it enables us to identify similarity measures which are useful in a given context and therefore for which efficient algorithms should be further investigated. Moreover, an accurate evaluation of the performance of even existing algorithms is not possible without a good understanding of the data sets occurring in practice.

In this work we attempt to fill this gap by studying similarity measures for clustering of similar stocks (which, of course, is an interesting problem on its own). Our approach is to cluster the stocks according to various measures (including several novel ones) and compare the results to the "ground-truth" clustering based on the Standard and Poor 500 Index. Our experiments reveal several interesting facts about the similarity measures used for stock-market data.

Categories and Subject Descriptors

I.5.3 [**Clustering**]: Similarity measures

Keywords

time series, stock, clustering, similarity measures, data mining

1. INTRODUCTION

In recent years, there has been a lot of interest in the database community in mining time series data. Such data naturally arise in business as well as scientific decision-support applications; examples include stock market data (probably the most studied time series examples in history), production capacities, population statistics, and sales amounts. Since the data sets occurring in practice tend to be very large, most of the work has focused on the design of efficient algorithms for various mining problems and, most notably, the search of similar (sub)sequences with respect to a variety of measures [1, 2, 3, 4, 5, 7, 8, 10].

Surprisingly, little work has been done on verifying *which* measures are most suitable for mining of a given class of data sets. Such work is of crucial importance, since it enables us to identify similarity measures which are useful in a given context and, therefore, for which efficient algorithms should be further investigated. Moreover, an accurate evaluation of the performance of the existing algorithms is not possible without good understanding of the data sets occurring in practice.

In this work we attempt to fill this gap by studying measures for clustering of similar stocks (see [11] for more information about mining financial data). We obtained the stock-market data for 500 stocks from the Standard & Poor (S & P) index for the year 1998. Each stock is a series of 252 numbers, representing the price of the stock at the beginning of an operational day. Every time series is assigned to one out of 102 clusters (e.g. "Computers (Hardware)", "Oil and Gas", etc). Assuming this classification as a "ground-truth", we try to re-create it by running a clustering algorithm on the stock data using a variety of similarity measures. Then the respective measures are evaluated by comparing the resulting clustering to the original S & P classification (we use other evaluation methods as well).

Our experiments exhibited several very interesting properties of stock market data. First of all, the best clustering results were obtained for a novel measure proposed in this paper which uses *piecewise normalization*. The main idea behind this approach is to split the sequences into blocks and perform separate normalization within each block. Our results indicate that this approach yields quite powerful results for stock-market data. Another interesting observation is that comparing the *normalized derivatives* of the price sequences resulted in better clusterings than comparing the actual sequences (this phenomenon is widely known in the financial community). In particular, the combination of both of the above ideas results in the highest quality clustering obtained in this paper, depicted in Table 5. One can observe that the majority of our clusters have a one-to-one correspondence with one of the original S & P clusters, in the sense that for each of these clusters (say C) the S & P cluster closest *to* C also chooses C as *its* closest cluster. All of the remaining clusters are not nearest neighbors of any S & P cluster.

The high quality of the clustering obtained using derivatives has very interesting implications, since the performance analysis for most of the times series data structures assumes that the sequences are smooth[1], which is clearly not the case for the derivatives. Therefore, our results suggest that new algorithmic techniques should be developed, to capture the scenarios in which non-smooth time series data are present.

2. SETUP DESCRIPTION

The Data. We have used the Standard and Poor 500 index (S&P) historical stock data published at http://kumo.swcp.com/stocks/ . There are approximately 500 stocks which daily price fluctuations are recorded over the course of one year.

Each stock is a sequence of some length d, where $d \leq 252$ (the latter number is the number of days in 1998 when the stock market was operational, but d can be smaller if the company is removed from the Index). We used only the day's opening price; the data also contains the closing price, and the low and high stock valuation for the day.

The data also contained the official S&P clustering information which groups the different stocks into industry groups based on their primary business focus. This information was also used in our experiments, with the assumption that it provides us with a basis for a "ground-truth" with which we can compare and rate the results of our unsupervised clustering algorithm. We abstracted the 102 members of this S&P clustering into 62 "superclusters" by combining closely related ones together, e.g., "Automobiles" and "Auto (Parts)" or "Computers (Software)" with "Computers (Hardware)".

Feature Selection. Our feature selection approach consists of three main steps, depicted on the following picture:

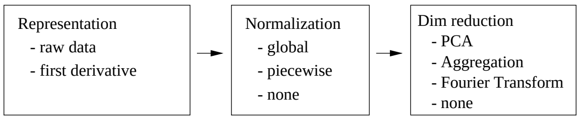

Figure 1: Feature extraction process

1. Representation choice: in this step we map the original time series into a point in d-dimensional space, where d is close to the length of the sequence. We use two types of mapping: identity and first derivative (or FD for short). In the first case, the whole sequence is considered to be one 252-dimensional point. In the second case, the i-th coordinate of the derivative vector is equal to the difference between the $(i+1)$-th and i-th value of the sequence. Both mappings are natural in the context of time-series data.

