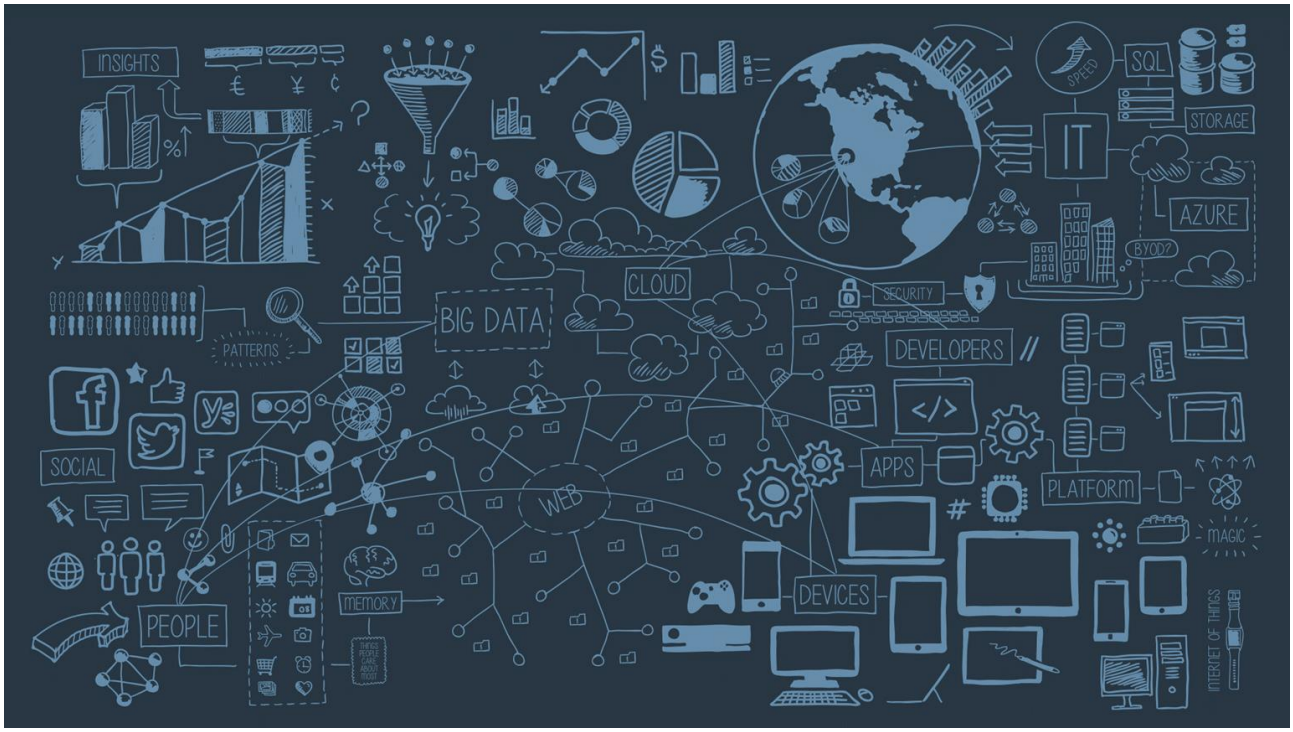


How Machine Learning

Applied To Big Data

WILL CHANGE

Business and Finance

Index

Executive Summary

What does the Data Revolution mean for investment management?

We believe:
the implications are significant. Big Data and Machine Learning can enable investment managers to see hidden connections and relationships between companies, including across industries.

We believe:
data analytics can enable investment managers to see these hidden relationships faster and sooner than market participants who are not similarly equipped-leading to a potential advantage in selecting investments.

1. Big Data and Machine Learning: Some Premises

The world has access to more data now than was conceivable even a decade ago. Businesses are accumulating new data faster than they can organize and make sense of it. They now have to figure out how to use this massive amount of data to make better decisions and sharpen their performance.

The new field of data science seeks to extract actionable knowledge from data, especially big data-extremely large data sets that can be analyzed to reveal patterns, trends, and associations. Data science extends from data collection and organization to analysis and insight, and ultimately to the practical implementation of what has been learned. This field intersects with all human activity and economics, finance, and business are no exception.

Data science brings the tools of machine learning, a type of artificial intelligence that gives computers the ability to learn without explicit programming (Samuel, 1959). These tools, coupled with vast quantities of data, have the potential to change the entire landscape of business management and economic policy analysis. Some of the changes offer much promise.

The rapid growth in the adoption of data science in business is no surprise, given the compelling economics of data science. In a competitive market, all buyers pay the same price, and the seller's revenue is equal to the price times the quantity sold. However, there are many buyers who are willing to pay more than the equilibrium price, and these buyers retain consumer surplus that can be extracted using big data for consumer profiling.

Charging consumers for different prices based on their analyzed profiles enables companies to get the highest price the consumer is willing and able to pay for a specific product. Optimizing price discrimination or market segmentation using big data is extremely profitable. This practice was the norm in some industries, for example, the airline industry, but is now being extended across the product spectrum.

Moreover, the gains from price targeting also enable firms to offer discounts to consumers who would not otherwise be able to afford the equilibrium price, thereby increasing revenue and expanding the customer base, and possibly social welfare.

Consumer profiling using big data is an important reason for the high valuations of firms such as Facebook, Google, and Acxiom, which offer products and services based on their customers' data. While big data may be used to exploit consumers, it is also changing business practices in a way that helps those same consumers. Firms are using the data generated from people's social media interactions to better understand their credit behavior.

Relating people's past credit history to their social media presence leads to improved credit-scoring systems. It also allows lenders to extend credit to people who might otherwise be turned down. In particular, big data eliminates the biases that arise when people make decisions based on limited information. This absence of fine-grained individual data led to redlining in loan applications, a practice dating to the 1930s. Mortgage lenders

would draw red lines around areas on a map to indicate that they would not make loans there because of the racial or ethnic composition. This stereotyping practice denied credit to entire segments of society. Big data, however, does away with stereotyping.

Coarse subjective data can now be replaced by finer, more individualized data. Credit scoring firms can exploit the heterogeneity detectable from people's social media interactions, texting streams, microblogs, credit card patterns, and profiling data—in addition to such typical demographic data as income, age, and location (Wei and others, 2014). The use of finer-grained data facilitates better classification of individuals by credit quality.

Economic forecasting has changed dramatically with data science methods. In traditional forecasting, key statistics about the economy—such as the quarterly GDP report are available only with considerable delay. Data science can get around these delays by relying on information that is reported more frequently, such as unemployment figures, industrial orders, or even news sentiment to predict those less frequently reported variables.

The collection of approaches engaged in this activity is known as "now-casting"—also termed the prediction of the present—but is better understood as real-time forecasting (see "The Queen of Numbers," in the March 2014 F&D).

Data science is also making inroads when it comes to analyze systemic financial risk. The world is more interconnected than ever, and measuring these ties promises new insight for economic decision making. Looking at systemic risk through the lens of networks is a powerful approach. Data scientists now use copious data to construct pictures of interactions among banks, insurance companies, brokers, and more.

It is obviously useful to know which banks are more connected than others. So it is about which banks have the most influence, computed using a method based on eigenvalues. Once these networks are constructed, data scientists can measure the degree of risk in a financial system, as well as the contribution of individual financial institutions to overall risk, offering regulators a new way of analyzing—and ultimately managing—systemic risk.

These approaches borrow extensively from the mathematics of social networks developed in sociology, and they are implemented on very large networks using advanced computer science models, culminating in a fruitful marriage of several academic disciplines.

