MARKOWITZ-STYLE QUARTIC OPTIMIZATION FOR THE
IMPROVEMENT OF LEVERAGED ETF TRADING

by

Jackson Paul DeWeese

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Professional Degree of Master of Science
in
Financial Mathematics
April 2013

Approved:

Dr. Marcel Blais, Advisor

Dr. Bogdan Vernescu, Head of Department
# TABLE OF CONTENTS

Abstract ................................................................................................................................. 2  
Background ............................................................................................................................. 3  
Leveraged ETFs ...................................................................................................................... 3  
   “Discount-and-up” ................................................................................................................... 4  
Markowitz Optimization ......................................................................................................... 4  
GARCH Model ....................................................................................................................... 5  
Methodology .......................................................................................................................... 7  
Data ......................................................................................................................................... 7  
Tracking Error and Momentum ............................................................................................... 8  
Time Series .............................................................................................................................. 9  
Markowitz-style Optimization ............................................................................................... 12  
Purchase and Sell Orders ........................................................................................................ 13  
Results .................................................................................................................................... 15  
Implementation ...................................................................................................................... 15  
Analysis ................................................................................................................................... 16  
   Sharpe Ratio .......................................................................................................................... 18  
   Returns ................................................................................................................................ 19  
   Influence of Volatility .......................................................................................................... 24  
Conclusion ............................................................................................................................... 28  
References .............................................................................................................................. 29  
Appendix ................................................................................................................................ 31  
A. MATLAB code .................................................................................................................... 31  
   I. Main body of data testing .................................................................................................. 31  
   II. Quartic optimization routine: ....................................................................................... 37  
   III. Quadratic optimization routine (for Markowitz) ............................................................ 38  
   IV. GARCH forecasting ....................................................................................................... 39  
B. Java Code ............................................................................................................................ 40  
   I. Data cleaner ...................................................................................................................... 40
ABSTRACT

This paper seeks to unconventionally maximize the volatility of a portfolio through a quartic optimization based on Markowitz's modern portfolio theory, which generally seeks to do exactly the opposite. It shows that through this method, a daily leveraged exchange traded fund (ETF) strategy investigated by Posterro can be significantly improved upon in terms of its Sharpe ratio. The original strategy seeks to use a combination of momentum trading and tracking error in leveraged ETFs to trade during the last half an hour of the trading day, but it suffers in a low volatility market. By maximizing the volatility to take better advantage of tracking error and momentum, this problem is addressed by both increasing the mean daily return and significantly decreasing the variance of the strategy's daily returns. GARCH forecasting is also implemented to assist in the maximization of the daily portfolios' variances, though this does not prove to make a statistically significant difference in the strategy's performance.
BACKGROUND

LEVERAGED ETFS

Exchange traded funds (ETFs) are tradable securities that attempt to track indices and commodities. They are similar to mutual funds in that they give an investor easier access to trade financial instruments they would be otherwise unable to trade (e.g. a single investor would generally be unable to trade the S&P500 as this would require in investing in 500 companies). Unlike a mutual fund, however, these exchange traded funds can be bought and sold much like a stock instead of only once per day at market closing. If an investor wished to go long on the price of gold for only a portion of the day, he could purchase gold ETF shares and sell them for a profit (or loss) later that same day.

Leveraged ETFs take this a step further by multiplying the daily return by their respective leverage factor (generally 2, -2, 3 or -3) by using financial derivatives such as swaps. Because this is a leverage of the daily return, over a longer span of time (e.g. a month, a year), the leveraged ETF's return will not accurately match a multiplier of the underlying asset’s return. For example, in the last year, the Russell 1000 index has returned 10.5% and a leveraged ETF (FAS) tracking it at x3 had a performance of 49.6% over the same time period, not 31.5% (Yahoo! Finance, 2013). Two of the largest leveraged ETF providers are Direxion and ProShares, together having almost 200 leveraged ETFs available to trade and encompassing all of the leveraged ETFs used in this project.

While non-leveraged ETFs experience negligible tracking error due to commissions, fees, etc. and have no need to rebalance frequently, leveraged ETFs must rebalance on a daily basis to correct their tracking error and maintain their leverage throughout the day (Trainor, 2010). Theoretically the rebalancing is done as closely as possible to the end of trading each day (Cheng and Madhavan, 2009). This leads to the popular strategy of momentum trading where an ETF is
bought near the end of trading if it is in the black for the day (i.e. positive), but while this strategy has shown potentially high returns, its variability and expected shortfall leave much to be desired (Posterro, 2009)

"DISCOUNT-AND-UP"

To improve upon this, Posterro studied the "discount-and-up" strategy, wherein momentum trading was only implemented if the leveraged ETF was not only trading up for the day, but also trading at a discount with respect to its target leverage. The result was a strategy that performed equally, if not superiorly, well with regards to expected return, but also with a more manageable volatility and expected shortfall. Posterro also found a positive correlation between the market volatility and the performance of the "discount-and-up" strategy, and posed the investigation in varying levels of market volatility, as low volatility led to a much lower returns for the strategy (Posterro, 2009). This paper seeks to do exactly that.