2. Normalization: in this step we decide if and how we should normalize the vectors. The standard normalization is done by computing the mean of the vector coordinates and subtracting it from all coordinates (note that in this way the mean becomes equal to 0) and then dividing the vector by its L2 norm. This step allows us to bring together stocks which follow similar trends but are valued differently, e.g., due to stock splits (note that our time series are not adjusted for splits). We also introduce a novel normalization method which we call *piecewise normalization*. The idea here is to split the sequence into windows, and perform normalization (as described above) *separately* within each window. In this way we take into account *local* similarities, as opposed to the global similarity captured by the normalization of the whole vector.

3. Dimensionality reduction: in this step we aim to reduce the dimensionality of the vector space while preserving (or perhaps even improving) the quality of the representation. Our first dimensionality reduction technique is based on the Principal Component Analysis (PCA). PCA maps vectors $\mathbf{x}^n$ in a d-dimensional space $(x_1, ..., x_d)$ onto vectors $\mathbf{z}^n$ in an M-dimensional space, where $M < d$. PCA finds d orthonormal basis vectors $\mathbf{u}_i$, called also *principal components*, and retains only a subset $M < d$ of these principal components to represent the projections of vectors $\mathbf{x}^n$ into the lower-dimensional space. PCA exploits the technique of Singular Value Decomposition(SVD), which finds the eigenvalues and eigenvectors of the covariance matrix
$$\Sigma = \sum_n (\mathbf{x^n} - \bar{\mathbf{x}})(\mathbf{x^n} - \bar{\mathbf{x}})^{\mathbf{T}}$$
where $\bar{\mathbf{x}}$ is the mean of all vectors $\mathbf{x^n}$. The principal components are shown to be the eigenvectors corresponding to the M largest eigenvalues of Σ and the input vectors are projected onto the eigenvectors to give the components of the transformed vectors $\mathbf{z}^n$ in the M-dimensional space.

   Our second technique, aggregation, is based on the assumption that local fluctuation of the stock (say, within the period of 10 days) is not as important as its global behavior, and therefore that we can replace a 10 day period by the average stock price during that time. In particular, we split the time domain into windows of length B (for $B = 5, 10, 20$ etc) and replace each window by its average value. Clearly, this decreases the dimensionality by a factor of B.

   Our third technique is based on the Fourier Transform (e.g., see [1] for the description). Basically, we used *truncated* spectral representations, i.e., we represented a time-series by only a few of its lowest frequencies.

Until now we described how we compare sequences of identical length. In order to compare a pair of sequences of different lengths, we take only the relevant portion of the

[1] E.g., [1, 7] approximate a sequence by removing all but few elements in the Fourier representation of a sequence; the quality of approximation in this case relies on the fact that high frequency component of a signal have low amplitude, which is clearly not the case for the derivative sequence.

longer time series, and perform the aforementioned processing only on that part. However, in order to perform PCA, we need to assume that all points have the same dimension, so instead, we pad all shorter sequences with zeros.

Similarity measure. We use the Euclidean distance between the feature vectors.

The clustering method. We use Hierarchical Agglomerative Clustering (HAC), which involves building a hierarchical classification of objects by a series of binary mergers (agglomerations), initially of individual objects, later of clusters formed at previous stages; see [6] for more details. A single "partition" (slice across the hierarchy) can then be taken at any level to give the desired number of clusters. We experimented with several rules for agglomeration. Merging two clusters which have the smallest maximum distance between two inter-cluster elements proved to yield the best results.

Rating and Comparing the Results. In order to evaluate the various results we got from applying the different feature selection mechanisms, we need the "ground-truth", i.e., some apriori classification which we can use for comparisons. The S&P clustering (provided in the input data as discussed above) serves exactly this purpose. This choice of "ground-truth" is based on the reasonable assumption that the pricing of each stock will be mainly influenced by factors specific to the particular industry sector to which this stock belongs.

We use the following measure that given the two clusterings $C = C_1 \dots C_k$ (say S & P clusters) and $C' = C'_1 \dots C'_k$ (say HAC clusters), computes their similarity using the following formula:

$$\mathrm{Sim}(C_i, C'_j) = 2\frac{|C_i \cap C'_j|}{|C_i| + |C'_j|}$$

and

$$\mathrm{Sim}(C, C') = \left(\sum_i \max_j \mathrm{Sim}(C_i, C'_j)\right)/k$$

Note that this similarity measure will return 0 if the two clusterings are completely dissimilar and 1 if they are the same. The measure has been already used for comparing different clusterings, e.g., in [9]. Note that this measure is not symmetric.