It is now possible to rank a firm by quarterly earnings outcomes in its 10K, an annual report on a company's financial performance filed with the U.S. Securities and Exchange Commission (SEC). A tally of risk-

related words in quarterly reports offers an accurate ranking system for forecasting earnings. Firms whose quarterly reports are harder to read tend to have worse earnings—most likely because they attempt to report bad news using obfuscating language (see Loughran and McDonald, 2014).

Using an age-old metric for readability, the Gunning Fog Index, it is easy to score financial reports on this attribute, and regulators such as the Consumer Financial Protection Bureau are looking into establishing readability standards.

Studies have even found that the mere length of the quarterly report is sufficient to detect bad news (longer reports presage earnings declines), again because obfuscation is correlated with verbiage; as an ultimate extension, the file size alone of companies' filings uploaded to the SEC's website signaled quarterly earnings performance. Much more is expected to emerge from this rapidly evolving area of work.

Within this category, the analysis of news flow is especially interesting. Hedge funds mine thousands of news feeds a day to extract the top five or ten topics and then track the evolution of the proportion of topics from day to day to detect tradable shifts in market conditions. A similar analysis would be useful to policymakers and regulators, such as central bankers. For example, it might be time to revisit interest rate policy when the proportion of particular topics discussed in the news (such as inflation, exchange rates, or growth) changes abruptly.

Topic analysis begins with construction of a giant table of word frequencies, known as the "term-document matrix," that catalogs thousands of news articles. Terms (words) are the rows of the table, and each news article is a column. This huge matrix can uncover topics through mathematical analysis of the correlation between words and between documents.

Clusters of words are indexed and topics detected through the use of machine learning such as latent semantic indexing and Latent Dirichlet Allocation (LDA). LDA analysis produces a set of topics and lists of words that appear within these topics.

These modeling approaches are too technical to be discussed here, but they are really just statistical techniques that uncover the principal word groupings in a collection of documents (for example, in the news stream). These language clues are likely to be widely used by economic policymakers and in political decision making—for example, in redefining the message in a political campaign.

Computers are more powerful than ever, and their ability to process vast amounts of data has stimulated the field of artificial intelligence. A new class of algorithms known as "deep learning nets"—inspired by biological neural networks—has proved immensely powerful in mimicking how the brain works, offering many successful instances of artificial intelligence.

Deep learning is a statistical methodology that uses artificial neural networks to map a large number of input variables to output variables—that is, to identify patterns.

Information is dissected through a silicon-and-software based network of neurons. Data are used to strengthen the connections between these neurons, much as humans learn from experience over time.

The reasons for the stunning success of deep learning are twofold: the availability of huge amounts of data for machines to learn from and the exponential growth in computing power, driven by the development of special-purpose computer chips for deep-learning applications. Deep learning powers much of the modern technology the world is beginning to take for granted, such as machine translation, self-driving cars, and image recognition, and labeling.

This class of technology is likely to change economics and policy very soon. Credit rating agencies are already using it to generate reports without human intervention. Large deep-learning neural networks may soon provide forecasts and identify relationships between economic variables better than standard statistical methods.

It is hard to predict which domains in the dismal science will see the biggest growth in the use of machine learning, but this new age has definitely arrived. As noted science fiction writer William Gibson put it, “The future is already here; it’s just not very evenly distributed.”

2. Financial Markets: Implications

Today, there exists an enormous amount of data about every company—data which potentially can influence stock prices and other investment opportunities. As humans, we struggle to keep up with the torrent of information from hundreds or even thousands of sources. New data-analysis tools can help find signals in the noise.

The analysis of structured data—numerical data sets and tables—is well-understood and generally available. But the analysis of unstructured data—news stories, social media posts, and financial filings, to give a few examples—is where considerable energy is focused today.

The investment-research industry publishes hundreds of thousands of research reports each year—and a single investor generally can only read a limited number of them. Today, with more than 13,000 individual analysts active in the market, issuing an estimated 2.7 million monthly estimate revisions, deep and systematic analysis of these reports maybe even farther than ever beyond the power of any single, unassisted human being (or group of human beings).

We estimate that a person who reads about 200 words per minute he/she would need more than 865 hours to read through a single month's worth of reports; this means he or she would be reading continuously for more than an entire month, without sleep.

Here's an example. We believe investment research analysts may sometimes be reluctant to raise or lower a price target or rating too rapidly. Analysts may instead opt to reflect new views incrementally, by changing the tone and view of the text they write in their reports.

Can there be a possibility of predicting some changes to price targets and analyst ratings? We believe the answer is yes—and the explosion in data analytics has created ways potentially to identify these cases. One emerging application of data to sell-side analyst research is to use data analysis to spot a positive turn in the language an analyst uses before an estimate or ratings upgrade takes place.

We believe that identifying an analyst's evolving views prior to the release of higher ratings potentially can provide investors advantages in the decision to buy or sell a stock. In the reverse scenario, we believe analysts preparing to downgrade a stock sometimes may publish increasingly negative language.

By utilizing a wide database of research reports, trends can be identified, potentially before they affect price returns.

One example is the way that Big Data can be used to supplement or even supersede the study of supplier and customer relationships. Traditionally, a manager seeking to forecast a company's sales performance can watch suppliers for order surges, lags, or bottlenecks in search of insight into current-quarter or future sales.

The same manager can study consumer-sentiment indicators and same-store sales in attempt to understand sales prospects. Today, data has broadened the opportunity for information advantage substantially. Managers today can study hundreds or even thousands of other relationships, which create the potential for new insights into company prospects. For instance, investment managers can use data to identify "clusters" of companies which share overlooked relationships.

These relationships could center on the way their businesses are affected by a rise or fall in oil prices or a change in weather conditions.

They could concern the manner in which new regulations affect industries besides the ones mentioned in news headlines or the way that important implications for a firm in one country may be buried in a patent filing in another. Each of these examples entails the linkage of data which cannot be downloaded from standard market-data terminals.

We believe this "unstructured" data has the most potential to transform investment opportunities. Asset managers who master this unstructured data by investing in state-of-the-art technology and deploying the most extensive data sources potentially can enjoy a competitive advantage over their peers.

3. Financial Markets: The Approach

Definition gave by McKinsey Global Institute in 2011:

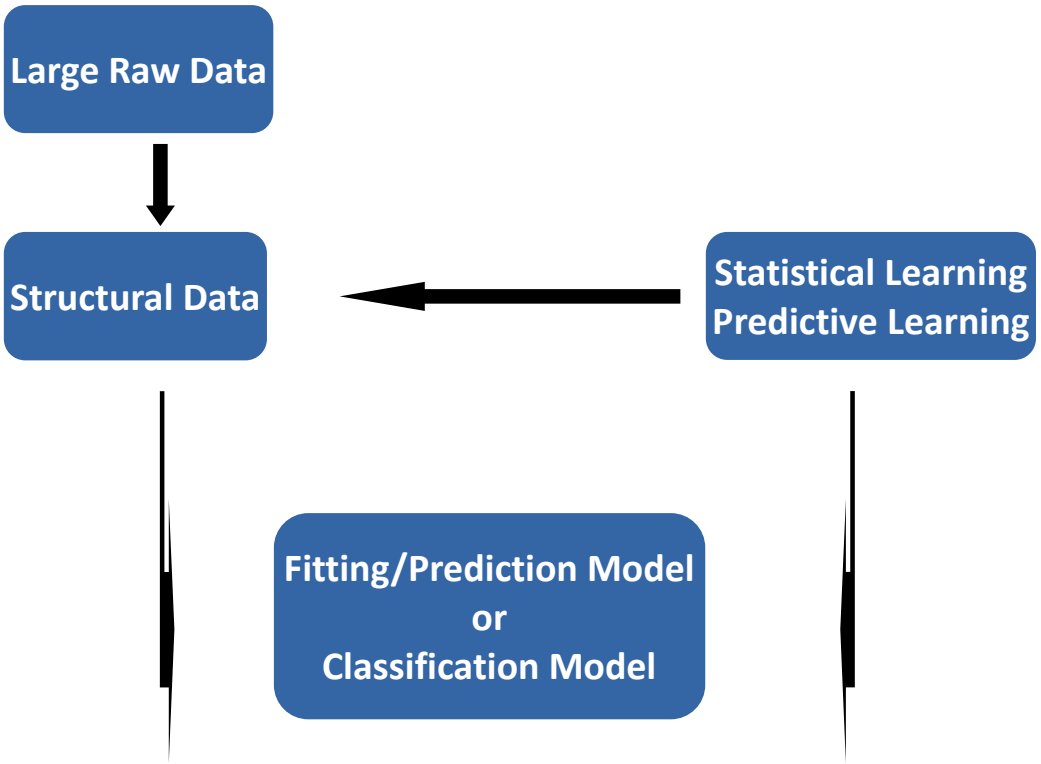
"Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze."