MARKOWITZ OPTIMIZATION

Efficient frontier theory seeks to take a portfolio of securities and find the ideal weights so as to minimize the variance with the condition of a minimum expected return based on historical returns (or maximizing the expected return with the condition of a maximum portfolio variance). Mathematically,

$$\min_w w^T \Sigma w \quad \text{s.t.} \quad R^T w = \mu_{\text{target}} \quad \text{and} \quad \sum_i w_i = 1$$

where $w$ is the weight vector, $\Sigma$ is the covariance matrix of the securities’ returns based on historical data,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1,N} \\ \vdots & \ddots & \vdots \\ \sigma_{1,N} & \cdots & \sigma_N^2 \end{bmatrix} \quad \text{s.t.} \quad \sigma_{i,j} = \sum_{t=0}^T \frac{(r_{i,t} - \mu_i)(r_{j,t} - \mu_j)}{30} \quad (\text{and} \quad \sigma_i^2 = \sigma_{i,i})$$

and $R$ is the expected return vector based on historical data.
The term being minimized \((w^T\Sigma w)\) represents the variance of the portfolio with the weight vector \(w\), while \(R^T w\) is the mean. The last constraint ensures the entire portfolio value is used (while still allowing short selling).

Efficient frontier theory assumes that the returns are stationary (i.e. their conditional variances tend to revert to their unconditional variances). Formally, stationarity of these returns requires that the joint distribution of \((r_{t_1}, r_{t_2}, \ldots, r_{t_n})\) is the same as the joint distribution of \((r_{t_1+h}, r_{t_2+h}, \ldots, r_{t_n+h})\) for all sets of time indices \(1 \leq t_1 < t_2 < \cdots < t_n\) and all integers \(h \geq 1\). The historical data allow an estimate the distributions of each return time series for each ETF to be made, and from there the optimization problem can determine the expected return and variance of the portfolio as a whole as a function of the portfolio weights.

The idea behind this is that risk and return must be balanced – seeking higher returns involves taking on higher risk, while seeking lower risk involves accepting lower returns. Rather than a linear relationship, the efficient frontier (the curve of each maximum possible return for all possible variances) is hyperbolic, because the covariance terms help to reduce the portfolio variance as the target return is increased along the efficient frontier.

This paper uses this concept, but alters it so as to maximize the variance of the portfolio, while minimizing the second-order variance. This is explained in the Methodology section under Markowitz-style Optimization.

**GARCH MODEL**

The generalized autoregressive conditional heteroskedasticity (GARCH) model is commonly used for financial time series due to the importance of variance in calculating derivative prices and
modeling risk. The assumption of stationarity is regular in financial models and is inherent in GARCH, though may not adequately capture extreme variation in returns. To alleviate this assumption, this paper uses the Student’s t-distribution in the GARCH model.

The GARCH model forecasts the future conditional variance of a time series and in this paper is used to forecast the volatility for each leveraged ETF for the last half an hour of trading.

The GARCH(1,1) model (with a Student t-distribution),

\[ \sigma_{\epsilon t+1}^2 = \omega + \alpha \epsilon_t^2 + \beta \sigma_t^2 \quad \text{s.t.} \quad X_t = \sigma_t \epsilon_t \quad \text{s.t.} \quad \epsilon_t \sim St(\nu), \quad E(\epsilon) = 0, \quad \text{and} \quad E(\epsilon \epsilon^T) = \frac{\nu}{\nu - 2} I, \]

is not the most sophisticated tool developed for the forecasting of volatility, but it is not significantly outperformed by some of the best models available, so it is chosen for the purposes of this paper (Hansen and Lunde, 2001). Covariance cannot be ignored in this modeling so a multivariate GARCH(1,1) model is chosen – BEKK(1,1), and a factor model is introduced to reduce the number of calculations required to forecast the volatility. Both are further explained in the Methodology section.

\[ * \quad \epsilon_t \text{ is a white noise process, } St(\nu) \text{ is the Student t-distribution with } \nu \text{ degrees of freedom, and } I \text{ is the identity matrix,} \begin{bmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix} \]
METHODOLOGY

DATA

Data collection and refinement is naturally the first and most tedious step. All historical data are collected from Bloomberg via their Excel plugin. For the purposes of this project, the data needed are minute-by-minute prices of the following leveraged ETFs,

