Abstract: Adaptive Portfolio Allocations

Historically Mean-Variance optimization (MVO) has been the most popular methodology within the finance industry to create allocations for client portfolios. This methodology, however, typically has a few assumptions in practice that may not necessarily hold up through time: 1) Portfolio inputs- specifically asset returns, volatility, and correlation will persist through time 2) The method of correctly integrating these inputs into a portfolio allocation can be adequately captured by the current formulas . Academic papers have explored the idea that returns are the least likely of the three portfolio inputs to persist through time, and to that effect have created alternative risk-based portfolio allocation schemes such as Risk Parity and Minimum Variance. While these allocation schemes have historically performed well, we prefer a more adaptive approach that is free of too many assumptions about what is important and how portfolio inputs should be integrated. We explore the possibility of creating a portfolio allocation scheme using a learning algorithm that can potentially still use all three traditional portfolio inputs, but would be capable of capturing unusual interactions or changes in the relative predictability of the inputs over time.

To that effect we utilize a K-Nearest Neighbors methodology to create an adaptive portfolio allocation scheme using the traditional MVO inputs, and compare its performance to that of traditional MVO with a Maximum Sharpe objective function. K- Nearest Neighbors is a non-parametric, lazy learning algorithm that makes no assumptions about the distribution of the input or output variables. It is able to compare the relationships between the independent and dependent variables merely by observing instances that have independent variables most similar to the instance which we're trying to predict, and capturing the average output from the "Nearest Neighbors." Unlike MVO, which will only give a specific set of outputs (i.e. the portfolio allocation), K-NN will give a distribution of output variables (the K-nearest neighbors) which can be directly observed. Additionally, the relationships between the variables change over time, as the market's responses to the input variables changes over time (i.e. a particular indicator value 10 years ago may not mean the same thing for market returns going forward as it did at that time). This makes the K-NN algorithm ideal for our purposes.

We therefore created a set of K-NN portfolios that attempted to create optimal portfolio allocations to the input securities by using a forward-looking mean variance optimizer to "learn" the appropriate positions for a period of time. The K-NN algorithm then attempted to use these outputs (controlling for 20 day lead) to learn what the best portfolio allocations would be based on the current inputs. This was done on a walk –forward basis without the benefit of hindsight. This is quite different from traditional MVO in that there was no forecast of return, risk, and correlation, but instead we used the historical values to create optimal portfolios. By controlling for the lead within the allocations, we were still able to allow the K-NN algorithm to utilize a sizable set of portfolio allocations without creating any forward-looking bias in the final allocations.

Using the same inputs as the traditional MVO (albeit in a different fashion), we found that the Adaptive Portfolio Allocation approach using K-NN consistently

outperformed the MVO portfolios over long periods of time for both asset classes and equity indices on a risk-adjusted basis. Furthermore, the K-NN portfolios showed lower risk and lower drawdowns in most cases. The most pronounced difference in performance was allocating between stocks and bonds. In this case, the adaptive K-NN approach substantially outperformed MVO which is of major practical interest to most portfolio managers.

In evaluating our K-NN portfolios, we wished to analyze how the algorithm responded to the various portfolio inputs. To that effect, we analyzed the relationship between the K-NN portfolio's performance to a range of allocation schemes, including Mean-Variance, Minimum Variance, Maximum Return, Equal Weight, Risk Parity, and Mean-Reversion. In analyzing the relationships between these portfolios over time, we observed the traditional portfolio to which the K-NN returns most closely correlated. Assuming that MVO is in fact the correct model for asset returns, we would expect that the K-NN historical allocations would correlate most significantly to an MVO/Max Sharpe allocation approach on the same time series. This would be the case if returns, volatility and correlation were all predictable and should be integrated according to the model to generate optimal allocations. In contrast, we found surprisingly that allocations over time for the K-NN approach had high correlations to risk-parity, minimum-variance, and equal weight portfolios. Furthermore, the correlations to a traditional MVO allocation were quite low. However, these correlations were time-varying. This suggests that traditional MVO is not always the best model and should be used with caution. The fact that the attribution to various traditional allocation profiles varied across asset pairs suggests that the optimal model may vary itself according to the universe chosen. This is a major challenge for a traditional MVO approach, but is easily addressed within our multi-asset Adaptive Portfolio Allocation framework.

# Adaptive Portfolio Allocations

David Varadi and Jason Teed

"*You have to be fast on your feet and adaptive otherwise your strategy is useless*"- *Charles de Gaulle*

Perhaps the most important concept that evolved in the last ten years for systematic investors was the necessity for models to adapt to changing market environments. It is well known to financial economists that markets are non-stationary, and this implies that their behavior can change over time. Not only do market returns vary over time, but the autocorrelation matrix that defines whether a market is for example trending or mean-reverting can also change dramatically. This creates a moving target for individuals that try to create systems to beat the market. Just as the archer aims for the bull's-eye: the target moves, hitting nothing but air. That is why bringing in an Olympic archer doesn't guarantee success, nor does bringing in a world class mathematician or physicist to predict the market. That doesn't mean that talent will not help (the best firms do have the best talent), but rather whether it is a human or a computer program, one must recognize that we need to aim for where the target is likely to be and constantly re-calibrate based on how conditions are changing.

The most important technology for money managers to adopt is portfolio allocation. This is the crucial process of determining the appropriate weightings of different asset classes and/or strategies in a portfolio. The current paradigm has not really evolved much from a standard Markowitz "Mean-Variance" approach, created in 1952. This model combines expected returns, volatility, and correlation between securities into one composite portfolio allocation based upon a rigorous mathematical approach. In practice, these forecasts/expected inputs are often replaced with historical values. Mean Variance Optimization (MVO) is still one of the most widely utilized allocation models in the industry despite the fact that the model makes the unrealistic assumption that portfolio inputs will remain stationary. While a large cohort of money managers and even academics are quick to criticize or point out the deficiencies of MVO, no one has really proposed anything that is truly different or better.