That's simple definition as quickly grow-up some aspects:

- Large dataset (Megabyte, Gigabyte, Terabyte, Petabyte, Exabyte)
- Unstructured data (networked data but fuzzy relationships)
- Data-driven research, business & decision
- High skills (IT, statistics, etc…)

The natural conclusion has been that with big data, there are certainly few big data problems and that these problems differ from one sector to another.

Chart 1



A possible rationalization of the problem is the Gartner 3V model of big data that fragment the data in 3/4 macro-areas:

1. VOLUME (as amount of data).
2. VELOCITY (as speed of data).
3. VARIETY (as data types)
4. VERACITY AND VALUE

Follow that scheme, and the first important question will be: How to transform raw data into structured (informative) data? That set of Heterogeneous variables should be transformed to comparable and workable data, new variables, valuable variables, and model variables.

Chart 2

The most difficult step is transforming X into Y:

- *Averaging, Averaging², Averaging³, …*
- Cutting X_1 into Y_1, Y_2
- Aggregating X_1, X_1, etc i.nto Y_1
- Creating class from X_1, X_2, etc
- Conditioning X_1, by X_2, etc
- Dummy variables everywhere

The Y variables are more important than the model g itself.

Another important consideration to do is who manage and work data today? During the entire period from 1970 to 1990, data has been managed by statisticians; projects were light and simple. In the last twenty years until 2010, data has been managed by information technology people.

During this time, projects were really heavy. Right know who manage the data? The Scientist. This people work in computer science, modeling statistic, and analytics.

4. Econometrics vs Machine Learning

The first thing to do is to analyze principal differences between a standard approach (econometrics) versus the new machine learning approach. The main considerations are about the pillars of this two approaches. If we look in the econometric models, we discover three principal milestones.

The first one is the starting line, and econometrics are based principally on economic theory; from this next pillars, the next logical consequences are the use of parametric models for analysis using statistical inference. These milestones are completely different in machine learning approach.

The first one pillar are data, and features indicate a more pragmatic method. Natural consequences are the use of non-parametric models with a cross-validation analysis system. As clear, the statistical tools are different:

Econometrics	Machine Learning
Linear Regression, Maximum Likelihood & GMM	Shrinkage Regression (Ridge, Lasso, Lars, Elastic Net, Spike, Slab)
Logic, Probit, Tobit, etc	Ensemble Learning (Boosting, Bagging)
ARMA, ARIMA, VaR, Cointegration, VECM, ARCH, GARCH	Random Forest, Neural Nets, SVM, Deep Learning

In the following table, a view "ensemble" of methods used in asset management. We have analyzed some algorithms, and we have tested it on different asset management working areas.

Table 1

	Bond Scoring	Stock Picking	Trend Filtering	Mean-Reverting	Index Tracking	High Freq. Tracking	Stock Classific.	Technical Analysis
Lasso		■	■	■	■	■		
NMF							■	■
Boosting	■	■				■		
Bagging	■	■				■		
Rnd. Forest	■			■				■
Neural Nets	■					■		■
SVM	■	■	■		■		■	■
Sparse Kalman					■	■		
K-NN	■							
K-Means	■						■	
Testing Protocol	■	■	■	■	■	■		■

5. Asset Management: Some Models

5.1 How Machine Learning and Hierarchical Risk Parity create Asset with RISK LESS

In this Machine Learning (ML) example, we introduce the Hierarchical Risk Parity (HRP) approach. HRP portfolios address three major concerns of quadratic optimizers in general and Markowitz's critical line algorithm (CLA) in particular: Instability, concentration, and underperformance.

HRP applies modern mathematics (graph theory and machine learning techniques) to build a diversified portfolio based on the information contained in the covariance matrix. However, unlike quadratic optimizers, HRP does not require the convertibility of the covariance matrix. In fact, HRP can compute a portfolio on an ill-degenerated or even a singular covariance matrix, an impossible feat for quadratic optimizers.

Monte Carlo experiments show that HRP delivers lower out-of-sample variance than CLA, even though minimum-variance is CLA's optimization objective. HRP also produces less risky portfolios out-of-sample compared to traditional risk parity methods.

Portfolio construction is perhaps the most recurrent financial problem. On a daily basis, investment managers must build portfolios that incorporate their views and forecasts on risks and returns. This is the primordial question that a 24 years old Harry Markowitz attempted to answer more than 6 decades ago.

His monumental insight was to recognize that various levels of risk are associated with different optimal portfolios in terms of risk-adjusted returns, hence the notion of "efficient frontier" (Markowitz, 1952). One implication was that it is rarely optimal to allocate all assets to the investments with highest expected returns. Instead, we should take into account the correlations across alternative investments in order to build a diversified portfolio.

Before earning his Ph.D. in 1954, Markowitz left academia to work for the RAND Corporation, where he developed the Critical Line Algorithm (CLA).

CLA is a quadratic optimization procedure specifically designed for inequality-constrained portfolio optimization problems. This algorithm is notable in that it guarantees that the exact solution is found after a known number of iterations and that it ingeniously circumvents the Karush-Kuhn-Tucker conditions (Kuhn and Tucker, 1951).

A description and open-source implementation of this algorithm can be found in Bailey and López de Prado [2013].

Surprisingly, most financial practitioners still seem unaware of CLA, as they often rely on generic-purpose quadratic programming methods that do not guarantee the correct solution or a stopping time.

Despite of the brilliance of Markowitz's theory, a number of practical problems make CLA solutions somewhat unreliable. A major caveat is that small deviations in the forecasted returns will cause CLA to produce very different portfolios (Michaud, 1998). Given that returns can rarely be forecasted with sufficient accuracy, many authors have opted for dropping them all together and focus on the covariance matrix.

This has led to risk-based asset allocation approaches, of which "risk parity" is a prominent example (Jurczenko, 2015).

Dropping the forecasts on returns improves; however, does not prevent the instability issues. The reason is, quadratic programming methods require the inversion of a positive-definite covariance matrix (all eigenvalues must be positive).

This inversion is prone to large errors when the covariance matrix is numerically ill-conditioned, i.e., it has a high condition number (Bailey and López de Prado [2012]).

5.2 Standard Model: Markowitz' Curse

The condition number of a covariance, correlation (or normal, thus diagonalizable) matrix is the absolute value of the ratio between its maximal and minimal (by moduli) eigenvalues. This number is lowest for a diagonal correlation matrix, which is its own inverse.