- TNA - Direxion Daily Small Cap Bull 3X / TZA - Direxion Daily Small Cap Bear 3X
- FAS - Direxion Daily Financial Bull 3X / FAZ - Direxion Daily Financial Bear 3X
- DRN - Direxion Daily Real Estate Bull 3X / DRV - Direxion Daily Real Estate Bear 3X
- TQQQ - ProShares UltraPro QQQ (2X) / SQQQ - ProShares UltraPro Short QQQ (-2X)
- SOXL - Direxion Daily Semiconductor Bull 3X / SOXS - Direxion Daily Semiconductor Bear 3X
- ERX - Direxion Daily Energy Bull 3X / ERY - Direxion Daily Energy Bear 3X

...and the following underlying indices,

- Russell 2000 Index (TNA/TZA)
- Russell 1000 Financial Services Index (FAS/FAZ)
- MSCI US Real Estate Investment Trust (DRV/DRN)
- Nasdaq 100 (TQQQ/SQQQ)
- PHLX Semiconductor Sector Index (SOXS/SOXL)
- Russell 1000 Energy (ERY/ERX)

These leveraged ETFs were chosen for their high volatility as well as high trading volume (both are essential to the strategy – the former is explained later and the latter is needed in order to ensure trades are executed in a timely, i.e., within the minute, fashion). The S&P 500 is also used. The date ranges are October 27th, 2011 to March 21st, 2012 and September 27th, 2012 to April 5th, 2013. With the data downloaded into Excel, a Java program was created to "clean" the data. Java was chosen for
its data structures (namely Hash Maps), which simplify this process computationally. The Java data cleaner discards unwanted data (any after-hours trading) and uses linear interpolation to fill in any missing data points. For example, if the data point at 3:46pm is missing for TNA, but TNA is priced at 120.10 at 3:45pm and 120.30 at 3:47pm, the program will automatically assume that TNA was trading at 120.20 at 3:46pm. For any days in which a substantial amount of data is missing (or a specific ETF was not trading the entirety of the day), those days are discarded as a whole for all ETFs. All of this together ensures a table that when read in with MATLAB as a CSV will have each row match the same timestamp (as Bloomberg’s output does not do this).

**TRACKING ERROR AND MOMENTUM**

After this data refinement, the data can be easily imported into MATLAB for further processing. The next step is to determine the tracking error for each ETF at each minute. This need only be done at 3:30pm of each day as that is the only time at which the tracking error is checked. An additional column is created for each ETF to indicate the tracking error, including polarity (i.e. positive or negative), at 3:30pm of every trading day. The formula for tracking error is

\[
\text{Tracking error} = \text{Day's } \% \text{ price change of ETF} - (\text{Leverage of ETF} \times \text{Day's } \% \text{ price change of underlying index}).
\]

For example, if TZA (leverage of -3) is trading at a gain of 2.5% and the underlying Russell 2000 is trading at a loss of 1%, the tracking error will be 0.5%, indicating that TZA should be trading half a percentage point higher (according to the goal of the issuer, which in this case is either Direxion or ProShares).

In a quite similar manner (and in the same for-loop within the MATLAB code), the momentum of the leveraged ETF is calculated (the momentum simply being the day's return at the given time). By default, the algorithm will look at the momentum for the entire day, which is just the
day’s percentage price change at 3:30pm, but to look at, for instance, the momentum for just the last half of the day, i.e., the return from 12:30pm to 3:30pm, this column is needed. After the each momentum calculation – whether the return at 3:30pm is copied, or intra-day momentum is calculated, a third additional column is created to indicate whether or not the particular ETF should be traded. The condition for trading is based on the relationship between the tracking error and momentum. Under the assumption that momentum will tend to continue and tracking error will be at least partially corrected, if the momentum is opposite the tracking error and has an absolute value of at least 1%, the ETF is considered for trading. For example, if TNA is trading at a loss of 2.5% and a tracking error of 0.5%, it is expected to continue trading negatively because of the momentum and for this to be further intensified by the correction of a positive tracking error (i.e. it is trading too high for its leverage), so a short position will be taken on this ETF.

TIME SERIES

Continuing in MATLAB, variance (of the intraday returns) time series are constructed for each day for the interval between 10:00am and 3:30pm (the first half an hour of the day can’t have a variance as the variance is calculated using the last thirty minutes; the last half an hour of the day will not be used as when the algorithm is run in real-time - the last half an hour of the day is obviously not known). The variance is calculated at each minute using the last half an hour of trading data, therefore the formula will be

\[
\sigma_i^2 = \sum_{s=0}^{29} \frac{(r_{t,s} - \bar{r}_t)^2}{30}, \quad \text{s.t. } \bar{r}_t = \sum_{s=0}^{29} \frac{r_{t,s}}{30} \quad \text{and } r_t = \frac{\text{ETF Price at time } t - \text{ETF Price at time } t-1}{\text{ETF Price time } t-1}.
\]

The covariance (of the intraday returns of any two ETFs \(i\) and \(j\)) time series are calculated in a similar manner.