A lot of different risk-based schemes such as Minimum Variance, Risk Parity and Maximum Diversification have gained popularity over the years. However, all these methods share the same basic mathematical framework to combine inputs using MVO as a foundation. They differ only in their assumptions of the importance of various inputs such as returns, volatility, or correlation. These methods ignore return inputs in isolation since historical returns are widely considered to be very noisy and contain little information. (Chopra, Ziemba, 1993) Instead, return inputs are assumed to have either a value of zero or are related in some way to volatility or correlations (or both). To their credit, many of these approaches achieve superior Sharpe ratios out of sample across a wide variety of real world data sets. (Chaves, HSU, Li, Shakernia, 2011)

But many of these studies (like virtually all trading system or model development) were done with the benefit of hindsight. It is always easy to show you can hit the target when it isn't moving. *What we really want to do is to develop a method that can learn to hit the target and change its input assumptions over time.*

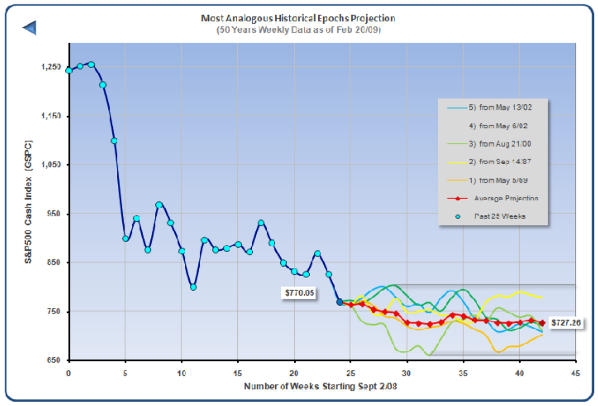
This naturally leads us down the path of creating algorithms that can learn from past data and evolve over time to change the method for creating portfolio allocations. The simplest and most intuitive machine-learning algorithm is the K-Nearest Neighbor method (K-NN). We will describe how this method works in greater detail, but the most important thing to understand is that it is a form of "case-based" reasoning. That is, it learns from examples that are similar to current situation by looking at the past. It shares a lot in common with how human beings make decisions. When portfolio managers talk about having 20 years of experience, they are really saying that they have a large inventory of past "case studies" in memory to make superior decisions about the current environment. The challenges required to harness this experience are the absence of configural thinking, the presence of cognitive biases, and an imperfect memory.

Fortunately, for a quantitative approach, we have the luxury of nearly unlimited data from which to gain effective experience. Unlike a portfolio manager, the K-NN algorithm also does not have the challenge of being biased or capable of remembering information. Furthermore and perhaps most importantly, K-NN employs a moving-window approach and continues to learn as time passes. While a portfolio manager or trader can get trapped in the past by looking at the market the same way, K-NN continues to attempt to adapt over time as conditions change.

Traditional portfolio allocation assumes that the inputs will be combined mathematically in a set manner. Returns are combined linearly while risk is combined as a quadratic function of volatility and correlations. This has a strong theoretical basis, but ignores several possibilities: 1) what if certain inputs are more or less important? 2) What if the appropriate method of combination requires a more complex function? 3) What if certain asset classes or strategies within a specified universe have varying input predictability? For example, what if high-yield bonds have predictable returns but Treasurys do not? 4) What if we want to combine a large set of return and risk-based indicators to determine portfolio allocation for which there is no theoretical foundation or calculation? This is but a small subset of possible questions for which there are no good answers or solutions within this framework.

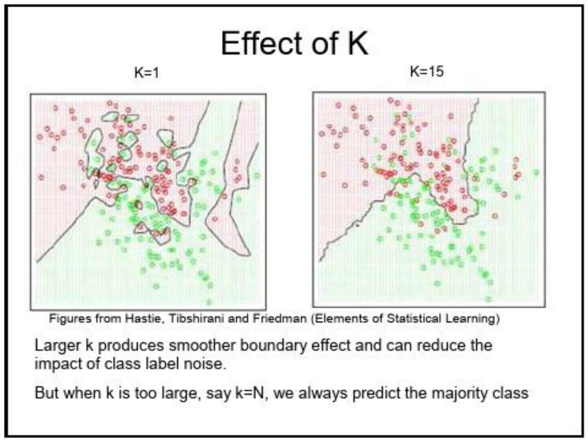
The benefit of using a K-NN approach is that we can inherently address all of these issues simultaneously. Furthermore, K-NN is ideal if we think that there are non-linearities in the relationships between the variables. This simple machine-learning algorithm permits a more flexible and data-adaptive approach. The K-nearest neighbor (K-NN) algorithm is a lazy-learning methodology, meaning that it does not train on the data before a query is made to it. It is also what is called non-parametric, that is, it makes essentially no assumptions about the behavior variable being predicted or the input variables, but is able to learn from relationships between variables, requiring no further inputs about variable distributions or correlations.

By training the K-NN to learn the optimal portfolio allocation for a given case example or set of similar cases, it can generate portfolio allocations that do not map to historical inputs in any structured manner. For example, if we train K-NN to optimize the Sharpe ratio, the mapping between the inputs and the outputs may not have a linear relationship to the final answer. Some inputs may not have much influence, while others may have more. Alternatively, a combination of inputs may yield a very pronounced prediction when they appear in a certain configuration. Ultimately K-NN does not use a function to compute these mappings. It simply says: "what happened historically when I saw patterns that are close to the current pattern?" The most common example of how K-NN is used is to predict future price movements from past price patterns. An example from Market Rewind's "Time Machine" shows clearly how this works below:




Note: Rolling twenty-five week Pearson matching engine courtesy of gummy-stuff.org. (Projecton Range High = 800 Low = 650 )

In this case, the pattern matching engine uses correlations to find the top five nearest matches to the current price pattern in order to see what happened historically. The choice of the number of nearest matches (or neighbors) is the “K” in K-NN. This is an important variable that allows the benefit of allowing the use the ability to trade-off accuracy versus reliability. Choosing a value for K that is too high will lead to matches that are not appropriate to the current case. Choosing a value that is too low will lead to exact matches but poor generalizability and high sensitivity to noise. The optimal value for K that maximizes out-of-sample forecast accuracy will vary depending on the data and the features chosen. The picture below clearly demonstrates this tradeoff:




At K=1, the data is overfit so that every green and red dot are accounted for separately. However, it is unlikely that this exact classification of the decision space will generalize accurately going forward. In contrast, with K=15, there are mislabeled dots on either side of the decision boundary but generally the positioning of the boundary is likely to lead to reliable and accurate prediction out of sample. If we set K=N, then we will always predict whichever outcome (or colored dot) that is most common in the dataset.

To help clarify how K-NN works it is useful to look at an example where we use the method to create a forecast using a popular technical indicator- the RSI.

## A Simple 5-Step Guide to Implementing K-NN

**Step 1:** Identify a feature or set of features to match to historical cases in the past example: let's use RSI(2) as an indicator

**Step 2:** Find the current case For today

example: **today's RSI(2)= 84**

**Step 3:** Choose "k" (in this case let's say we want the top 5 matches) k=5

**Step 4:** Compute a distance metric to find nearest neighbors and rank the closes matches

**Step 5:** Find the next day returns for the nearest matches and average them to find the forecast

**Top 5 Historical Matches In the Last Year**

| RSI(2) | Distance | Rank | Next Day Return |
|---|---|---|---|
| 84.2 | 0.2 | 1 | 0.50% |
| 83.8 | 0.2 | 2 | -0.41% |
| 83.7 | 0.3 | 3 | 0.24% |
| 84.6 | 0.6 | 4 | -0.36% |
| 84.9 | 0.9 | 5 | -1.38% |

Tomorrow's Forecast: **-0.28%**

As stated earlier, the K-NN algorithm makes essentially no assumptions about the behavior of the independent or dependent variables, that is, it merely observes the relationship between the variables over the lookback window, whatever they might be. The algorithm would work for any type of relationship that exists between the independent and dependent variables. Mean-Variance (MVO) makes the assumption that returns, volatility and correlation should be integrated using very specific formulas. MVO assumes that these inputs are always relevant to portfolio allocation, and they are always combined in the same proportion every time. As a consequence, MVO lacks the ability to adapt to changing relationships and while it is a robust framework it may therefore be the wrong model to use over time.