As we add correlated (multi-co-linear) investments, the condition number grows. At some point, the condition number is so high that numerical errors make the inverse matrix too unstable. A small change on any entry will lead to a very different inverse.

This is Markowitz' curse: The more correlated the investments, the greater the need for diversification, and yet the more likely we will receive unstable solutions. The benefits of diversification often are more than offset by estimation errors. Increasing the size of the covariance matrix will only make matters worse, as each covariance coefficient is estimated with fewer degrees of freedom.

In general, we need at least $12(N+1)$ independent and identically distributed (IID) observations in order to estimate a covariance matrix of size N that is not singular. For example, estimating an invertible covariance matrix of size 50 requires at the very least 5 years of daily IID data. As most investors know, correlation structures do not remain invariant over such long periods by any reasonable confidence level.

The severity of these challenges is optimized by the fact that even naïve (equally weighted) portfolios have been shown to beat mean-variance and risk-based optimization out-of-sample (De Miguel et al., 2009).

5.3 Machine Learning Model: Hierarchical Relationships

These instability concerns have received substantial attention in recent years, as Kolm et al. [2013] have carefully documented. Most alternatives attempt to achieve robustness by incorporating additional constraints (Clarke et al., 2002), introducing Bayesian priors (Black and Litterman, 1992) or improving the numerical stability of the covariance matrix's inverse (Ledoit and Wolf [2003]).

All the methods discussed so far, although published in recent years, are derived from (very) classical areas of mathematics: Geometry, linear algebra, and calculus.

A correlation matrix is a linear algebra object that measures the cosines of the angles between any two vectors in the vector space formed by the returns series (see Calkin and López de Prado [2014a, 2015b]).

One reason for the instability of quadratic optimizers is that the vector space is modeled as a complete (fully connected) graph, where every node is a potential candidate to substitute another.

In algorithmic terms, inverting the matrix means evaluating the partial correlations across the complete graph. Small estimation errors are magnified, leading to incorrect solutions. Intuitively it would be desirable to drop unnecessary edges.

Let's consider for a moment the practical implications of such topological structure. Suppose that an investor wishes to build a diversified portfolio of securities, including hundreds of stocks, bonds, hedge funds, real estate, private placements, etc.

Some investments seem closer substitutes of one another, and other investments seem complementary to one another. For example, stocks could be grouped in terms of liquidity, size, industry, and region, where stocks within a given group compete for allocations.

In deciding the allocation to a large publicly-traded U.S. financial stock like J.P. Morgan, we will consider adding or reducing the allocation to another large publicly-traded U.S. bank like Goldman Sachs, rather than a small community bank in Switzerland, or a real estate holding in the Caribbean.

And yet, to a correlation matrix, all investments are potential substitutes to each other. In other words, correlation matrices lack the notion of hierarchy. This lack of hierarchical structure allows weights to vary freely in unintended ways, which is a root cause of CLA's instability.

Thinking to visualize a hierarchical structure known as a tree we can introduce two desirable features: a) It has only N-1 edges to connect N nodes, so the weights only rebalance among peers at various hierarchical levels; and b) the weights are distributed top-down, consistent with how many asset managers build their portfolios, e.g., from asset class to sectors to individual securities.

For these reasons, hierarchical structures are designed to give not only stable but also intuitive results.

In this example, we have studied a new portfolio construction method that addresses CLA's pitfalls using modern mathematics: Graph theory and machine learning. This Hierarchical Portfolio Construction (HRP) method uses the information contained in the covariance matrix without requiring its inversion or positive-definitiveness.

In fact, HRP can compute a portfolio based on a singular covariance matrix, an impossible feat for quadratic optimizers. The algorithm operates in three stages: Tree clustering, quasidiagonalization, and recursive bisection.

a. Tree Clustering

Consider a *T*N* matrix of observations *X*, such as returns series of *N* variables over T periods. We would like to combine these *N* column-vectors into a hierarchical structure of clusters so that allocations can flow downstream through a tree graph.

1. We compute a *N*N* correlation matrix with entries $p = \{pi,j\} i,j = 1,\ldots,N$, where $pi,j = p\{Xi,Xj\}$
   a) We define the distance measure: $d\colon (Xi,Xj) \subset B \to \mathbb{R} \in [0,1], di,j = d[Xi,Xj] = \sqrt{12(1-pi,j)}$
   b) This allow us to compute a *N*N* distance matrix $D = \{di,j\} i,j = 1,\ldots,N$. Matrix $D$is a proper metric space, in the sense of:
      i. Non-Negativity $d[X,Y] \geq 0$
      ii. Coincidence $d[X,Y] = 0 \Leftrightarrow X = Y$
      iii. Symmetry $d[X,Y] = d[X,Y]$
      iv. Sub-Additivity $d[X,Z] \leq d[X,Y] + d[Y,Z]$
2. We compute the Euclidean distance between any two column-vectors of D, $\vec{d}\colon (Di,Dj) \subset B \to \mathbb{R} \in [0,\sqrt{N}], \overrightarrow{di},j = \vec{d}[Di,Dj] = \sqrt{\sum(dn,i-dn,j)2Nn} = 1$. Note the difference between distance metrics $di,$ and $\overrightarrow{di},j$. Whereas $di$is defined on column-vectors

of X, $\overrightarrow{di},j$is defined of column-vectors of D(a distance of distances). Therefore, $\vec{d}$is a distance defined over the entire metric space D, as each $\overrightarrow{di},$ is a function of the entire correlation matrix (rather than a particular cross-correlation pair).

3. We cluster together the pair of columns (*i**,*) such that (*i**,*j**) = argmin$(i,j) \neq j\{\overrightarrow{di},j\}$, and denote this cluster as $\mu[1]$.
4. W need to define the distance between a newly formed cluster $[1]$and the single (un-clustered) items, so that $\{\overrightarrow{di},j\}$may be updated. In hierarchical clustering analysis, this is known as the "linkage criterion." For example, we can define the instance between an item $i$of $\vec{d}$and the new cluster $\mu[1]$as $\overrightarrow{di}\mu[1] = min[\{\overrightarrow{di},j\}j \in \mu[1]]$(the nearest point algorithm).
5. Matrix $\{\overrightarrow{di}\}$is updated by appending $\overrightarrow{di},$ $\mu[1]$, and dropping the clustered columns and rows $j \in \mu[1]$.

6. Applied recursively, steps 3-5 allow us to append $N-1$such clusters to matrix D, at which point t, the final cluster contains all of the original items and clustering algorithm stops.

This stage allow us to define the linkage matrix as a $(N-1)x4$ matrix with structure $Y=\{(ym,1ym,2,ym,3ym,4)\}m=1,\ldots,N-1$i.e.with one 4-tuple per cluster. Items $(ym,1,ym,2)$report the constituents. Item $ym,3$report the distance between $ym,1$and $ym,2$tath is $ym,3=\overrightarrow{dym},1,ym,2$. Item $ym,3\leq N$report the number of original items included in cluster m.

b. Quasi-Diagonalization

This stage reorganizes the rows and columns of the covariance matrix so that the largest values lie along the diagonal. This quasi-diagonalization of the covariance matrix (without requiring a change of basis) renders a useful property.

Similar investments are placed together, and dissimilar investments are placed far apart. The algorithm works as follows: We know that each row of the linkage matrix merges two branches into one. We replace clusters in $(yN-1,1,yN-1,2)$with their constituents recursively, until no clusters remain.