\[
\sigma_{i,j,t} = \sum_{s=0}^{29} \frac{(r_{i,t,s} - \bar{r}_{i,t})(r_{j,t,s} - \bar{r}_{j,t})}{30}
\]
The second-order variance is not something traditionally defined outside the use of this algorithm (in fact after much research, nothing with any resemblance closer than kurtosis was found, and this does not properly capture the desired information). The formula is as follows:

$$\tilde{\xi}_t = \sum_{s=0}^{29} \frac{(\sigma_{t-s} - \bar{\sigma}_t)^2}{30}, \text{ s. t. } \bar{\sigma}_t = \sum_{s=0}^{29} \frac{\sigma_{t-s}}{30}.$$

The second-order covariances are also calculated in a similar manner:

$$\tilde{\xi}_{ij,kl,t} = \sum_{s=0}^{29} \frac{(\sigma_{ij,t-s} - \bar{\sigma}_{ij,t})(\sigma_{kl,t-s} - \bar{\sigma}_{kl,t})}{30}.$$

This implies $N^4$ (or $N(N+1)(N+2)(N+3)/4!$ without repeats) second-order covariances as opposed to $N^2$ (or $N(N+1)/2$ without repeats) covariances. Because of this method of calculation, the second-order variance is of course only calculated for 10:30am to 3:30pm. Note that half an hour is chosen as the period for variance because the ETFs will be held at the end of the day for roughly half an hour. Given that the structure of these formulae match, it is feasible to apply standard GARCH techniques. Unfortunately given the magnitude of the terms and their interrelations, such a GARCH model would rapidly become cumbersome computationally. Fortunately, the Engle test for heteroscedasticity shows the variance time series to be fairly homoscedastic, thus eliminating the need for a model accounting for conditional variance. The Engle test returns a p-value to test the significance of the presence of heteroscedasticity in the given data set. Upon sampling the variances, the p-values returned results quite higher than any acceptable level (p-levels ranged around 0.4 and greater).

A multivariate GARCH model with factors uses the time series data to forecast the volatility for the last half an hour of trading (i.e. when positions in the ETFs will be held). The advantage (and necessity) of a multivariate GARCH model is the interaction between the individual univariate GARCH processes for each ETF. The covariances cannot be treated as constant in the GARCH model, nor can they be ignored in the context of this problem as they contribute heavily to the overall
variance of the portfolio holdings. Of the several methods available for multivariate GARCH, BEKK (Baba, Engle, Kroner and Kraft) is chosen for the purposes of this project. (The two asset case of the BEKK model is given below.) The number of parameters in a BEKK model with a mere five return series is an astounding 65 and when it increases to 10, it becomes over 250! To severely alleviate this computationally expensive model, factors are introduced. In a multivariate GARCH factor model, a smaller set of factors are used to create a multivariate system, and then applied to the original processes as univariate GARCH models as explained below.

The factors chosen are the S&P500 and VIX after a few tests of goodness of fit on various possible standard factors (including the French and Fama factors, i.e., the difference in the market and risk-free rates, difference in small and big market capitalization returns, and difference in high and low book-to-market ratio returns). These are fit to a BEKK(1,1) GARCH model (for a total of only 11 parameters), which is defined as follows:

\[
\sigma_{t,F_1}^2 = a_{0,11} + a_{1,11}^2 F_{t-1,1}^2 + 2a_{1,11}a_{1,12}F_{t-1,1}F_{t-1,2} + a_{1,12}^2 F_{t-1,2}^2 + b_{1,1}^2 \sigma_{t-1,1}^2 + 2b_{1,1}b_{1,2} \sigma_{t-1,2}^2
\]

\[
\sigma_{t,F_1F_2} = a_{0,12} + (a_{1,11}a_{1,22} + a_{1,12}a_{1,21})F_{t-1,1}F_{t-1,2} + a_{1,11}a_{1,21}F_{t-1,1}^2 + a_{1,12}a_{1,12}^2 F_{t-1,2}^2
\]

\[
+ (b_{1,1}b_{2,2} + b_{1,2}b_{2,1}) \sigma_{t-1,1}^2 + b_{1,1}b_{2,1} \sigma_{t-1,2}^2 + b_{2,2}b_{2,1} \sigma_{t-1,2}^2
\]

\[
\sigma_{t,F_2}^2 = a_{0,22} + a_{1,22}^2 F_{t-1,2}^2 + 2a_{1,22}a_{1,21}F_{t-1,1}F_{t-1,2} + a_{1,21}^2 F_{t-1,1}^2 + b_{2,2}^2 \sigma_{t-1,1}^2 + 2b_{2,2}b_{2,1} \sigma_{t-1,2}^2
\]

where

\[
F_t = \begin{cases} 
F_{t,1} = r_{t,S&P500} = \frac{\text{Value of S&P at time } t - \text{Value of S&P at previous close}}{\text{Value of S&P at previous close}} \\
F_{t,2} = r_{t,VIX} = \frac{\text{Value of VIX at time } t - \text{Value of VIX at previous close}}{\text{Value of VIX at previous close}}.
\end{cases}
\]

