Algebraic Techniques for Technical Analysis of Stock Market Data.

Naren Ramakrishnan
Department of Computer Sciences
Virginia Tech.

and

Ananth Grama
Department of Computer Sciences
Purdue University.

Motivation:

- Current technical analysis frameworks rely on rudimentary methods such as moving averages and standard deviations to analyze individual sequences of indicators. This is dangerous since individual stocks may be sensitive to changes in fundamentals.
- Applying such techniques to entire indices damps out critical patterns exhibited by groups of stocks.
- Applying rules mined from indices (when available) universally across all stocks in the index lead to sub-optimal trading strategies.
- Patterns in a single stock or a group of stocks over variable time-scales are damped out by techniques such as moving averages.

Technical Objectives:

Analyzing time series data of stock prices with a view to identifying specific trends:

- Given a set of indicators and a selection of stocks, identify stocks exhibiting similar behavior with respect to the indicators (error bounded grouping).
- Identify dominant behavior of each grouping as a time-series of indicators.
- Analyze dominant behavior vectors for patterns at varying time-scales.

Ongoing work:

- Identifying and analyzing phenomena at variable time-scales within the same time-series. For example, precisely estimating trading channels and Bollinger bands.

Algebraic Underpinnings:

Consider the problem of analyzing a large number of very high dimensional binary attributed vectors.

The objective of analysis is to determine a (much) smaller number of representative binary attributed vectors such that every vector in the original set is within a bounded (given) Hamming distance ε from some vector in the representative vector set.

- The problem is np-complete (exponential in dimensionality).
- There are no methods currently available (other than ours to the best of our knowledge) that scale to extremely high dimensions while guaranteeing error bounds.
- The problem can also be thought of as error-bounded clustering or vector quantization over discrete (binary) spaces.

Compressing Binary-Attributed Vectors:

Example:

$$\text{Attribute vector}\begin{matrix}1\\2\\3\end{matrix}\begin{bmatrix}0 & 1 & 1\\0 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

Can simply be represented as:

2: [0 1 1]

1: [1 1 1]

Notice that this induces a mapping of each vector to one of the representative vectors based on the hamming distance. The grouping of vectors is inherent in this mapping. Also note that the representative vectors themselves quantify dominant group behavior.

This problem is, however, extremely difficult computationally.

Compressing Attribute Vectors:

Consider the following rank-1 set of vectors (stacked together as a matrix in which each row represents a vector):

$$T = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Since the order of the vectors is not important, this is equivalent to 4 vectors each being [0,1,0,1,1].

<u>But</u>: Attribute vector sets are never rank-1.

Compressing Attribute Vectors:

Decompose matrix into sequence of rank-1 matrices (Singular Value Decomposition or SVD)!

However:

- Singular vectors are orthogonal. This introduces negative values into the representative vectors!
- Singular vectors (rows) contain non-integral (non-discrete or continuous) values that often do not make physical sense for binary attributed vectors (a stock is either up or down-it is never 0.55 up).
- Non-integral column vectors do not make sense as approximations to rows.

Compressing Attribute Vectors:

What about non-integral values?

Use discrete transforms (SDDs)!

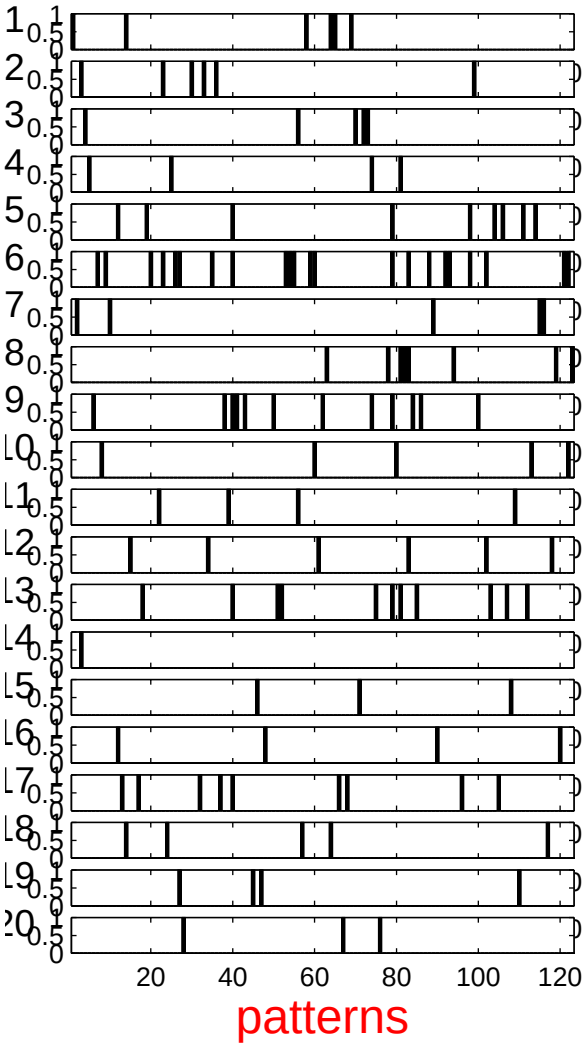
Discrete transforms are variants of SVDs in which vectors can only take values 0/1/-1. Singular values can take arbitrary values though.