These replacements preserve the order of the clustering. The output is a sorted list of original (un-clustered) items.

b. Recursive Bisection

This stage has delivered a quasi-diagonal matrix. The inverse-variance allocation is optimal for a diagonal covariance matrix. We can take advantage of these facts in two different ways: a) Bottom-up, to define the

variance of a continuous subset as the variance of an inverse-variance allocation; b) top-down, to split allocations between adjacent subsets in inverse proportion to their aggregated variances.

The following algorithm formalizes this idea:

1. The algorithm is initialized by:
   a. Setting the list of items: $L=\{L0\}$with $L0\{n\}n=1,\ldots,N$
   b. Assigning a unit weight to all items : $wn=1,\forall n=1,\ldots,N$
2. If $|Li|=1,\forall Li\in L$, then stop
3. For each $Li\in L$such that $|Li|>1$:
   a. Bisect Liinto two subsets, $Li(1)\cup Li(2)=Li$where $|Li(1)|=\int[12|Li|]$, and the order is preserved
   b. define the variance of $Li(j),j=1,2$, as the quadratic form $\nabla i(j)\equiv\widetilde{wi}(\grave{j})Vi(j)\widetilde{wi}(j)$where $Vi(j)$is the covariance matrix between the constituents of the $Li(j)$bisection, and $\widetilde{wi}(j)=$

$diag[Vi(j)] - 11tr[diag[Vi(j)] - 1]$, were $diag[.]$ and $tr[.]$ are the diagonal and trace operators

c. Compute the split factor: $\alpha i = 1 - \nabla i(1)\nabla i(1) + \nabla i(2)$so that $0 \leq \alpha i \leq 1$

d. Re-Scale allocations $wn$by a factor of $\alpha i, \forall n \in Li(1)$

e. Re-Scale allocation $wn$by a factor of $(1 - \alpha i), \forall n \in Li(2)$

4. Loop to stop 2

Step 3.b takes advantage of the quasi-diagonalization bottom-up because it defines the variance of the partition using $L(j)$using inverse-variance weightings $\widetilde{wi}(j)$.

Step 3.c takes advantage of the quasi-diagonalization top-down because it splits the weight in inverse proportion to the cluster's variance. This algorithm guarantees that $0 \leq wi \leq 1, \forall i = 1, \ldots$, and $\sum wiNi = 1 = 1$, because at each iteration we are splitting the weights received from higher hierarchical levels. Constraints can be easily introduced in this stage, by replacing the equations in steps 3.c-3.e according to the user's preferences.

This concludes a first description of the HRP algorithm, which solves the allocation problem in deterministic logarithmic time $(n) = 0(\log 2n)$. Next, we will put to practice what we have learned and evaluate the method's accuracy out of sample.

5.4 From Standards to Machine Learning H.R.P: A Numerical Example

We begin by simulating a matrix of observations X, of order (10000x10). On this random data, we compute HRP's allocations, and compare them to the allocations from two competing methodologies: 1)

Quadratic optimization, as represented by CLA's minimum-variance portfolio (the only portfolio of the efficient frontier that does not depend on returns' means); and 2) traditional risk parity, exemplified by the Inverse-Variance Portfolio (IVP). We apply the standard constraints that $0 \leq wi \leq 1$(non-negativity), $\forall i = 1, \ldots$,and $\sum wiNi = 1 = 1$(full investment).

Incidentally, the condition number for the covariance matrix in this example is only 150.9324, not particularly high, and therefore not unfavorable to CLA.

A characteristic outcome of the three methods studied: CLA concentrates weights on a few investments, hence becoming exposed to idiosyncratic shocks. IVP evenly spreads weights through all investments, ignoring the correlation structure.

This makes it vulnerable to systemic shocks. HRP finds a compromise between diversifying across all investments and diversifying across cluster, which makes it more resilient against both types of shocks.

From the allocations in Table2, we can appreciate a few stylized features: First, CLA concentrates 92.66% of the allocation on the top 5 holdings, while HRP concentrates only 62.57%.

Second, CLA assigns zero weight to 3 investments (without the $0 \leq wi$constraint, the allocation would have been negative).

Third, HRP seems to find a compromise between CLA's concentrated solution and traditional risk parity's IVP allocation.

What drives CLA's extreme concentration is its goal of minimizing the portfolio's risk. And yet both portfolios have a very similar standard deviation $(\sigma HRP = 0{,}4640, \sigma CLA = 0{,}4486)$. So CLA has discarded half of the investment universe in favor of a minor risk reduction.

The reality, of course, is, CLA's portfolio is deceitfully diversified because any distress situation affecting the five top allocations will have a much greater negative impact on CLA's than HRP's portfolio.

Table 2

Weight	CLA	HRP	IVP
1	14,44%	7,00%	10,36%
2	19,93%	7,59%	10,28%
3	19,73%	10,84%	10,36%
4	19,87%	19,03%	10,25%
5	18,68%	9,72%	10,31%
6	0,00%	10,19%	9,74%
7	5,86%	6,62%	9,80%
8	1,49%	9,10%	9,65%
9	0,00%	7,12%	9,64%
10	0,00%	12,79%	9,61%

Out-of-Sample Monte-Carlo Simulation (Standard Approach)

In our numerical example, CLA's portfolio has lower risk than HRP's in-sample. However, the portfolio with minimum variance in-sample is not necessarily the one with minimum variance out-of-sample.

It would be all too easy for us to pick a particular historical dataset where HRP outperforms CLA and IVP (for a discussion on overfitting and selection bias, see Bailey and López de Prado [2015]).

Instead, in this section, we evaluate via Monte Carlo the performance out-of-sample of HRP against CLA's minimum-variance and traditional risk parity's IVP allocations.

This will also help us understand what features make a method preferable to the rest, regardless of anecdotal counter-examples.

First, we generate 10 series of random Gaussian returns (520 observations, equivalent to 2 years of daily history), with 0 mean and an arbitrary standard deviation of 10%. Real prices exhibit frequent jumps (Merton, 1976), and returns are not cross-sectional independent, so we must add random shocks and a random correlation structure to our generated data.

Second, we compute HRP, CLA, and IVP portfolios by looking back at 260 observations (a year of daily history). These portfolios are re-estimated and rebalanced every 22 observations (equivalent to a monthly frequency).

Third, we compute the out-of-sample returns associated with those three portfolios. This procedure is repeated 10,000 times.

All mean portfolio returns out-of-sample are essentially 0, as expected. The critical difference comes from the variance of the out-of-sample portfolio returns: $(\sigma HRP2 = 0{,}0671\sigma CLA2 = 0{,}1157\sigma IVP2 = 0{,}0928)$. Although CLA's goal is to deliver the lowest variance (that is the objective of its optimization program), its performance happens to exhibit the highest variance out-of-sample, and 72.47% greater variance than HRP's.

In other words, HRP would improve the out-of-sample Sharpe ratio of a CLA strategy by about 31.3%, a rather significant boost. Assuming that the covariance matrix is diagonal brings some stability to the IVP. However, its variance is still 38.24% greater than HRP's.