The ETFs are then fit to the factors using regression techniques:

\[
X_t = a + BF_t + \epsilon_t \tag{1}
\]
Once this the regression is completed and the mGARCH model for the factors is constructed, the next 30 time steps (i.e. the last half an hour of the trading day) can be calculated to determine the forecast of the volatility of each ETF for the trading period using the following system:

$$\Sigma_{x,t|t-1} = B \Sigma_{F,t|t-1} B' + \Sigma_{e,t|t-1}$$

where $$\Sigma_{F,t|t-1}$$ is a forecast of the covariance matrix of the factors and $$\Sigma_{e,t|t-1}$$ is a diagonal matrix made up of the variance forecasts of the univariate residuals from (1).

**MARKOWITZ-STYLE OPTIMIZATION**

The purpose of obtaining high volatility is twofold: greater momentum and greater tracking error to be corrected at the end of the day. In Posterro’s paper, a decrease in market volatility proved highly detrimental to his strategy’s performance. To address this issue, this paper seeks a method of targeting the highest volatility possible. In order to ensure consistently high volatility for the portfolio of leveraged ETFs, the optimization problem becomes to minimize the variability of the variance itself as defined by

$$\mathbb{E}[(w \otimes w) : (w \otimes w)] = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} w_i w_j w_k w_l \Xi_{i,j,k,l}$$  \hspace{1cm} (2)^\dagger$$

s.t. $$\Xi = \sum_{i,j,k,l=1}^{N} \text{cov}(\sigma_{ij}, \sigma_{kl})(\hat{e}_i \otimes \hat{e}_j \otimes \hat{e}_k \otimes \hat{e}_l)$$ \hspace{1cm} i.e. $$\Xi_{i,j,k,l} = \xi_{ijkl}^\dagger$$

The weights given to each ETF are sought to minimize the above, given that the variance of the portfolio is equal to or greater than 90% of the highest variance of the selected ETFs for the day. Higher than this target risks the algorithm going “all-in” for a single ETF, whereas lower than this

\[\dagger\quad \otimes \text{ is the dyadic tensor product s.t. } a \otimes b = \begin{bmatrix} a_1 b_1 & \cdots & a_1 b_n \\ \vdots & \ddots & \vdots \\ a_n b_1 & \cdots & a_n b_n \end{bmatrix}.\]

\[\dagger\quad \hat{e}_i \text{ is the standard basis vector, i.e., it denotes the vector with a 1 in the } i^{th} \text{ coordinate and 0’s elsewhere.}\]

\[\dagger\quad \xi_{ijkl} \text{ is defined as it is in the ‘Time Series’ section } \left(\sum_{s=0}^{29} (\sigma_{ij,t-s}-\bar{\sigma}_{ij,t})(\sigma_{kl,t-s}-\bar{\sigma}_{kl,t})\right).\]
risks the algorithm giving too low of a variance. The variance of the portfolio return is defined intuitively as

\[ w^T \Sigma w \quad \text{s.t.} \quad \Sigma_{i,j} = \sigma_{i,j}^{\$} \]

The minimization problem then becomes:

\[
\min_{w} (w \otimes w) : \Xi : (w \otimes w) \quad \text{s.t.} \quad \left\{ \begin{array}{l}
(a) \quad w^T \Sigma w \geq 90\% \ast \max \text{diag} \Sigma \\
(b) \quad \sum_{i=1}^{N} w_i = 1 \text{ or } \left( \sum_{i=1}^{N} w_i = -0.5 \text{ and } w_i < 0 \, \forall \, i \in \{1, \ldots, N\} \right) ^{**} \\
(c) \quad -0.5 \leq w_i \leq 1 \, \forall \, i \in \{1, \ldots, N\}
\end{array} \right.
\]

with the additional constraints

\[-0.5 \ast 1_N(i) \leq w_i \leq 1_p(i) \quad \text{s.t.} \quad 1_N(i) = \begin{cases} 1 & \text{if Tracking error of ETF } i > 0 \text{ and } r_i < -0.01 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad 1_p(i) = \begin{cases} 1 & \text{if Tracking error of ETF } i < 0 \text{ and } r_i > 0.01 \\ 0 & \text{else} \end{cases}\]

Because this problem is minimizing a multivariate polynomial of degree of four, it has one and exactly one global minimum (albeit multiple local minimums as with the original Markowitz optimization). As with the GARCH calculations, for each day, only those ETFs which meet the condition of opposite momentum and tracking error are considered for a non-zero weight in the portfolio. In addition, the weight of any points chosen to be non-zero must have the same sign as the momentum of the ETF (i.e. an ETF that has a positive return for the day will have a long position, and one with a negative return will have a short position).