The above approach, although very natural, has the following disadvantage: the quality of the clusters depends not only on the similarity measure, but also on the clustering algorithm used. To avoid this problem, we complement our evaluation by using a method akin to Precision-Recall curves widely used in the Information Retrieval community. Here they are defined as follows. For each stock (say S) we "issue a query", that is rank all of the other stocks S' according to the distance between S and S', the closest ones first. We consider the stocks belonging to the same S & P cluster as S to be "relevant" and all other stocks to be "not relevant". Then we define a graph which for every rank i depicts the percentage of relevant stocks among the i stocks closest to P. The final curve is obtained by averaging the curves obtained for all stocks S.

3. RESULTS

We started from finding the right parameters for the eigenvalue decomposition for each particular feature-type.

After applying SVD on the raw data, it turned out that 97.62% of the eigenvalue weight is in the first 5 values, and 98.88% is in the first 10 values. This suggests that a projection of this 252-dimensional data in 10- or even 5-dimensional space will result in a negligible loss of information (see Figure 2 and 3).

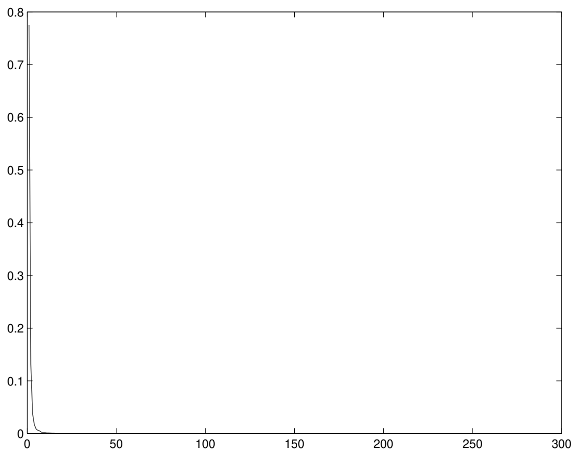

Figure 2: All eigenvalues for raw data before global normalization

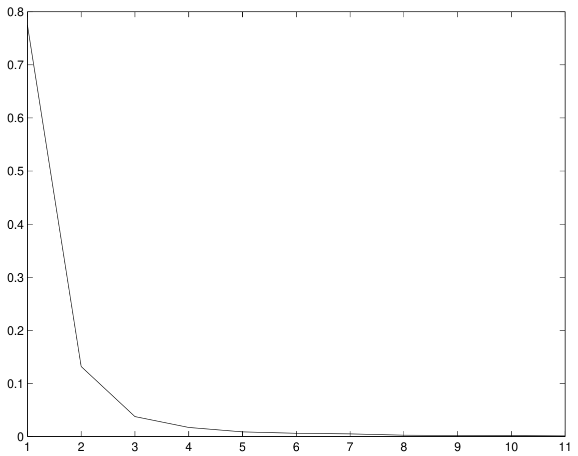

Figure 3: First 11 eigenvalues for raw data before global normalization

After global normalization of the data, we observe an increase in the error of dimensionality reduction. Now, in the first 10 eigenvalues is 90.47% of the weight, while 94.99% is in the first 20, and 98.19% in the first 50. Nevertheless, this could still allow us to achieve a significant reduction (see Figure 4 and 5).

After taking the first derivative, 61.95% of the eigenvalue weight is in the first 20 eigenvalues, 80.48% in the first 50, and 92.42% in the first 100 (see Figure 6).