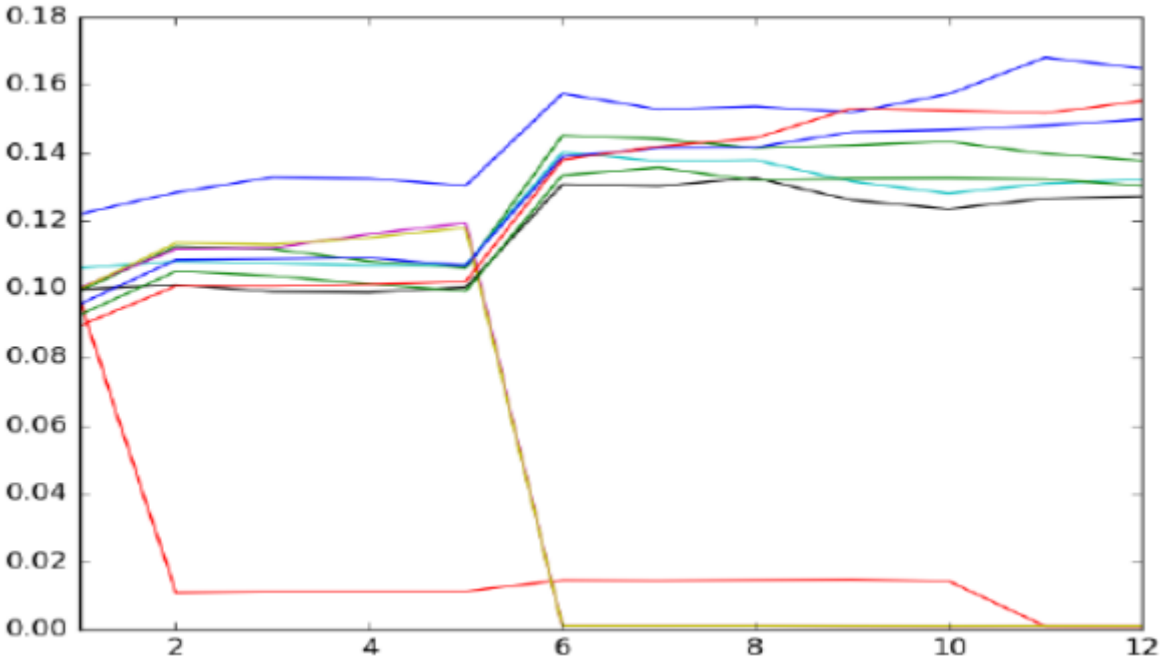
This variance reduction out-of-sample is critically important to risk parity investors, given their use of substantial leverage. See Bailey et al. [2014] for a broader discussion of in-sample vs. out-of-sample performance.

The mathematical proof for HRP's outperformance over Markowitz's CLA and traditional risk parity's IVP is somewhat involved and beyond the scope of this introductory paper. In intuitive terms, we can understand the above empirical results as follows: Shocks affecting a specific investment penalize CLA's concentration.

Shocks involving several correlated investments penalize IVP's ignorance of the correlation structure. HRP provides better protection against both, common and idiosyncratic shocks, by finding a compromise between diversification across all investments and diversification across clusters of investments at multiple hierarchical levels.

In the following charts, we plots the time series of allocations for the first of the 10,000 runs.

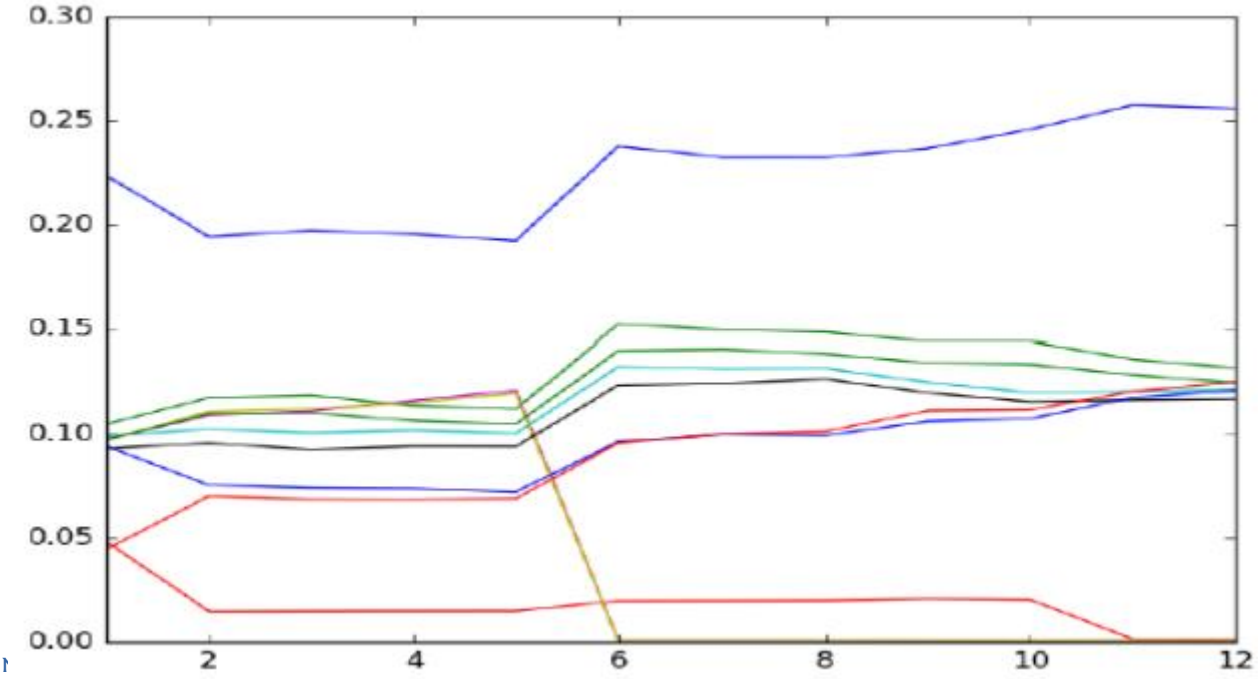
Chart 3



Between the first and second rebalance, one investment receives an idiosyncratic shock, which increases its variance. IVP's response is to reduce the allocation to that investment and spread that former exposure across all other investments.

Between the fifth and sixth rebalance, two investments are affected by a common shock. IVP's response is the same. As a result, allocations among the seven unaffected investments grow over time, regardless of their correlation.

Chart 4

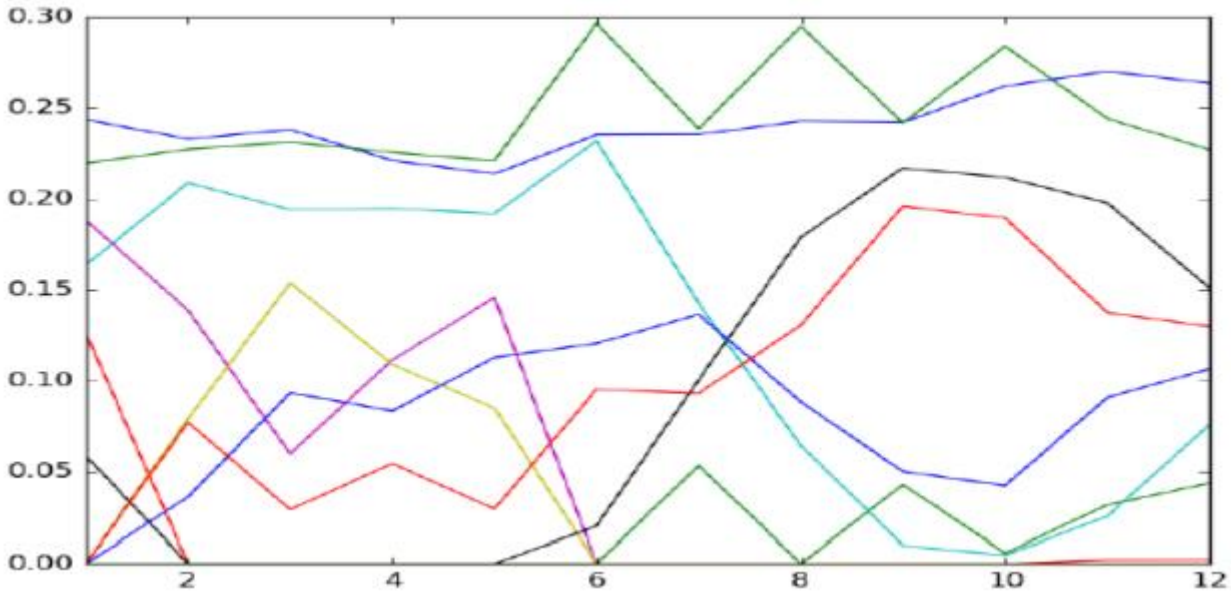


HRP's response to the idiosyncratic shock is to reduce the allocation to the affected investment and use that reduced amount to increase the allocation to a correlated investment that was unaffected.