**PURCHASE AND SELL ORDERS**

Once the portfolio weights are determined for each day, the positions are simulated to be opened at 3:30pm and closed at 3:59pm (the latter time to ensure that the position will not be held

\^{\$} \sigma_{ij} \text{ is defined as it is in the 'Time Series' section } \left( \sum_{t=0}^{\text{lag}} (r_{t+5} - r_{t+3})(r_{t+5} - r_{t+3}) \right). \\
^{**} \text{The second half of constraint b) is in case there is a day with only short positions.}
overnight). In a real-time execution of this strategy, the market orders would be set at 3:30 and 
would be good for one-minute. If a market order did not execute within one minute of being 
entered, the order would be cancelled, assuming a trading volume too low to risk holding an 
overnight position in the case that the sell order also took a long time to execute at 3:59pm.

With any trading account, whether corporate or personal, there are margin limits and thus 
it makes sense to apply similar limits to the short-selling. For the purposes of this project, a 
practical lower limit of -50% and upper limit of 100% is applied to each element of the portfolio. 
Generally when creating a portfolio such a large tolerance for each weight would be considered 
risky, but because there are only a limited number of ETFs each day that are going to be included in 
the portfolio, quite easily only one, and this is already an inherently speculative trading strategy, 
such a constraint is considered acceptable.
RESULTS

IMPLEMENTATION

In the process of coding and testing, the following noteworthy items were encountered.

In the Markowitz-style optimization of the new strategy, the MATLAB function *fmincon* is used (see Appendix). As detailed in the methodology section, the function to be minimized is (2) and the constraints given are that the variance is at least 90% of the maximum ETF variance, the total weights equal 100%, and no single position is less than -50% or more than 100% of the portfolio’s value. The interior-point algorithm (selected from the MATLAB Optimization Toolbox) with a starting point of an equally weighted portfolio gave the best results for the optimization problem, with the default algorithm for *fmincon* (trust-region-reflective) unable to handle inequality constraints. The default starting point for *fmincon* often resulted in the optimization algorithm generating error codes and ending prematurely.

The GARCH modeling proved to be quite computationally expensive as expected, even with the addition of a factor model. Fortunately, no single day’s calculations took longer than a minute, which means when the algorithm is run in real time, the data retrieved at 3:29pm can be used for the orders placed by 3:30pm. The minute-by-minute data for the VIX was oddly not available from Bloomberg, so only the S&P500 returns were used as a factor in the GARCH factor model. In a comparison of the new strategy with and without GARCH forecasting, while the cumulative return at the end of the period showed an improvement of over 3%, a significance test on the two mean returns returned a p-value of 0.41, i.e. the GARCH had no statistically significant improvement. Because of this conclusion, GARCH forecasting is deemed unnecessary in the context of this optimization problem (especially for how computationally expensive it is).

†† The MATLAB code was all run on a Windows 7 system with a 2.60GHz Intel processor and 8GB of available RAM.
ANALYSIS

To test the effectiveness of the strategy, several benchmarks were coded and tested for comparison. First and foremost, a standard benchmark of the S&P500 is used, with the alteration that it is only being bought at 3:30pm and sold at 3:59pm as it is only fair to judge the benchmark on the time that the strategy is used. For further comparison, a buy and hold (i.e. buy once at the beginning of the period and sell only at the end) S&P500 benchmark is included.

The second benchmark was the original Markowitz optimization mentioned in the background section: targeting the ETFs with the highest returns for the day while trying to minimize the variance of the holdings as a portfolio. The Markowitz strategy followed the same time constraint as the S&P for when it was tested and allows short-selling to the same extent that is granted to the new strategy. Again, for comparison, a Markowitz strategy utilizing the entire day is used; it functions the same as the buy and hold benchmark, but there is a daily rebalancing just before closing.

Thirdly and most importantly to this paper, a benchmark of the original Posterro strategy is conducted. Because the Posterro paper only uses one pair of ETFs (FAS/FAZ), the Posterro strategy is run two ways: a) only FAS/FAZ and no short-selling is allowed as within the original paper; b) with all ETFs considered in an equal weight portfolio (i.e. if a leveraged ETF fits the parameters of the “discount and up” condition it is included in the portfolio for that day, and of those included ETFs, equal weights are given to all) and short selling is allowed (i.e. any ETF which qualifies for the opposite of the “discount and up” – trading down for the day, and at a premium with regard to the tracking error – is also considered in the portfolio, just as with the new strategy). ‘B)’ is referred to as Posterro+. Below is a table with the p-values measuring the significance of the difference between the mean daily returns of each benchmark case with the new strategy. In addition are the
means and volatilities of the returns for each case, the cumulative return, and finally a rough estimate of the annualized percentage return (simply the mean return to the power of 252).