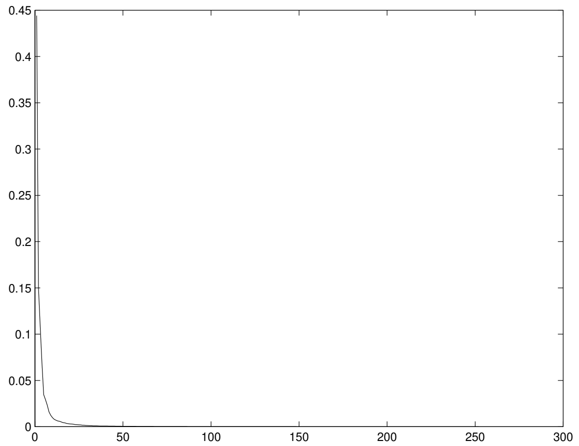

Figure 4: All eigenvalues for raw data after global normalization

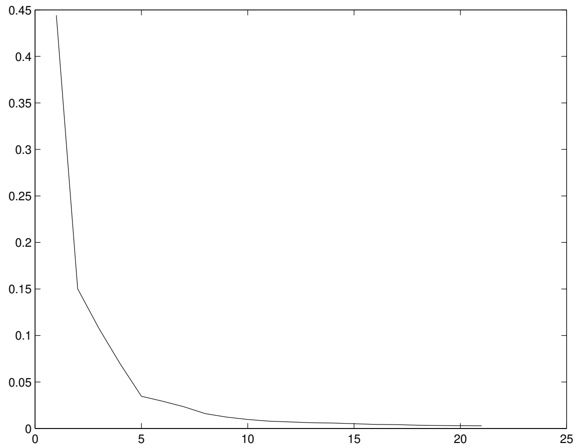

Figure 5: First 25 eigenvalues for raw data after global normalization

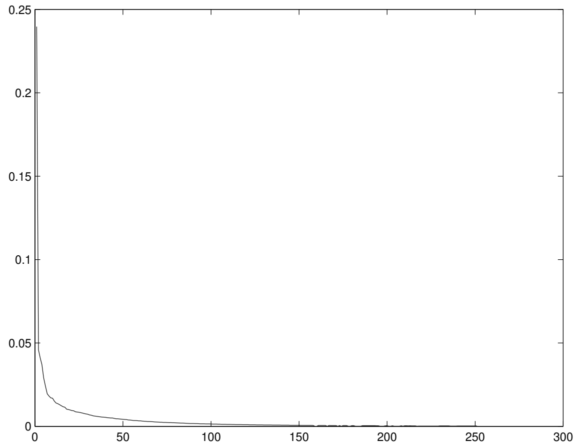

Figure 6: Plot of all 252 eigenvalues after applying the first derivative

The last transformation we used was taking the first derivative, then normalizing. Here, we observe the biggest dimensionality "dispersal". Only 75.23% of the eigenvalue weight is contained in almost half of the original dimensions, while 52.20% is in the first 50, and only 21.47% in the first 10 (see Figure 7).

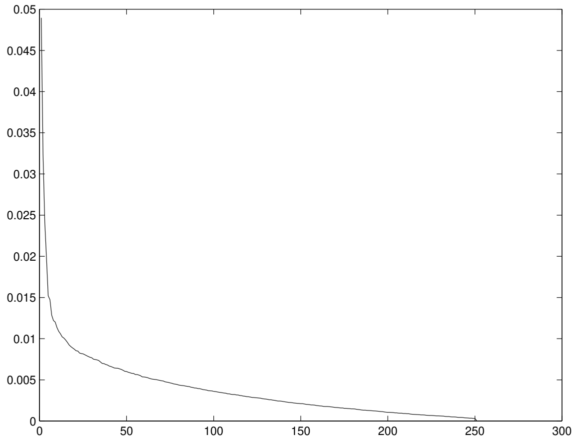

Figure 7: Plot of all 252 eigenvalues after applying globally normalized first derivative

To summarize: we get tremendous dimensionality reduction on the raw data – 97.62% of the information is only in 5 dimensions ! As we process the data (normalization and first derivatives (FD)), we get a dimensionality "dispersal". The most "dispersed" is the data that was pre-processed the most.

Therefore, for the further dimensionality reduction experiments, we choose the dimensionality for the above scenarios to be equal to respectively 5, 10, 50 and 100.

Clusters. The results of the clusterings (together with their parameters) are depicted in Tables 1, 2, 3 and 4. Table 1 was obtained by clustering the stocks according using 8 variants of the similarity metrics, from the space

$$\{\text{raw data, first derivative}\} \text{ x } \{\text{global normalization, no normalization}\} \text{ x } \{\text{all dimensions, reduced dimension}\}$$

The number of clusters was set to 62 (i.e., equal to the number of S & P "superclusters").

The second table shows the results for dimensionality reduction via aggregation. In this case, we vary the size of the aggregation window, the data representation (raw data or FD) and normalization (global or none).

We also performed experiments where the dimensionality reduction was done via Fourier Transform. Since, by Parseval's Theorem [1], the Euclidean norm of a raw vector is equal to the norm of its spectral representation, we performed the experiments on *truncated* spectral representa-

FD	Norm	Dims	Sim(S&P,HAC)	Sim(HAC,S&P)
N	N	all	0.183	0.210
N	N	5	0.197	0.210
N	Y	all	0.222	0.213
N	Y	10	0.211	0.212
Y	N	all	0.154	0.198
Y	N	50	0.172	0.207
Y	Y	all	0.290	0.298
Y	Y	100	0.310	0.310

Table 1: The clustering results, with PCA dimensionality reduction

FD	Norm	AggWin	Sim(S&P,HAC)	Sim(HAC,S&P)
N	N	none	0.183	0.210
N	N	5	0.192	0.217
N	N	10	0.193	0.215
N	N	20	0.192	0.213
N	Y	none	0.228	0.217
N	Y	5	0.217	0.212
N	Y	10	0.221	0.216
N	Y	20	0.215	0.220
Y	N	none	0.152	0.197
Y	N	5	0.190	0.211
Y	N	10	0.195	0.217
Y	N	20	0.178	0.208
Y	Y	none	0.288	0.294
Y	Y	5	0.225	0.217
Y	Y	10	0.230	0.231
Y	Y	20	0.211	0.211

Table 2: The clustering results, with dimensionality reduction via aggregation

tions, i.e., we kept only a few of its lowest frequencies. The resulting experiments are presented in Table 3.