As a response to the common shock, HRP reduces allocation to the affected investments and increases allocation to uncorrelated ones (with lower variance).

CLA allocations respond erratically to idiosyncratic and common shocks. If we had taken into account rebalancing costs, CLA's performance would have been very negative.

Chart 5



5.5 Machine Learning and SVM to Create Provisional Financial Market Direction

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Many stock prediction studies focus on using macroeconomic indicators, such as CPI and GDP, to train the prediction model.

However, daily data of the macroeconomic indicators are almost impossible to obtain. Thus, those methods are difficult to be employed in practice. In this paper, we propose a method that directly uses prices data to predict market index direction and stock price direction.

The emergence of machine learning and artificial intelligence algorithms has made it possible to tackle computationally demanding mathematical models in stock price direction prediction. Frequently adopted methods include artificial neural networks (ANNs), Bayesian networks, and support vector machine (SVM).

Amongst them, ANNs have drawn significant interests from several researchers in the stock price forecasting in the past decades.

The ANNs are robust in model specification compared to parametric models, which makes it frequently applied in forecasting stock prices and financial derivatives. Guresen et al. (2011) reported the validity of ANNs in stock market index prediction.

Grudnitski and Osburn (1993) applied ANNs to predict gold futures prices. The drawback of the price prediction is that the price is highly volatile so as to result in large regression errors. Compared to the price

prediction, the stock direction prediction is less complex and more accurate.

The stock direction prediction has been recently addressed in several research articles, which consider different variants of ANNs (Saad et al., 1998). However, one drawback of ANNs is that the efficiency of predicting unexplored samples decreases rapidly when the neural network model is too over-fitted by available observations.

In other words, the noisy stock information may lead ANNs to a complex model, which might result in the over-fitting problem.

The predominant methods in the stock market direction prediction are the approaches based on SVM. Since the SVM implements the structural risk minimization principle, it often achieves better generalization performance and lower risk of overfitting than the ANNs (Cortes and Vapnik, 1995).

A major drawback of SVM for the direction prediction is that the input variables lie in a high-dimensional feature space, ranging from hundreds to thousands. The storage of the variables requires a lot of memory and computation time.

Specifically, a stock market consists of several hundreds of stocks, which leads to the high dimensionality of the variables. Therefore, it is of considerable importance to conduct dimension reduction to acquire an efficient and discriminative representation before classification. Under the dimensionality reduction, curse of dimensionality could be effectively managed (Cortes and Vapnik, 1995).

A common unsupervised feature extraction method is principal component analysis (PCA) (Pearson, 1901) by which principal components are obtained through the manipulation of original data. The PCA has been widely used to deal with high dimensional data sets in many areas, such as protein dynamics reduction, spectral data reduction, and face patterns reduction.

Interestingly, the adaptation of the PCA feature selection to stock prices data analysis is rarely found, to the best of our knowledge.

In stock prices data, there exists a common phenomenon that is called co-movement between stocks due to the institutional investors¡¯ common ownership of subsets of stocks in their portfolios (Pindyck and Rotemberg, 1993).

Shiller (1989) showed the co-movements of returns between the USA and UK markets using simple regression tests.

As a matter of fact, the co-movement exits not only between stocks in a domestic market (internal) but also between two tightly connected stock markets (external). This fact stimulate us to consider both internal and external factors for predicting individual stocks and market index directions.

The macroeconomic indicators [such as consumer price index (CPI), gross national product (GNP), and gross domestic product (GDP)] may be high internal impact factors for the prediction. However, daily data of those macroeconomic factors are impossible to obtain and analyze in reality. For simplicity and generality, we only handle stock prices data, which are timely and easy to access. As for the external factors, we take daily S&P 500 index values and exchange rates (EXR) into account.

Both factors can be obtained and managed easily. Thus, the method in this sample secondly contributes to the stock prediction in practical aspect compared with most of the state-of-the-art approaches.

Supporting Vector Machine: The Model

The structure of the proposed model is shown below. Let $x_i \in \mathbb{R}$denote a column vector of the daily rates of return of stock $i, i = 1, \ldots, n$wich is obtained from $p$daily market observation. The matrix $X = (x_i)^t$can be reduced to the principal component matrix $Y = (x_k)^t, K = 1, \ldots, m, m \ll n$by minimizing the variance of the linear transformation of X. Define the contribution rate of the k^{th}principal component as:

$$\frac{\lambda_k}{\sum_{i=1}^{n} \lambda_t}$$

where $\lambda_t$represents the variance of y_k. The cumulative rate of the first $m$principal components is:

$$\frac{\sum_{i=1}^{m} \lambda_t}{\sum_{i=1}^{n} \lambda_t}$$

Along with these principal components, internal factors and external factors F = (F1, F2, …, Fh)T are utilized as input data, i.e., {Y, F}. Considering the co-movement property in a market, we find that the co-moved stocks are informative as internal factors.

Besides, since the market index itself is a beacon of the domestic economy and tread, it is also informative for forecasting.

The input to SVM is the data set $R = (rj, wi)$ where $rj = (yj, Fj)(j = 1, \ldots, p)$ is a row vector denoting jth daily data in p observation days. $wj \in 0,1$ Is a binary variable that represents the upward or downward direction of the stock market movement of the jthday.

The downward direction is represented by 0 and the upward by 1. The input data is carefully divided into two parts, i.e., training data and testing data. As addressed by financial analysis recently, the data periods in most computer technique related articles are selected limitedly.

In order to refrain from limited sample selection, the training data and testing data are goes parallel using rolling windows of ten years data to ensure that the predictions are made using all the information available at that time.

Besides, unlike several studies testing in-sample data, we compute the one-day-ahead predictions, i.e., out-sample data. The details of data periods and how they are divided into training and testing data are explained in the next section.

The training data is utilized to acquire a classifier by training SVM. The classifier function of stock movement directions is defined as:

$$dir = f(r) = sgn(\sum_{j=1}^{p} w_j \alpha_j^{\overline{ts}} r_j^T + v^{\overline{ts}})$$

where $\alpha^{\overline{ts}}$ and $v^{\overline{ts}}$ are optimal values of Lagrange multipliers and intercepts of the corresponding hyperplanes, respectively.