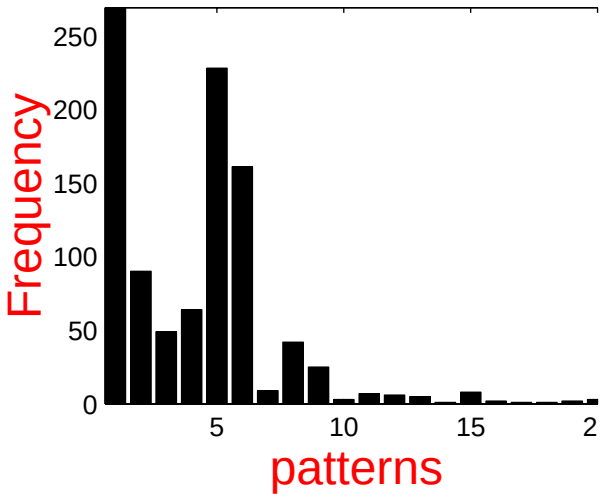
Compressing Attribute Vectors:

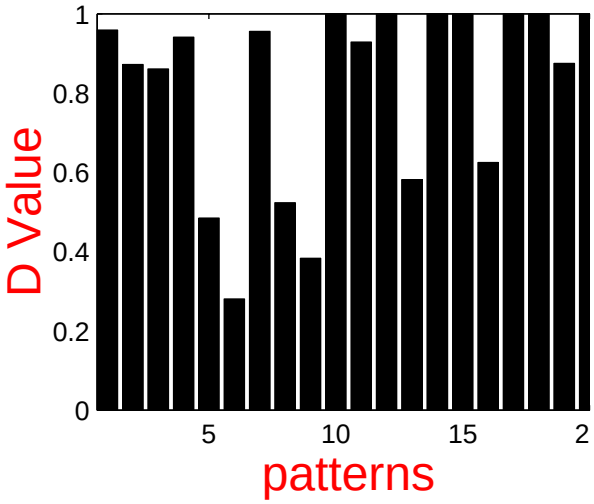
What about relaxing orthogonality?

- Compute first discrete singular vector.
- Eliminate all attribute vectors that are well approximated by this singular vector.
- If no vectors match the singular vector in the approximate sense, remove the best few matches and re-insert them at the end.
- Repeat process until patterns are statistically insignificant.

So How Does it Really Work in Practice?







Dataset generated from IBMs synthetic association dataset generator.

Analyzing Stock Data- An Experiment:

- Gathered data on 103 stocks selected largely at random.
- Data corresponds to high, low, open, close, and volume for each stock over a two year period.
- The following indicators were used:
  - Each days trading range is quantized into three regions. A stock can transition from any of the three regions at opening to any of the three regions at close (9 possible combinations of which only one is true).
  - Average trading price is higher compared to previous days average.
  - Volume increases by greater than 2% compared to previous day.
  - Ratio of change in price and volume for current day is higher than previous day.

Analyzing Stock Data:

- In this way, each stock corresponds to 15 attributes for each day.
- Over a two year period, each stock corresponds to a single vector of dimension 7800 (15 attributes times 520 trading days).
- There are 103 such vectors forming rows of the matrix.

Compressing Stock Data:

Analyzing the vectors in our compression framework with varying error bounds (Hamming distances) resulted in groupings that were extremely intuitive. This is impressive considering that there is no prior knowledge of the stocks built into the analysis framework.

Some of the groupings are:

egrp, msft, sape, tecd, vcom, vsea

aklm, vrsa

coke, lnce

cost, naut, safc

elnk, ifmx, lcos, stmp

amzn, atvi, bosa, eftd, intc, mcicp, trid, vias, vshp, vtss

aapl, bnbn, dell, ebay, hits, ibm, mcaf, mqst, novl, psft

Notes on Compression of Stock Data:

- It is not about identifying stocks in the same sector; rather about identifying stocks that behave identically with respect to selected indicators.
- Grouping is only as good as the indicators.
- In addition to the groupings, we also get dominant group behavior. This can be analyzed to develop trading strategies.
- The compression framework also tells us which indicators match well within a group and which ones contribute to the most error. The latter can be dropped from further analysis.
- The indicators can be supplied to our framework; i.e., our framework is independent of the indicators. Investors can program their own portfolios and sets of indicators and work with them. It would be easy to provide a drop down menu with a choice of indicators for this purpose.

Current work:

- Taking the dominant group behavior vector and stacking it using a sliding time window of specified number of days, we can identify dominant behavior over specified number of days. For example, if the average price is down and the volume is up, it is followed by an average price up and volume up the next day. Different such rules can be determined for each group, and for a single group over varying time windows.
- Trading channels are determined by maxima and minima that occur over variable time-scales. This would require compressing and expanding time selectively. This is also possible in our compression framework by using time as one of the attributes.