One modification that K-NN often requires is normalization of the inputs so that their relative standard deviations are the same. This is necessary particularly if the features use different scales or have different variances in practice. This normalization helps to ensure consistency in relating a current set of features to examples of the same feature in the past and also helps to ensure that the importance of different features in the composite distance metric is equivalent by design. For example, if we used an RSI indicator that ranged from 0-100 as one feature, but used a ROC (rate of change) indicator which was represented in returns as another feature, there would be substantial noise in matching the current case appropriately with past examples.

## Methodology

**Note**: *for a simple 5-step example of how we created adaptive portfolio allocations please see Appendix A*

Testing was performed on pairs of four major asset classes (Large-Cap domestic stocks, Intermediate Treasurys, Gold, and Broad Commodities) and separately on pairs of four major stock indexes.[1] The period tested for the asset classes was from 4/13/1976 to 12/31/13, as we began at the first date for which daily data was available for all securities plus a 2000 day training period. We also maximized the amount of data available to our models trained on equity indices, resulting in a starting date of 8/9/1995. Our overall goal was to maximize the period tested for each comparative group and was limited only by the shortest asset class for which daily observations were available. We feel that while the results of asset classes are not directly comparable to the equity indices as a result, as we are exploring the feasibility of a methodology rather than selecting appropriate securities for use and backtesting a strategy, that this approach is acceptable.

The training for each model began with learning the best in-sample portfolio allocations for the lookback period. For each day, we took the returns of each security, twenty days into the future, and used an MVO to calculate the portfolios with the highest Sharpe, minimum variance, and maximum return, the latter two calculated only on the S&P vs Treasurys and to be utilized only in an attribution analysis: the focus of this paper will be on maximizing portfolio Sharpe. This was repeated for every day within the training set up to twenty days short of the end of the testing period, beyond which it was not possible to calculate a 20 day portfolio. This avoided any assumptions in future security returns as we used the actual returns over the forecast horizon to create optimal security weights over the horizon rather than predicting them.

Once the training was finished for a pair, we calculated the inputs for our K-NN algorithm which were separated into two groups. The first group was a relative form of our indicators while the second was a self-normalized form of the indicators. The indicators that we chose were simply security momentum, standard deviation and correlation to ensure that our K-NN algorithm didn't have access to any information that the MVO didn't have, but merely used it differently.

**Indicators**

Indicators that were used as inputs for both the K-NN portfolios and the MVO portfolios were historical returns, volatility, and correlation. The inputs calculated for the MVO for comparison were based on six sets of lookbacks: MVO portfolios were run using return and standard deviation indicators based on lookbacks of 20, 40, 60, 120, and 252 days as well as inputs which were a weighted average of all of these parameters. Correlation matrices were also based on these lookbacks.

For the K-NN inputs, the indicators calculated were of two types, resulting in two different sets of K-NN runs, relative return based indicators "Rel", and self-normalized indicators "N". As mentioned previously, K-NN algorithms perform better when their features are normalized so that any one feature does not dominate the distance calculation. The Relative return based indicators were calculated such that the values of each security were normalized relative to the other security in the portfolio, explicitly, the return algorithm was:

$$Normsdist\left(\frac{(Avg\ Ret\ A - Avg\ RetB) - Avg(Avg\ Ret\ A - Avg\ RetB)}{Std(Avg\ Ret\ A - Avg\ Ret\ B)}\right)$$

While the equation for the relative Standard Deviation was:

$$Normsdist\left(\frac{(Std(A) - Std(B)) - Avg(Std(A) - Std(B))}{Std\big(Std(A) - Std(B)\big)}\right)$$

As both of these indicators were calculated on a relative basis, the value calculated for the indicator of asset B is merely a mirror of asset A's indicator, therefore, only the indicators calculated on asset A were included in the Rel_K-NN portfolios which, when combined with correlation, resulted in three final inputs for the Rel_K-NN portfolios. As correlations are inherently relative in nature, the correlation was self-normalized for both sets of runs, that is, it was normalized by its own distribution, calculated as follows:

$$Normsdist\left(\frac{Correl(A,B) - Avg(Correl(A,B)}{Std\big(Correl(A,B)\big)}\right)$$

The Self-Normalized Return indicator was normalized in similar fashion for both securities, and both were added to the feature space of the K-NN optimizer.

$$Normsdist(\frac{Avg\ Ret(A) - Avg(Avg\ Ret\ A)}{Std(Avg\ Ret\ A)})$$

Finally, the self-normalized Standard Deviation Indicator was also normalized in this fashion, resulting in 5 final inputs for the Self-Normalized or "N_K-NN" runs:

$$Normsdist(\frac{Std(A) - Avg(Std(A)}{Std\big(Std(A)\big)})$$

Where we created the indicators in a nested fashion, the same lookback was used within the nesting, resulting in a doubling of the required data to calculate the indicator value, i.e. a lookback of 20 would require 40 days to initialize.

Indicators were created for periods of 20, 40, 60, 120, 252, and a weighted average of these values. This was done to determine how sensitive our learning algorithm would be to parameter selection compared to MVO and to give us a more robust sense of the success/fail rate of our methodology.

After having calculated the inputs to our K-NN algorithm, we entered them into the feature space, and performed our K-NN analysis. As mentioned before, we used Euclidean distance to measure the difference between different day's indicator values, and we also assumed that each indicator was equally important by weighting it equally with the other indicators. Our only assumption was that we knew nothing about the importance of the indicators in the feature space or their relationships with one another. Continuing in this fashion, we also did not select a specific K days with which to compare our current day, but rather selected a range of Ks to make our selection base more robust to potential changes in an “optimal” K selection. The K’s chosen were in percentages of the size of the training space, which were 5%, 10%, 15% and 20% resulting essentially in a weighted average of the top K instances. Additionally, when we performed the K-NN analysis, we used lookback windows of 2000, 1500, and 1000 and averaged the resulting output values. We did this so that we could ensure that our results were not dependent upon choosing any particular lookback period. We felt that this kept the algorithm more robust to market changes in feature relevance, though one could most certainly train for an optimal lookback window and top K on a forward-looking basis, we’ll leave it to the reader to explore these concepts.

For every twentieth day, we calculated the K nearest neighbors from the lookback period length to minus 20 days (to avoid any look-ahead bias as we trained our output variable with returns twenty days in the future). We repeated this for each lookback period and indicator type (“Rel” and “N”) and averaged each model’s top K portfolio weights (our K-NN dependent variable) to arrive at our final portfolio weight. (i.e. Rel_K-NN20, N_K-NN20, Rel_K-NN40, etc.). Once weights were calculated for the entire testing periods, we calculated the returns of each portfolio over the testing period and collected the results.