In the last table (Table 4), we show the results obtained when using piecewise normalization. Again, we vary the window size (note that the role of the window is different than in the previous set of experiments) and data representation.

One can observe that the best result (i.e., with the highest Sim measure) was obtained when using the combination of piecewise normalization with window of length 15 and first derivative. The resulting clustering is depicted in Table 5. Unfortunately, the whole description of the clusters is too big to be included in the paper. However, we present the clusters in the following form. The ith row corresponds to one of the HAC clusters (say C_i). The second column shows the name of the S&P cluster which is the closest to C_i according to Sim(HAC, S&P) measure. The remaining columns show the names of S&P clusters which choose C_i to be the HAC cluster closest *to them*, listed in the order of similarity Sim(S&P, HAC), the most similar first. Notice that the column two and three are almost always equal, with the exception of clusters 4, 35, 36, or when the third column is empty.

FD	Norm	Freqs	Sim(S&P, HAC)	Sim(HAC,S&P)
N	N	5	0.191	0.197
N	N	10	0.203	0.204
N	N	25	0.192	0.196
N	N	50	0.193	0.202
N	Y	5	0.215	0.217
N	Y	10	0.210	0.208
N	Y	25	0.221	0.229
N	Y	50	0.225	0.224
Y	N	5	0.202	0.215
Y	N	10	0.189	0.209
Y	N	25	0.191	0.217
Y	N	50	0.190	0.212
Y	Y	5	0.198	0.209
Y	Y	10	0.235	0.236
Y	Y	25	0.247	0.240
Y	Y	50	0.232	0.234

Table 3: The clustering results after Fourier Transform

Window	FD	Sim(S&P,HAC)	Sim(HAC,S&P)
10	N	0.322	0.326
15	N	0.307	0.314
30	N	0.270	0.273
45	N	0.266	0.281
60	N	0.246	0.241
75	N	0.255	0.257
10	Y	0.338	0.334
15	Y	0.346	0.339
30	Y	0.330	0.329
45	Y	0.346	0.333
60	Y	0.316	0.310
75	Y	0.310	0.297

Table 4: The clustering results, with piecewise normalization

A few example clusters are depicted in Tables 6, 7, 8, 9 and 10.

Precision-recall curves. In order to make our observations independent from the clustering algorithms, we also computed Precision-Recall curves (PR-curves) for a variety of measures and compare them to the PR-curve for normalized derivatives (see Figures 8, 9 and 10). This allows us to make a visual estimation of the influence of various parameters (feature extraction algorithm, normalization etc) on the clustering quality. In each case, one can observe that the higher quality clustering corresponds to a PR-curve with higher values at the beginning of the curve, which corresponds to higher precision at the beginning.

4. DISCUSSION

The experiments described in the previous section support several interesting general observations. First of all, normalizing the input vectors in *any* form always improved the quality of the results. This behavior is very natural, since normalization enables us to reduce the effect of translation and scaling of the sequences. However, it turned out that