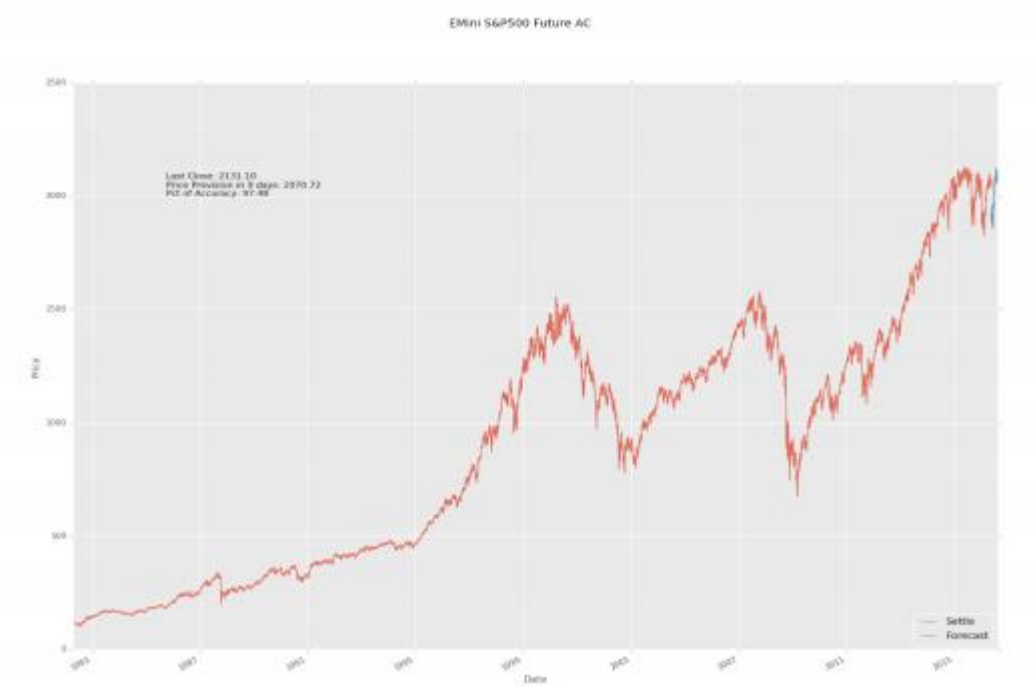
By introducing kernel tricks, the non-linear decision function for a stock direction prediction becomes:

$$dir = f(r) = sgn(\sum_{j=1}^{p} w_j \alpha_j^{\overline{ts}} K(r_j, r) + v^{\overline{ts}})$$

The testing data is used to test the model according to the classifier f(r). In reality, the training model can be designed to update real-timely so as to make full use of the present information.

Supporting Vector Machine: The Procedure and Result

Chart 5



References

Bailey, D., and M. López de Prado. "Balanced Baskets: A new approach to Trading and Hedging Risks." Journal of Investment Strategies, Vol. 1, No. 4 (2012), pp. 21-62. Available at http://ssrn.com/abstract=2066170.

Bailey, D., and M. López de Prado. "An open-source implementation of the critical-line algorithm for portfolio optimization." Algorithms, Vol. 6, No. 1 (2013), pp. 169-196. Available at http://ssrn.com/abstract=2197616.

Bailey, D., J. Borwein, M. López de Prado, and J. Zhu. "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-Of-Sample Performance." Notices of the American Mathematical Society, Vol. 61, No. 5 (2014), pp. 458-471. Available at http://ssrn.com/abstract=2308659.

Bailey, D., and M. López de Prado. "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting, and NonNormality." Journal of Portfolio Management, Vol. 40, No. 5 (2014), pp. 94-107

Black, F., and R. Litterman. "Global portfolio optimization." Financial Analysts Journal, Vol. 48 (1992), pp. 28–43.

Brualdi, R. The Mutually Beneficial Relationship of Graphs and Matrices. Conference Board of the Mathematical Sciences, Regional Conference Series in Mathematics, Nr. 115, (2010).

Calkin, N., and M. López de Prado. "Stochastic Flow Diagrams." Algorithmic Finance, Vol. 3, No. 1 (2014), pp. 21-42. Available at http://ssrn.com/abstract=2379314.

Calkin, N., and M. López de Prado. "The Topology of Macro-Financial Flows An Application of Stochastic Flow Diagrams." Algorithmic Finance, Vol. 3, No. 1 (2014), pp. 43-85. Available at http://ssrn.com/abstract=2379319.

Clarke, R., H. De Silva, and S. Thorley. "Portfolio constraints and the fundamental law of active management." Financial Analysts Journal, Vol. 58 (2002), pp. 48–66.

The Economist. Data, Data Everywhere. February 2010.

McAfee A., Brynjolfsson E. Big Data: The Management Revolution. Harvard Business Review, 2012.

McKinsey Global Institute. Big Data: The Next Frontier for Innovation, Competition, and Productivity. 2011.

McKinsey Global Institute. Disruptive Technologies: Advances that will Transform Life, Business, and the Global Economy. 2013.

Breiman L. Bagging Predictors. Machine Learning, 24, 1996.

Efron B., Hastie T., Johnstone I., Tibshirani R. Least Angle Regression. Annals of Statistics, 32(2), 2004.

Freund Y. Boosting a Weak Learning Algorithm by Majority. Information and Computation, 121(2), 1995.

Freund, Y. Shapiro R.E. Experiments with a New Boosting Algorithm. Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kaufman, 1996.

Friedman J.H. Multivariate Adaptive Regression Splines. Annals of Statistics, 19(1), 1991.

Friedman J.H., Hastie T., Tibshirani R. Additive Logistic Regression: A Statistical View of Boosting. Annals of Statistics, 28(2), 2000.

Guyon I., Elisseeff A. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 2003.

Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Second edition, Springer, 2009.

Lee D.D., Seung H.S. Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, 401, 1999.

Shapiro R.E. The Strength of Weak Learnability. Machine Learning, 5(2), 1990.

Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society B, 58(1), 1996.

Tropp J.A., Gilbert A.C. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. IEEE Transactions on Information Theory, 53(12), 2007.

Vapnik V. Statistical Learning Theory. John Wiley and Sons, 1998.

Zou H., Hastie T., Tibshirani R. On the "Degrees of Freedom" of the Lasso. Annals of Statistics, 35(5), 2007.

Cochrane J.H. (2011). Presidential Address: Discount Rates. Journal of Finance, 66(4), pp. 1047-1108.

Harvey C.R., Liu Y. and Zhu H. (2014). . . . and the Cross-Section of Expected Returns. SSRN, www.ssrn.com/abstract=2249314.

Novy-Marx, R. (2014). Predicting Anomaly Performance with Politics, The Weather, Global Warming, Sunspots, and The Stars. Journal of Financial Economics, 112(2), pp. 137-146.

Cazalet Z. and Roncalli, T. (2014). Facts and Fantasies About Factor Investing. Lyxor Research Paper, 112 pages.

Belloni A., Chen D., Chernozhukov V., Hansen C. Sparse Models and Methods for Optimal Instruments with An Application to Eminent Domain. Econometrica, 80(6), 2012.

DeMiguel V., Garlappi L., Nogales F.J., Uppal R. A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms. Management Science, 55(5), 2009.

Fan J., Zhang J., Yu K. Vast Portfolio Selection with Gross-exposure Constraints. Journal of the American Statistical Association, 107(498), 2012.

Giamouridis D., Paterlini S. Regular(sized) Hedge Fund Clones. Journal of Financial Research, 33(3), 2010.

Kalaba R., Tesfatsion L. Time-varying Linear Regression via Flexible Least Squares. Computers & Mathematics with Applications, 17(8), 1989.

Kim S-J., Koh K., Boyd S., Gorinevsky D. `1 Trend Filtering. SIAM Review, 51(2), 2009.

Montana G., Triantafyllopoulos K., Tsagaris T. Flexible Least Squares for Temporal Data Mining and Statistical Arbitrage. Expert Systems with Applications, 36(2), 2009.

Roncalli T. Introduction to Risk Parity and Budgeting. Chapman & Hall, 2013.

Roncalli T., Weisang G. Tracking Problems, Hedge Fund Replications, and Alternative Beta. Journal of Financial Transformation, 31, 2011.

Scherer B. Portfolio Construction & Risk Budgeting. Third edition, Risk Books, 2007