For comparative portfolios, a traditional Mean-Variance Optimizer using Max Sharpe as the objective function (tangency portfolio) was run on a monthly basis using the raw indictors described above. Six portfolios based on these indicators were calculated to compare against our K-NN portfolios (i.e. MVO20, MVO40, MVO60, etc.). We additionally created out-of-sample portfolios based on Minimum Variance, Maximum Return, Equal Weight, Risk Parity, and Mean reversion (using the MVO, but with return signs of opposite value) using the same parameter sets where applicable.

Additionally we wished to test the efficacy of K-NN algorithms on multi-asset portfolios without introducing the problem of dimensionality with too-large a feature space. The problem of dimensionality occurs when the size of the feature space is large relative to the number of observations available. Therefore, to explore multi-asset portfolios, we took the average weight of each security from a single-pair run, and averaged them across all pair runs. This was done for asset classes as a group (portfolio of 4), and equity indices as a group (portfolio of 3), and calculated the returns for the portfolios. We then ran a traditional MVO portfolio containing these securities for proper comparison.

Lastly, to test the efficacy of using Returns, Variance, and Correlation in creating portfolio allocations, we compared the results of our K-NN portfolios based on optimizing for maximum Sharpe on six major pairings (excluding commodities) with our out-of-sample comparative portfolios. The comparison was done with a rolling 60 day correlation between the K-NN portfolios returns and returns of the comparative portfolios. We then ranked the K-NN portfolio by the out-of-sample portfolio with which it was most closely correlated to give an idea of how the algorithm was behaving through time.

This process was repeated on the S&P vs. Intermediate Treasurys portfolio utilizing a Minimum Variance and Maximum Return K-NN as mentioned earlier, in an attempt to show how the relationship between these portfolios and their out-of-sample counterparts differ.

**Results: Portfolios of two assets:**

To compare the efficacy of K-NN over traditional MVO, we aggregated the performance statistics of the six K-NN models created from the differing indicator lookbacks and performance statistics of the MVO portfolios across the same lookback parameters that we tested. Results were grouped into two broad sets, heterogeneous- basic asset classes (S&P, Gold, Intermediate Treasurys, and Broad Commodities) and homogenous- equity indices (S&P 500, Dow Jones Industrial Average, and Russell 2000).

The results demonstrated that the Adaptive Portfolio Allocation approach tended to outperform MVO portfolio allocation on a risk-adjusted basis. This was broadly true regardless of the indicator set chosen (whether “Rel” or “N” inputs into

K-NN). On 5 out of the 6 asset class pairs, the Sharpe ratio achieved for either K-NN model was greater than the comparable MVO portfolio. The greatest differential in portfolio performance between K-NN and MVO was in the portfolio consisting of the S&P 500 and Intermediate Treasurys. In this case the adaptive K-NN approach had a Sharpe ratio that was over 30% higher than MVO. K-NN also had higher returns, lower risk, and lower drawdowns trading the stock and bond pair than mean-variance. This result is an especially important finding because most portfolio allocation decisions for active portfolio managers revolve around the optimal allocation between stocks and bonds.

On equity indexes, we found that K-NN was also superior- outperforming MVO on two out of three pairs, as well as beating on a raw return basis. Interestingly, we also found that the K-NN portfolios also improved (lowered) the maximum drawdown on each run, which was also seen in 4 out of the 6 asset class portfolios.

**Results: Multi-Asset Portfolios**

In our multi-asset portfolios, we saw similar outperformance for Adaptive Portfolio Allocations versus MVO. Using the multi-asset approach on asset classes, the K-NN methodology beat MVO methodology on average by 21.3% on a risk/adjusted or Sharpe ratio basis.  We did also note that the distribution of the back-test returns, standard deviations, and DD across parameters for the MVO approach was substantially higher than K-NN approach. This suggests that the K-NN methodology is more robust than MVO to the choice of lookback period for input parameters.

In mapping out the portfolio weights through time, the K-NN portfolios demonstrated much more stability through time, making more balanced portfolios at any point in time than the comparable MVO portfolio.

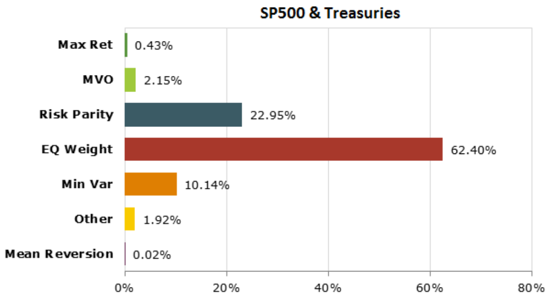
We did also note that the 252 day performance of the MVO on asset classes seemed to be a special case in that its returns and Sharpe were much higher than other parameter lengths on the MVO portfolios. This likely reflects the momentum effect that is documented in countless academic studies. While it has been persistent historically, there is no guarantee it will persist into the future.

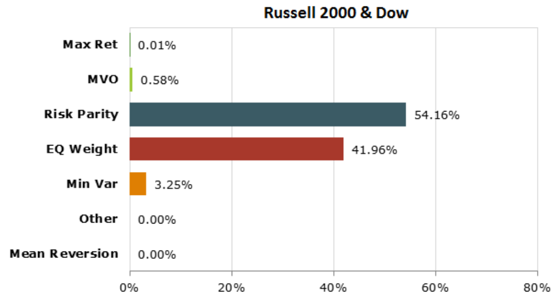
We also wanted to determine whether or not there was any synergy in combining MVO and K-NN. To that effect, we created a 50/50 portfolio combination of these portfolios and calculated the resulting statistics. For the period tested, there does seem to be some synergy in combining these two methodologies in that the average portfolio Sharpe was higher than for K-NN or MVO alone.

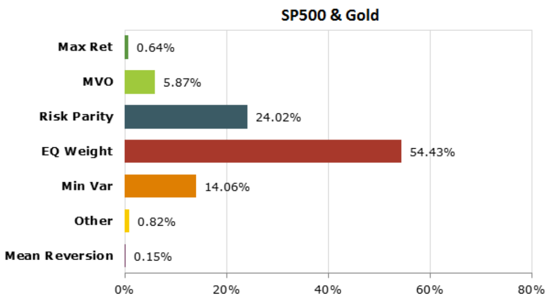
Within our equity index portfolios K-NN continued to outperform mean variance optimization, on a Return, Standard Deviation, and Max DD basis. Once again, the portfolio weights were much more stable through time. The average return on our K-NN portfolios was a 0.88% higher on an annualized basis, and average standard deviation was 12 basis points lower. We did not find that there was, on average, a synergistic relationship between the two types of portfolios, with the Sharpe of a 50/50 blending of them still underperforming the K-NN portfolios.