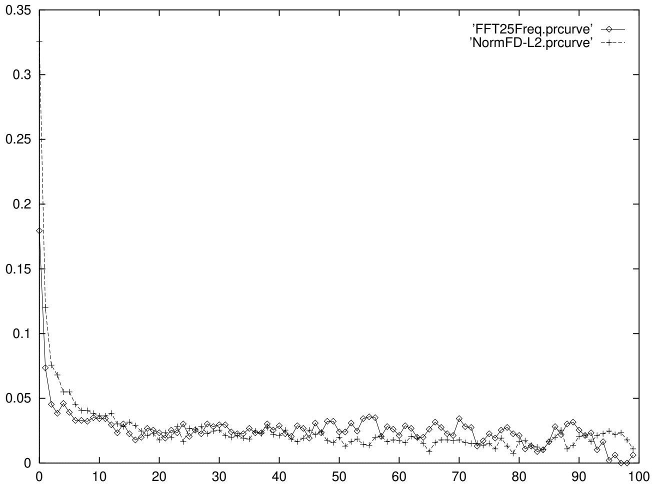

Figure 8: Fourier Transform vs. globally normalized derivatives

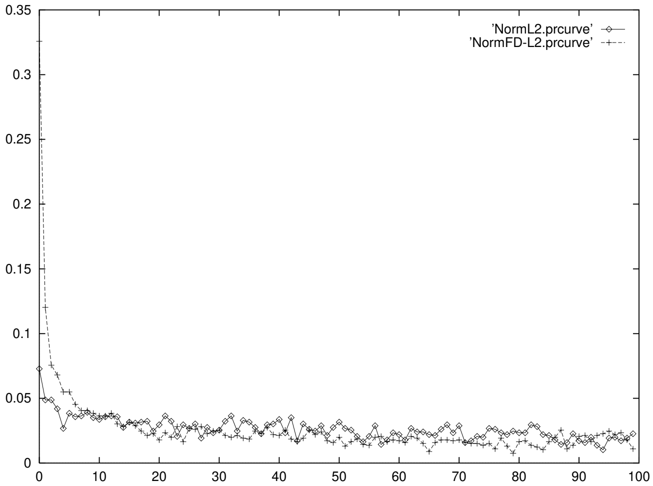

Figure 9: Globally normalized raw data vs. globally normalized derivatives

applying the same normalization to the whole sequence is not the best solution, and that one can obtain better results by using piecewise normalization, for a carefully chosen window size. The fact that piecewise normalization behaves so good can be explained by observing that it greatly reduces the effect of "local anomalies" on the distance between two stock indices. This is due to the fact that such "anomalous" events affect only one or two windows, while others can still adjust to each other if their behavior is similar, even if the actual valuations are very different.

Another observation is that using the normalized derivatives of the sequence data resulted usually in better results than using raw data. This phenomenon is widely known in the financial community and can be explained by arguments similar to the above ones, i.e., that the local anomalies have only limited influence on the distance between derivatives, as opposed to large influence in case of the raw data. However, we believe that more research has to be done in order to fully understand this behavior. For example, just computing the first derivatives without normalization actually worsens the performance (as compared to the raw data results).

Another set of interesting observations comes from the attempts to reduce the dimensionality of the data. Although the raw, unprocessed data seems to have the lowest intrinsic dimensionality (according to PCA), we are not able to build very good clusters out of it. It is actually interesting to note that in the case when we get best clusterings (i.e., we use normalized derivatives) the data does not seem to be prone to dimensionality reduction, since the projection onto a 100-dimensional space preserves only 75.23% of the total eigenvalue weight. However, even in this case, the clustering was better than for the full data.

5. RELATED WORK

There has been a significant amount of recent research focused on designing efficient algorithms for similarity search and mining of time series. This work has been started by Agrawal et al [1], who showed that indexing time sequences can be done very efficiently using Fourier Transform. This work has been extended in [7] to finding similar subsequences. Another important contribution has been done by Agrawal et al in [2], who also addressed the issue of sequence transformations (like scaling and translations) as well as noise resilience. Further work in this area has been done by Rafiei-Mendelzon [10], Bollobas et al [3], Das et al [4] and Huang-Yu [8].

6. ACKNOWLEDGMENTS

We would like to thank Prof. Carlo Tomasi and Kathy Pullen for their help.

7. REFERENCES

[1] R. Agrawal, C. Faloutsos, A. Swami, "Efficient Similarity Search in Sequence Databases", *Proc. of FODO'93*, Lecture Notes in Computer Science 730, Springer Verlag, 69-84.

[2] R. Agrawal, K. Lin, H. S. Sawhney, K. Shim, "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases", *Proc. of VLDB'95.*

[3] B. Bollobas, G. Das, D. Gunopulos, H. Mannila, "Time-Series Similarity Problems and Well-Separated Geometric Sets", *Proc. Symposium on Computational Geometry 1997*, p. 454-456.

[4] G. Das, D. Gunopulos, H. Mannila, "Finding Similar Time Series", *Proc. KDD* 1997, p. 88-100.

[5] G. Das, K. Lin, H. Mannila, G. Renganathan, P.Smyth, "Rule Discovery from Time Series", *Proc. KDD* 1998, p. 16-22.

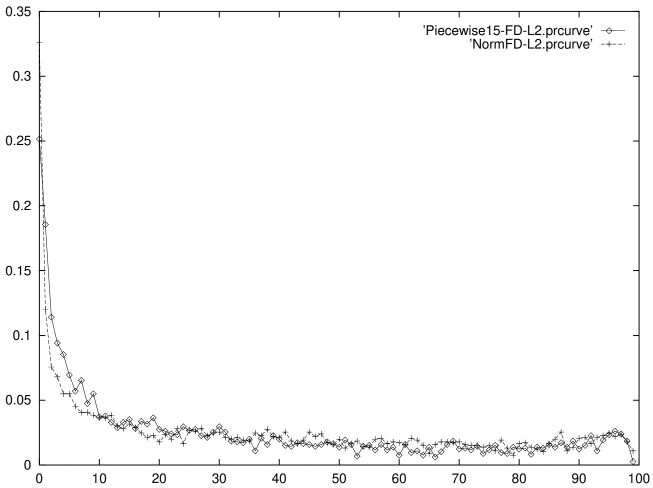

Figure 10: Piecewise normalization vs. global normalization (for derivatives)

[6] W. Frakes and R. Baeza-Yates, editors. "Information Retrieval: Data Structures and Algorithms", Prentice-Hall, 1992.

[7] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases",*Proc. SIGMOD'94*, 419-429.

[8] Y. Wu Huang, P. S. Yu, "Adaptive Query Processing for Time-Series Data" *Proc. KDD'99*, 282-286.

[9] B. Larsen, C. Aone, "Fast and effective text mining using linear-time document clustering", *Proc. KDD'99*, 16 - 22.