<table>
<thead>
<tr>
<th>Strategy name</th>
<th>p-value measuring significance in difference of mean return to new strategy (α = 0.10)</th>
<th>Mean of daily returns</th>
<th>Volatility (standard deviation of daily returns)</th>
<th>Cumulative return</th>
<th>Est. APR (252 trading days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S&amp;P500</td>
<td>0.006 – significantly lower</td>
<td>0.011%</td>
<td>0.191%</td>
<td>1.35%</td>
<td>2.81%</td>
</tr>
<tr>
<td><em>buy and hold</em></td>
<td>0.081 – significantly lower</td>
<td>0.065%</td>
<td>0.726%</td>
<td>8.37%</td>
<td>17.79%</td>
</tr>
<tr>
<td>Markowitz</td>
<td>0.012 – significantly lower</td>
<td>0.005%</td>
<td>0.496%</td>
<td>0.50%</td>
<td>1.27%</td>
</tr>
<tr>
<td>“buy and hold”</td>
<td>0.025 – significantly lower</td>
<td>-0.617%</td>
<td>4.668%</td>
<td>2.63%</td>
<td>-78.98%</td>
</tr>
<tr>
<td>Posterro</td>
<td>0.045 – significantly lower</td>
<td>0.066%</td>
<td>0.335%</td>
<td>8.45%</td>
<td>18.09%</td>
</tr>
<tr>
<td>Posterro+</td>
<td>0.483 – insignificantly diff.</td>
<td>0.196%</td>
<td>1.566%</td>
<td>25.33%</td>
<td>63.80%</td>
</tr>
<tr>
<td>New strategy</td>
<td></td>
<td><strong>0.203%</strong></td>
<td><strong>0.828%</strong></td>
<td><strong>27.76%</strong></td>
<td><strong>66.70%</strong></td>
</tr>
<tr>
<td>without GARCH</td>
<td>0.409 – insignificantly diff.</td>
<td>0.179%</td>
<td>0.806%</td>
<td>24.08%</td>
<td>56.94%</td>
</tr>
</tbody>
</table>

Table 1: Benchmark comparison results for Sept. 27th, 2012 – April 5th, 2013

It is readily apparent from the table that the first five strategies fall quite short of the performance of the new strategy, but the most important detail to notice is that the new strategy is able to reduce the volatility of Posterro+ by nearly a factor of two while having a roughly equivalent mean return.

<table>
<thead>
<tr>
<th>Strategy name</th>
<th>p-value measuring significance in difference of mean return to new strategy (α = 0.10)</th>
<th>Mean of daily returns</th>
<th>Volatility (standard deviation of daily returns)</th>
<th>Cumulative return</th>
<th>Est. APR (252 trading days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markowitz</td>
<td>0.465 – insignificantly diff.</td>
<td>0.015%</td>
<td>0.454%</td>
<td>1.41%</td>
<td>3.85%</td>
</tr>
<tr>
<td>“buy and hold”</td>
<td>N/A (greater than)</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Posterro</td>
<td>0.338 – insignificantly diff.</td>
<td>-0.027%</td>
<td>0.511%</td>
<td>-2.66%</td>
<td>-6.58%</td>
</tr>
<tr>
<td>Posterro+</td>
<td>0.010 – significantly lower</td>
<td>-0.270%</td>
<td>2.695%</td>
<td>-25.79%</td>
<td>-49.41%</td>
</tr>
<tr>
<td>New strategy‡‡</td>
<td></td>
<td>0.026%</td>
<td><strong>1.309%</strong></td>
<td><strong>1.71%</strong></td>
<td><strong>6.77%</strong></td>
</tr>
</tbody>
</table>

Table 2: Benchmark comparison results for Oct. 27th, 2011 – March 21st, 2012

Interestingly in the second set of data, the new strategy now performs statistically-significantly better than Posterro+ but not than the original Posterro strategy (and again has well over twice the standard deviation). This suggests that the additions made for Posterro+, i.e., more ETFs and the “down and premium” short selling, magnify both the return and risk of the original,

‡‡ S&P500 minute-by-minute data were not available for testing for this period, therefore the S&P benchmark is unavailable, and the GARCH forecasting could not be used in the testing of the older data set.
but the new strategy’s optimization techniques serve to curb this increase in volatility, while still improving the overall return.

**SHARPE RATIO**

Another way to look at this is via the Sharpe ratio, \( S = \frac{E(R-R_f)}{\text{std}(R-R_f)} \), where the risk-free rate \( R_f \) is proxied by the federal funds rate. The table below illustrates that the new strategy has almost double the Sharpe ratio over the equivalent Posterro+, though interestingly even with such a substantial difference, it is not statistically significant. The following formula is used to determine the variance of the difference of two Sharpe ratios,

\[
Var_{\text{diff}} = 1 + \frac{SR_a^2}{4} \left[ \frac{\mu_{a,1}^2}{\sigma_a^4} - 1 \right] - SR_a \frac{\mu_{a,2}}{\sigma_a^3} + 1 + \frac{SR_b^2}{4} \left[ \frac{\mu_{b,1}^2}{\sigma_b^4} - 1 \right] - SR_b \frac{\mu_{b,2}}{\sigma_b^3}
\]