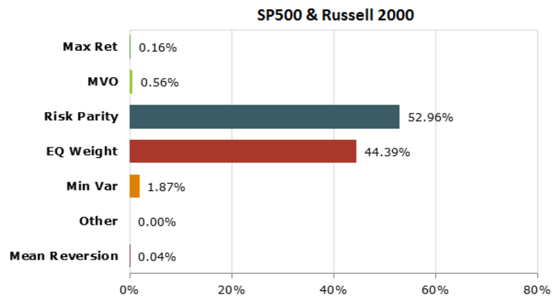
**Portfolio Attribution Analysis**

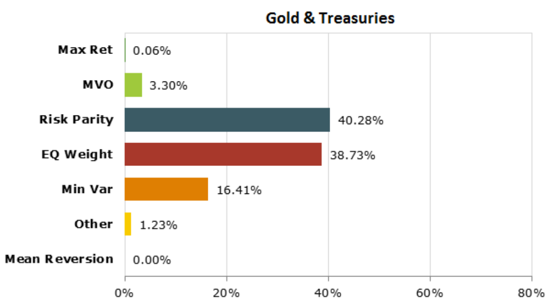
In the comparison to our out-of-sample comparative portfolios, we found that by-and-large, that equal weight and risk parity had the highest correlation on average with the K-NN portfolios without exception. This would seem to indicate that the K-NN algorithm found a weak relationship between our historical inputs and future returns, defaulting it either to an equal weight algorithm (no information) or risk parity (information on volatility is important, but correlations and return are not, as neither are required to calculate a risk parity portfolio). Of course, this also reflects the research that risk-based portfolio allocation using risk parity or minimum variance has validity. Our hypothesis is that relative returns do in fact have limited information and they dilute the ability for K-NN to pick up a meaningfully different result than equal weight much of the time. The volatility and correlations may in fact have some meaningful information but this is probably diluted by the relative return feature in computing distance. Interestingly, very few pairs showed any relationship with the out-of-sample MVO portfolio, the highest being S&P500 vs. Gold, correlating highly only 5.87% of the time. Since MVO weights are dominated by relative return estimates, this seems to support our contention.

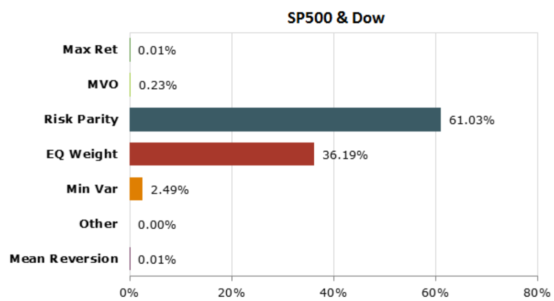


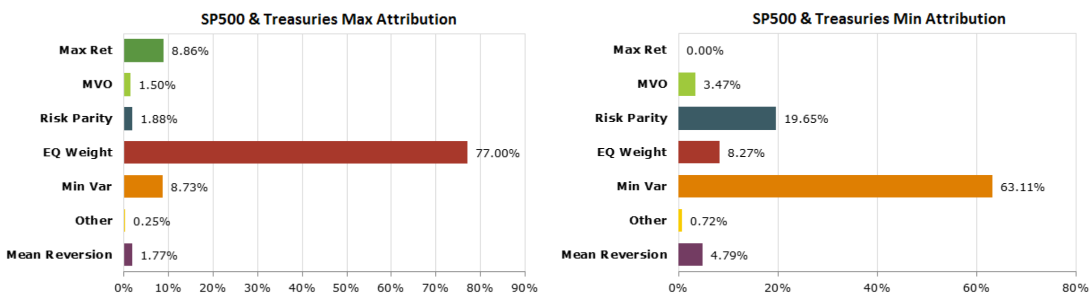




To test this idea further, we performed an attribution analysis on a K-NN portfolio on the S&P vs. Intermediate Treasurys that was trained on the optimal maximum return portfolio allocations, which only requires security return as an input. This portfolio represents the highest returning portfolio on the efficient frontier. We found that this model correlated most highly with Equal Weight 77% of the time, far beyond the Equal weight attribution of the maximum Sharpe K-NN models, suggesting that the inputs could not accurately predict the maximum return weights and the future security returns used to create them. Therefore, even though our inputs may have had financially beneficial information contained within them, they could not be utilized to determine what the future returns going forward would be, or, indeed, whether one security over another would outperform. An equal weight outcome from the K-NN portfolio suggests that there was essentially no relationship between returns of the securities over the subsequent 20 day period, and the input indicators. This further implies that blindly using MVO using historical inputs in portfolio allocation could cause highly error-prone weighting schemes.



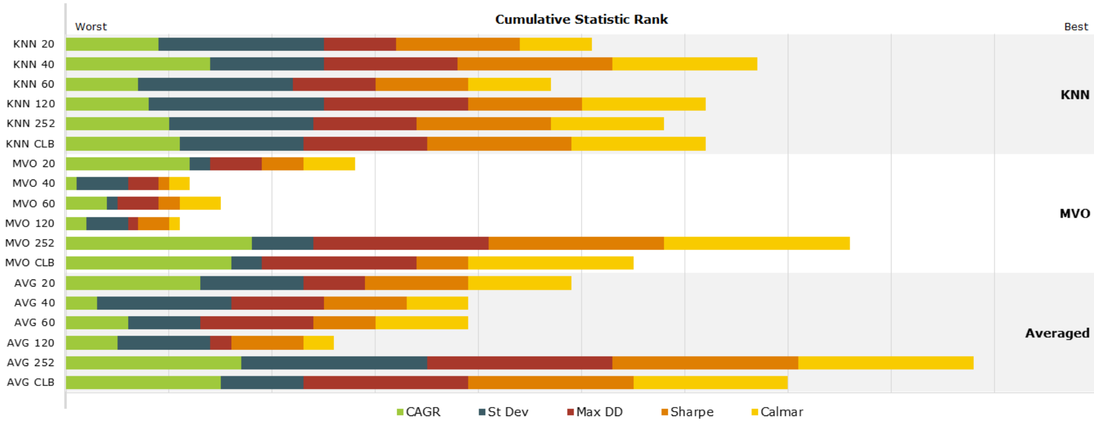

We repeated this exercise for testing the efficacy of variance and correlation together by comparing a Minimum Variance K-NN portfolio on the same pair and performing an attribution analysis (we used Minimum Variance as the target objective function and training algorithm). We found that the majority of the time, that the K-NN portfolio correlated most highly with the traditional minimum variance portfolio, suggesting that there was a significant relationship between the feature space and the future variance and correlation of the securities, upon which the learned portfolio allocations were based. The second most highly correlated out-of-sample portfolio was the risk parity portfolio, which uses volatility only in its calculation.  Again these results support our initial hypothesis—at least for stocks and bonds.