[10] D. Rafiei, A. Mendelzon, "Similarity-Based Queries for Time Series Data", *Proc. SIGMOD'97*, 13-25.

[11] A. S. Weigend, "Data Mining in Finance: Report from the Post-NNCM-96 Workshop on Teaching Computer Intensive Methods for Financial Modeling and Data Analysis (1997)", *Proc. Fourth International Conference on Neural Networks in the Capital Markets NNCM-96*, p. 399-411.

LU: Communications Equipment
CSCO: Computers (Networking)
DELL: Computers (Hardware)
INTC: Electronics (Semiconductors)
CPQ: Computers (Hardware)
IBM: Computers (Hardware)
KLAC: Equipment (Semiconductor)
AMAT: Equipment (Semiconductor)
TLAB: Communications Equipment
ADBE: Computers (Software & Services)
COMS: Computers (Networking)
AAPL: Computers (Hardware)
CA: Computers (Software & Services)
HWP: Computers (Hardware)
SEG: Computers (Peripherals)
EMC: Computers (Peripherals)
SUNW: Computers (Hardware)
MU: Electronics (Semiconductors)
TXN: Electronics (Semiconductors)
LSI: Electronics (Semiconductors)
UIS: Computers (Software & Services)
MOT: Communications Equipment
DGN: Computers (Hardware)
ORCL: Computers (Software & Services)
NOVL: Computers (Software & Services)
CMCSK: Broadcasting (Television, Radio & Cable)
TCOMA: Broadcasting (Television, Radio & Cable)
TWX: Entertainment
ADSK: Computers (Software & Services)
PMTC: Computers (Software & Services)
NSM: Electronics (Semiconductors)
AMD: Electronics (Semiconductors)
EK: Photography/Imaging

Table 6: Cluster # 1

WB: Banks (Major Regional)
STI: Banks (Major Regional)
PNC: Banks (Major Regional)
NOB: Banks (Major Regional)
BKB: Banks (Major Regional)
NCC: Banks (Major Regional)
BBK: Banks (Major Regional)
WFC: Banks (Major Regional)
MEL: Banks (Major Regional)
STT: Banks (Major Regional)
FITB: Banks (Major Regional)
BAC: Banks (Money Center)
SUB: Banks (Major Regional)
ONE: Banks (Major Regional)
KEY: Banks (Major Regional)
FTU: Banks (Money Center)
USB: Banks (Major Regional)
RNB: Banks (Major Regional)
CMA: Banks (Major Regional)
MWD: Financial (Diversified)
MER: Investment Banking/Brokerage
LEH: Investment Banking/Brokerage
SCH: Investment Banking/Brokerage
AIG: Insurance (Multi-Line)
SAI: Financial (Diversified)
NTRS: Banks (Major Regional)
BT: Banks (Money Center)
JPM: Banks (Money Center)
CMB: Banks (Money Center)
CCI: Financial (Diversified)
AXP: Financial (Diversified)
BK: Banks (Major Regional)
GDT: Health Care (Medical Products & Supplies)
KRB: Consumer Finance
NSC: Railroads
CSX: Railroads
DRI: Restaurants
TAN: Retail (Computers & Electronics)
MTG: Financial (Diversified)
MAR: Lodging-Hotels
HLT: Lodging-Hotels
MMM: Manufacturing (Diversified)
PCAR: Trucks & Parts

Table 7: Cluster # 3

BR: Oil & Gas (Exploration & Production)
APA: Oil & Gas (Exploration & Production)
APC: Oil & Gas (Exploration & Production)
MRO: Oil (Domestic Integrated)
ORX: Oil & Gas (Exploration & Production)
PZL: Oil (Domestic Integrated)
KMG: Oil & Gas (Exploration & Production)
AHC: Oil (Domestic Integrated)
UCL: Oil (Domestic Integrated)
UPR: Oil & Gas (Exploration & Production)
SNT: Natural Gas
OXY: Oil (Domestic Integrated)
XON: Oil (International Integrated)
MOB: Oil (International Integrated)
CHV: Oil (International Integrated)
ARC: Oil (Domestic Integrated)
P: Oil (Domestic Integrated)
RD: Oil (International Integrated)
AN: Oil (International Integrated)
MDR: Engineering & Construction
RDC: Oil & Gas (Drilling & Equipment)
SLB: Oil & Gas (Drilling & Equipment)
HAL: Oil & Gas (Drilling & Equipment)
BHI: Oil & Gas (Drilling & Equipment)
HP: Oil & Gas (Drilling & Equipment)

Table 8: Cluster # 8

F: Automobiles
CAT: Machinery (Diversified)
GM: Automobiles
DAL: Airlines
AMR: Airlines
U: Airlines
LUV: Airlines
CYM: Metals Mining
PD: Metals Mining
AR: Metals Mining
RLM: Aluminum
AA: Aluminum
AL: Aluminum
N: Metals Mining
CS: Computers (Networking)
ALT: Iron & Steel
SIAL: Chemicals (Specialty)