**Conclusion**

In this paper we introduced a new framework for adaptive portfolio allocation using K-NN as a base learning algorithm. This framework permits substantial flexibility in the choice of indicators for portfolio allocation, and it also allows for adaptation to changing relationships over time in the portfolio inputs used to generate allocations. We compared this simple K-NN approach to a dynamic MVO approach on a maximum Sharpe objective function and found that K-NN consistently outperformed on both heterogeneous and homogenous data sets on a risk-adjusted basis. Upon further analysis using portfolio attribution, we found that the adaptive portfolio allocation approach did not rely as much on relative returns as it did on volatility and correlations over time. Furthermore, the fact that allocations were often similar to an equal weight portfolio demonstrated the general uncertainty of the portfolio indicator inputs in aggregate. The adaptive allocation approach managed to dynamically balance this uncertainty over time and shift more towards a probabilistic allocation that did not overweight or over-react to poor information. This was demonstrated in the superior out-of-sample performance and also the stability of the transition maps which show the historical allocations to various assets as a function of time versus MVO which showed both considerable noise and turnover. Future research could be taken in a lot of different directions. Clearly superior inputs, indicators, or forecasts could dramatically improve performance and that is a natural avenue to explore. There is no question that the K-NN approach itself can be used separately as a component to forecast each individual input—that is use K-NN to forecast returns, volatility and correlations separately. Furthermore, a lot of work could be done in the area of using this approach on multiple assets simultaneously rather than just using pairs. Finally, another interesting area is to continue to explore the use of adaptive portfolio allocations as an input to be combined with a more traditional approach. Ultimately we have demonstrated at the very least that the Adaptive Portfolio Allocation approach is both conceptually appealing and shows promising results for further research.

## S&P500 & Gold & Treasuries & Commodities

Statistics by Method from 1968 through 2013




### Statistic Summary by Method and Lookback

| | KNN | | | | | | MVO | | | | | | Averaged | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 20 | 40 | 60 | 120 | 252 | CLB | 20 | 40 | 60 | 120 | 252 | CLB | 20 | 40 | 60 | 120 | 252 | CLB |
| **CAGR** | 7.20% | 7.41% | 7.19% | 7.19% | 7.28% | 7.34% | 7.39% | 6.18% | 6.88% | 6.79% | 9.66% | 8.26% | 7.39% | 6.88% | 7.12% | 7.06% | 8.52% | 7.88% |
| **St Dev** | 7.48% | 7.55% | 7.50% | 7.47% | 7.51% | 7.52% | 9.93% | 9.35% | 10.06% | 9.63% | 8.40% | 9.75% | 7.70% | 7.52% | 7.86% | 7.77% | 7.29% | 7.82% |
| **Max DD** | 27.18% | 26.54% | 27.12% | 26.33% | 26.88% | 26.80% | 30.69% | 32.61% | 31.99% | 42.83% | 22.31% | 26.15% | 27.76% | 26.93% | 26.83% | 34.00% | 21.05% | 25.95% |
| **Sharpe** | 0.96 | 0.98 | 0.96 | 0.96 | 0.97 | 0.98 | 0.74 | 0.66 | 0.68 | 0.70 | 1.15 | 0.85 | 0.96 | 0.91 | 0.91 | 0.91 | 1.17 | 1.01 |
| **Calmar** | 0.26 | 0.28 | 0.26 | 0.27 | 0.27 | 0.27 | 0.24 | 0.19 | 0.22 | 0.16 | 0.43 | 0.32 | 0.27 | 0.26 | 0.27 | 0.21 | 0.40 | 0.30 |

**Top 50% highlighted by statistic color**

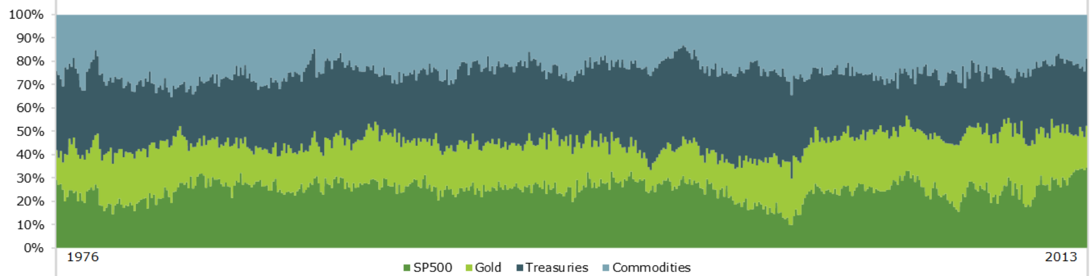
CLB = Combined Lookback

### Statistic Summary by Method

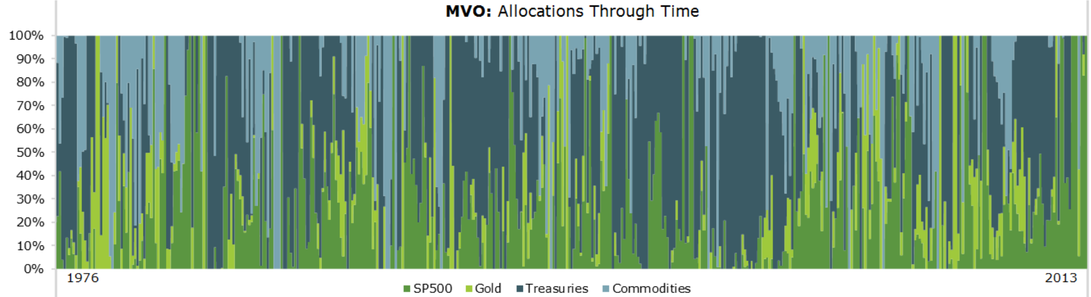
| | KNN | | MVO | | Averaged | |
|---|---|---|---|---|---|---|
| | Average | St Dev | Average | St Dev | Average | St Dev |
| **CAGR** | 7.27% | 0.09% | 7.53% | 1.25% | 7.48% | 0.62% |
| **St Dev** | 7.51% | 0.03% | 9.52% | 0.60% | 7.66% | 0.22% |
| **Max DD** | 26.81% | 0.33% | 31.10% | 6.97% | 27.09% | 4.15% |
| **Sharpe** | 96.84% | 0.92% | 79.85% | 18.40% | 97.77% | 10.15% |
| **Calmar** | 27.12% | 0.56% | 25.88% | 10.06% | 28.39% | 6.67% |

**Best highlighted by statitic color**

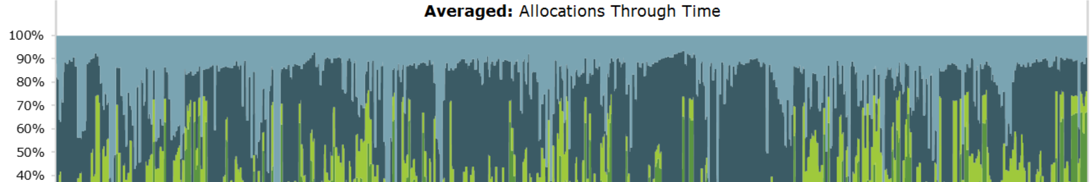
**KNN:** Allocations Through Time




**MVO:** Allocations Through Time




**Averaged:** Allocations Through Time



## Asset Classes

**SP500 & Treasuries**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 8.02% | 8.31% | 7.68% |
| St Dev | 8.21% | 8.08% | 9.77% |
| Max DD | 24.21% | 21.65% | 29.53% |
| Sharpe | 0.98 | 1.03 | 0.79 |
| Calmar | 0.33 | 0.40 | 0.26 |

**SP500 & Gold**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 9.88% | 9.59% | 9.99% |
| St Dev | 13.29% | 13.29% | 16.70% |
| Max DD | 35.46% | 35.42% | 46.97% |
| Sharpe | 0.74 | 0.72 | 0.60 |
| Calmar | 0.28 | 0.27 | 0.23 |

**Gold & Treasuries**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 7.30% | 7.29% | 8.29% |
| St Dev | 9.57% | 9.54% | 13.53% |
| Max DD | 32.32% | 32.90% | 47.54% |
| Sharpe | 0.76 | 0.76 | 0.61 |
| Calmar | 0.23 | 0.22 | 0.18 |

**SP500 & Commodities**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 6.92% | 6.84% | 6.88% |
| St Dev | 11.20% | 11.24% | 13.46% |
| Max DD | 44.31% | 44.99% | 51.99% |
| Sharpe | 0.62 | 0.61 | 0.51 |
| Calmar | 0.16 | 0.15 | 0.13 |

**Treasuries & Commodities**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 5.51% | 5.45% | 7.75% |
| St Dev | 6.62% | 6.60% | 8.79% |
| Max DD | 24.10% | 23.95% | 23.66% |
| Sharpe | 0.83 | 0.83 | 0.88 |
| Calmar | 0.23 | 0.23 | 0.33 |

**Gold & Commodities**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 4.40% | 4.66% | 5.12% |
| St Dev | 12.00% | 12.11% | 14.94% |
| Max DD | 50.85% | 51.35% | 46.53% |
| Sharpe | 0.37 | 0.38 | 0.34 |
| Calmar | 0.09 | 0.09 | 0.12 |

## Equity Indices

**Russel 2000 & Dow**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 9.53% | 9.39% | 7.98% |
| St Dev | 19.70% | 19.65% | 20.43% |
| Max DD | 54.82% | 54.58% | 55.66% |
| Sharpe | 48.36% | 47.78% | 39.08% |
| Calmar | 17.39% | 17.20% | 14.36% |

**SP500 & Russel 2000**

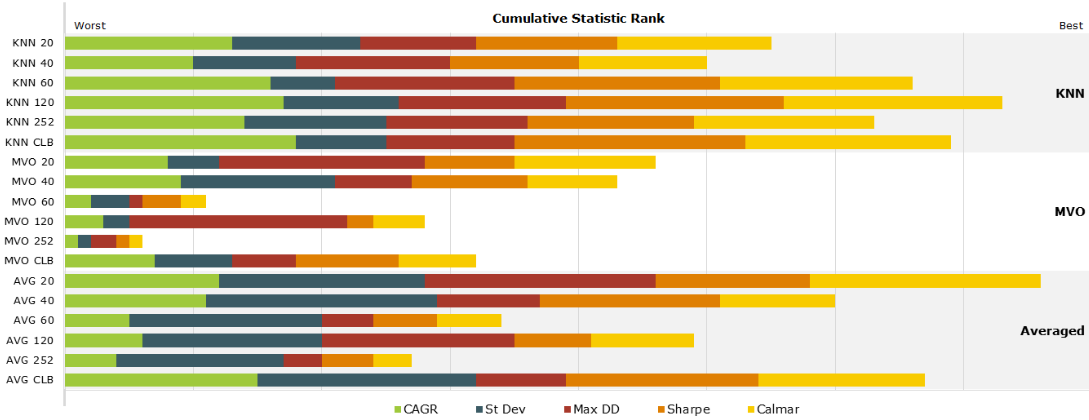
| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 8.86% | 8.86% | 8.50% |
| St Dev | 20.76% | 20.74% | 21.38% |
| Max DD | 57.00% | 56.63% | 57.52% |
| Sharpe | 42.67% | 42.73% | 39.80% |
| Calmar | 15.54% | 15.65% | 14.78% |

**SP500 & Dow**

| | KNN_Rel_Ret | KNN_N_Ret | MVO |
|---|---|---|---|
| | Avg | Avg | Avg |
| CAGR | 9.21% | 9.23% | 9.56% |
| St Dev | 19.11% | 19.11% | 19.09% |
| Max DD | 53.34% | 53.19% | 53.43% |
| Sharpe | 48.21% | 48.30% | 50.09% |
| Calmar | 17.27% | 17.35% | 17.90% |

## S&P500 & Dow & Russell 2000

Statistics by Method from 1968 through 2013




### Statistic Summary by Method and Lookback

| | KNN | | | | | | MVO | | | | | | Averaged | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 20 | 40 | 60 | 120 | 252 | CLB | 20 | 40 | 60 | 120 | 252 | CLB | 20 | 40 | 60 | 120 | 252 | CLB |
| **CAGR** | 9.06% | 9.00% | 9.31% | 9.34% | 9.12% | 9.37% | 8.88% | 8.89% | 8.02% | 8.02% | 7.22% | 8.87% | 9.02% | 9.00% | 8.73% | 8.74% | 8.22% | 9.18% |
| **St Dev** | 20.21% | 20.23% | 20.29% | 20.21% | 20.20% | 20.26% | 20.35% | 20.12% | 20.41% | 20.47% | 20.49% | 20.27% | 20.03% | 19.91% | 20.05% | 20.06% | 20.09% | 20.00% |
| **Max DD** | 55.48% | 55.34% | 55.15% | 55.15% | 55.43% | 55.45% | 54.56% | 56.17% | 58.29% | 54.50% | 58.11% | 56.27% | 52.71% | 55.69% | 56.67% | 54.75% | 56.73% | 55.80% |
| **Sharpe** | 44.81% | 44.50% | 45.91% | 46.23% | 45.14% | 46.25% | 43.66% | 44.18% | 39.31% | 39.19% | 35.24% | 43.75% | 45.06% | 45.21% | 43.55% | 43.58% | 40.92% | 45.88% |
| **Calmar** | 16.32% | 16.26% | 16.89% | 16.94% | 16.45% | 16.89% | 16.28% | 15.83% | 13.76% | 14.72% | 12.43% | 15.76% | 17.12% | 16.16% | 15.41% | 15.97% | 14.49% | 16.45% |