Table 9: Cluster # 22

CPL: Electric Companies
FPL: Electric Companies
DUK: Electric Companies
AEE: Electric Companies
PEG: Electric Companies
HOU: Electric Companies
UCM: Electric Companies
EIX: Electric Companies
PCG: Electric Companies
CIN: Electric Companies
BGE: Electric Companies
NSP: Electric Companies
AEP: Electric Companies
ED: Electric Companies
SO: Electric Companies
DTE: Electric Companies
FE: Electric Companies
D: Electric Companies
CSR: Electric Companies
GPU: Electric Companies
TXU: Electric Companies
PPL: Electric Companies
ETR: Electric Companies
PE: Electric Companies

Table 10: Cluster # 38

No.	HAC vs S&P	S&P vs HAC 1	S&P vs HAC 2	S&P vs HAC 3
0	Natural Gas (0.416667)	Natural Gas (0.416667)	Entertainment (0.117647)	
1	Computers (0.6)	Computers (0.6)	Electronics (0.272727)	Equipment (0.114286)
2	Retail (0.518519)	Retail (0.518519)		
3	Banks (0.657534)	Banks (0.657534)	Financial (0.185185)	Lodging-Hotels (0.08889)
4	Agricultural (0.285714)	Electrical (0.166667)		
5	Computers (0.129032)			
6	Biotechnology (0.25)	Biotechnology (0.25)	Office (0.222222)	
7	Truckers (0.285714)	Truckers (0.285714)		
8	Oil (0.901961)	Oil (0.901961)		
9	Household (0.4)	Household (0.4)		
10	Personal Care (0.285714)	Personal Care (0.285714)		
11	Iron (0.428571)	Iron (0.428571)	Footwear (0.4)	
12	Telephone (0.592593)	Telephone (0.592593)	Tobacco (0.190476)	
13	Natural Gas (0.153846)			
14	Homebuilding (0.333333)	Homebuilding (0.333333)	Air (0.133333)	
15	Financial (0.142857)			
16	Services (0.111111)			
17	Photography/Imaging (0.25)			
18	Electric (0.142857)			
19	Retail (0.238095)			
20	Chemicals (0.216216)			
21	Auto (0.166667)			
22	Airlines (0.380952)	Airlines (0.380952)	Aluminum (0.3)	Metals (0.363636)
23	Housewares (0.4)	Housewares (0.4)	Engineering (0.222222)	
24	Consumer (0.222222)	Consumer (0.222222)		
25	Auto (0.24)	Auto (0.24)		
26	Oil (0.129032)			
27	Hardware (0.166667)	Hardware (0.166667)	Broadcasting (0.133333)	
28	Health (0.55814)	Health (0.55814)		
29	Aerospace/Defense (0.222222)			
30	Insurance (0.27027)			
31	Homebuilding (0.333333)			
32	Gaming (0.285714)	Gaming (0.285714)		
33	Insurance (0.368421)	Insurance (0.368421)		
34	Waste (0.333333)	Waste (0.333333)		
35	Restaurants (0.210526)	Specialty Printing (0.117647)		
36	Chemicals (0.214286)	Trucks (0.181818)		
37	Services (0.222222)	Services (0.222222)		
38	Electric (0.96)	Electric (0.96)		
39	Photography/Imaging (0.3333)	Photography/Imaging (0.3333)		
40	Gold (0.909091)	Gold (0.909091)		
41	Publishing (0.266667)	Publishing (0.266667)	Building (0.2)	
42	Natural (0.133333)			
43	Paper (0.608696)	Paper (0.608696)	Containers (0.3)	
44	Beverages (0.4)	Beverages (0.4)	Agricultural (0.4)	
45	Distributors (0.222222)	Distributors (0.222222)	Foods (0.210526)	
46	Telecommunications (0.153846)			
47	Foods (0.133333)			
48	Computers (0.0714286)			
49	Iron (0.25)			
50	Investment (0.25)	Investment (0.25)		
51	Restaurants (0.333333)	Restaurants (0.333333)		
52	Chemicals (0.307692)	Chemicals (0.307692)		
53	Railroads (0.333333)	Railroads (0.333333)		
54	Manufacturing (0.173913)	Manufacturing (0.173913)	Machinery (0.153846)	
55	Paper (0.428571)			
56	Aerospace/Defense (0.666667)	Aerospace/Defense (0.666667)		
57	Telecommunications (0.333333)	Telecommunications (0.333333)		
58	Leisure (0.285714)	Leisure (0.285714)	Textiles (0.222222)	
59	Savings (0.666667)	Savings (0.666667)		
60	Power (0.666667)	Power (0.666667)	Communications (0.2)	
61	Computers (0.0689655)			

Table 5: Comparison of S&P and HAC-62 clusterings.