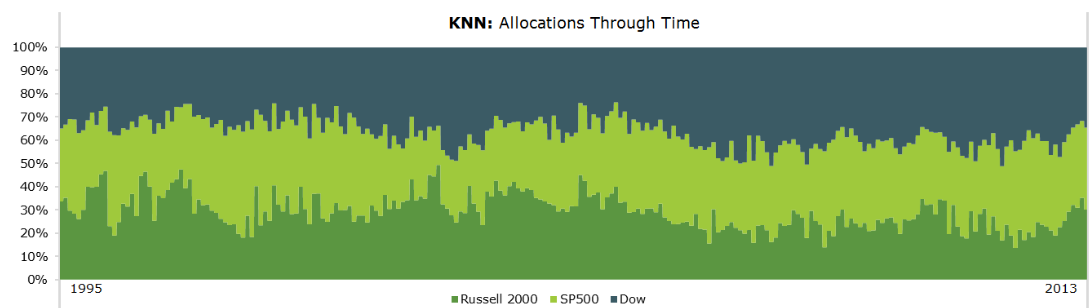
**Top 50% highlighted by statistic color**

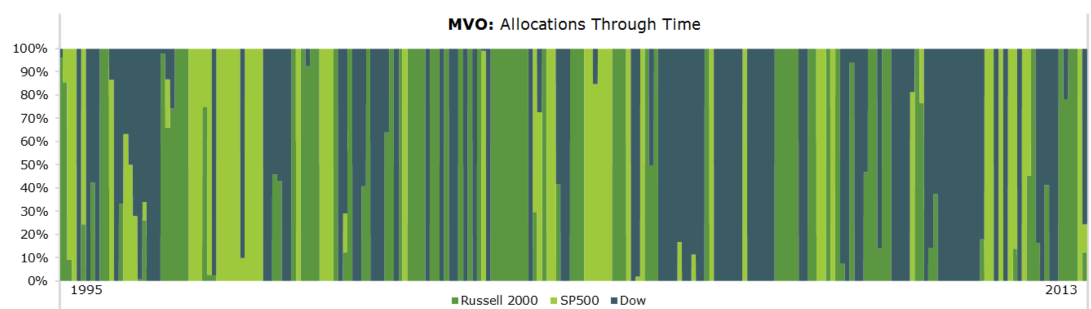
CLB = Combined Lookback

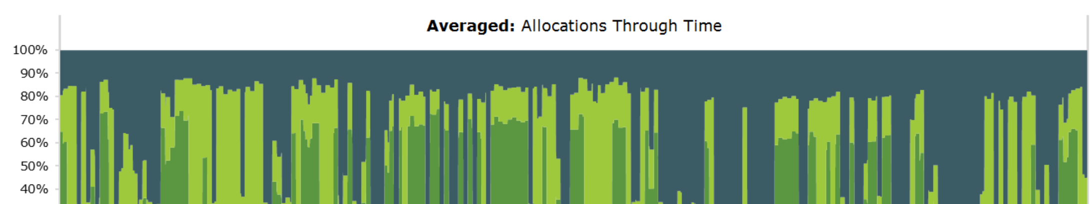
### Statistic Summary by Method

| | KNN | | MVO | | Averaged | |
|---|---|---|---|---|---|---|
| | Average | St Dev | Average | St Dev | Average | St Dev |
| **CAGR** | 9.20% | 0.16% | 8.32% | 0.68% | 8.82% | 0.34% |
| **St Dev** | 20.23% | 0.03% | 20.35% | 0.14% | 20.02% | 0.06% |
| **Max DD** | 55.33% | 0.15% | 56.32% | 1.64% | 55.39% | 1.50% |
| **Sharpe** | 45.47% | 0.76% | 40.89% | 3.58% | 44.03% | 1.79% |
| **Calmar** | 16.63% | 0.31% | 14.80% | 1.48% | 15.93% | 0.90% |

**Best highlighted by statitic color**



# Appendix A:

### A Simple 5-Step Guide to Implementing K-NN for Adaptive Portfolio Allocation

**Step 1:** Choose normalized measures of the three features used for portfolio allocation
example: relative return (stocks to bonds), relative volatility (stocks to bonds) and relative correlation (of stock/bond)

**Step 2:** Find the current case For today

| | Rel Ret | Rel Vol | Rel Cor |
|---|---|---|---|
| example: | 87 | 56 | 63 |

**Step 3:** Choose "k" (in this case lets say we want the top 5 matches) therefore we set ***K=5***

**Step 4:** Compute a distance metric to find nearest neighbors and rank the closes matches. In this paper we used Euclidean Distance which is calculated as:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

Where q represents today's values of indicators 1:n, and p represents the value of indicators 1:n of the all the instances in the lookback window, taken one at a time.

**Step 5:** Find the next day returns for the nearest matches and average them to find the forecast

- Find the closest K instances within the lookback window by distance, and average the output values for those K instances to arrive at the next period allocation.

**Top 5 Historical Matches In the Last 2000 Days**

| Rel Ret | Rel Vol | Rel Cor | Distance (Euclidean) | Rank | Optimal Next Period Allocation | |
|---|---|---|---|---|---|---|
| | | | | | % in stocks | % in bonds |
| 88 | 55 | 64 | 1.73 | 1 | 55% | 45% |
| 89 | 57 | 62 | 2.45 | 2 | 62% | 38% |
| 85 | 53 | 63 | 3.61 | 3 | 53% | 47% |
| 90 | 60 | 65 | 5.39 | 4 | 57% | 43% |
| 83 | 49 | 70 | 10.68 | 5 | 64% | 36% |

| **Next Period Portfolio Allocation:** | **% in stocks** | **% in bonds** |
|---|---|---|
| (average of allocations for all nearest matches) | **58.2%** | **41.8%** |

# Appendix B: Data Sources

| | From | To | Symbol | Name | Source | Notes |
|---|---|---|---|---|---|---|
| Treasury | 1/2/1962 | 2/28/1994 | USGG10YR Index | US Government 10 Year Yield | Bloomberg | Yield to Price Conversion |
| Treasury | 2/28/1994 | 7/26/2002 | XIUSA000MJ | Barclays Intermediate Treasury TR USD | Morning Star | |
| Treasury | 7/26/2002 | Present | IEF | IEF US Equity | Bloomberg | |
| Gold | 4/1/1968 | Present | GOLDLNPM | London Gold PM | Bloomberg | |
| CCI | 9/4/1956 | Present | CCI | Continuous Commodity Index | Bloomberg | |
| S&P500 | 2/24/1950 | 12/31/1987 | ^GSPC | S&P 500 | Yahoo | |
| S&P500 | 1/4/1988 | Present | SPTR | SP500 Total Return | Bloomberg | |
| Russell 2000 | 9/11/1987 | Present | ^Rut | ^Russell 2000 | Yahoo | |
| Dow | 9/30/1987 | Present | XIUSA000PF | Dow Jones Industrial Average TR USD | Morning Star | |

## Works Cited:


Chaves, D., Hsu, J., Li, F., & Shakernia, O. (2011). Risk parity Portfolios vs. Other Asset Allocation Heuristic Portfolios, The Journal of Investing, *20*(1), 108-118.

Chopra, V., Ziemba, W. (1993). The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio choice, The Journal of Portfolio Management, *19*(2), 6-11.