\[-2 \left[ \rho_{a,b} + SR_a SR_b \left[ \frac{\mu_{a,1} \mu_{b,1}}{\sigma_a^2 \sigma_b^2} - 1 \right] - \frac{1}{2} SR_a \frac{\mu_{a,2}}{\sigma_a \sigma_b^2} - \frac{1}{2} SR_b \frac{\mu_{b,2}}{\sigma_b \sigma_a^2} \right] \]

and from there the standard error of the difference can be calculated as

\[
SE (SR_a - SR_b) = \frac{\sqrt{Var_{\text{diff}}}}{\sqrt{T}}
\]

in order to test the significance of each difference. (J.D. Opdyke, 2007)

<table>
<thead>
<tr>
<th>Strategy name</th>
<th>Sharpe ratio</th>
<th>p-value measuring significance in difference of mean return to new strategy (( \alpha = 0.10 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>S&amp;P500</td>
<td>0.0576</td>
<td>0.088 – significantly lower</td>
</tr>
<tr>
<td>Markowitz</td>
<td>0.0101</td>
<td>0.0003 – significantly lower</td>
</tr>
<tr>
<td>Posterro</td>
<td>0.1970</td>
<td>0.178 – insignificantly different</td>
</tr>
<tr>
<td>Posterro+</td>
<td>0.1252</td>
<td>0.113 – insignificantly different</td>
</tr>
<tr>
<td>New strategy</td>
<td><strong>0.2452</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Sharpe ratio analysis for Sept. 27\(^{th}\), 2012 - April 5\(^{th}\), 2013

\(\rho_{a,b} = \frac{E[(a-E(a))(b-E(b))]}{\sigma_a \sigma_b}\) is the correlation between \(a\) and \(b\). \(\mu_{3a} = E \left[ (a - E(a))^3 \right]\) and \(\mu_{4a} = E \left[ (a - E(a))^4 \right]\) are the third and fourth central moments of \(a\). \(\mu_{2a,2b} = E \left[ (a - E(a))^2 (b - E(b))^2 \right]\) is the joint second central moment of the joint distribution of \(a\) and \(b\). \(\mu_{1a,2b} = E \left[ (a - E(a))(b - E(b))^2 \right]\) and \(\mu_{1b,2a} = E \left[ (b - E(b))(a - E(a))^2 \right]\).
Adding the older data set gives a p-value for comparing the Sharpe ratios of Posterro+ and the new strategy decreased to within a significant threshold (assuming an alpha of 10%).

<table>
<thead>
<tr>
<th>Strategy name</th>
<th>Sharpe ratio</th>
<th>p-value measuring significance in difference of mean return to new strategy (α = 0.10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posterro</td>
<td>0.0602</td>
<td>0.121 – insignificantly different</td>
</tr>
<tr>
<td>Posterro+</td>
<td>-0.0045</td>
<td>0.091 – significantly lower</td>
</tr>
<tr>
<td>New strategy</td>
<td>0.1169</td>
<td>-</td>
</tr>
</tbody>
</table>


RETURNS

The chart below illustrates the cumulative returns of the four strategies giving a visual perspective of the respective volatilities of each strategy. The lower volatility of the new strategy relative to Posterro+ is readily apparent, though the new strategy is deceptively superior to the remaining benchmarks judging from this chart alone.

Chart 1: Cumulative returns of benchmarks and new strategy for Sept. 27th, 2012 - April 5th, 2013
The second chart below shows the less successful of the two data sets. The Posterro+ having the highest variability suffers the greatest, but the new strategy despite having the second highest variability remains in the lead for performance, albeit understandably by less.

![Chart 2: Cumulative returns of benchmarks and new strategy for Oct. 27th, 2011 – March 21st, 2012](image)

The distribution of the returns for the new strategy is as shown in the histogram below, as well as an overlay of fitted normal and Student t-distributions.
From the histogram, the distribution appears to have a more heavily weighted right tail (i.e. more positive returns) though has one low outlier (a loss of -5.28%). The skewness (or third standardized moment), as defined by

$$\gamma_1 = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right],$$

of the return distribution is equal to -0.3047, suggesting a negative skew, but if the sole outlier is ignored, the skewness is instead a positive 0.2673. From a chi-square goodness of fit test, the data fit neither a normal distribution nor a Student t-distribution, having p-values on the magnitude of $10^{-4}$ for both tests. Interestingly, though the data do not follow either, the normal distribution shows a better fit than the Student t-distribution, suggesting the data follow a distribution more ideal than most return distributions (assumed to be Student t).

For comparison, the following two charts show the distributions of the original Posterro strategy as well as Posterro+.


The skewnesses of Posterro and Posterro+ are -0.2001 and -0.9159. While the negative skewness of Posterro is partially due to the high number of days of inactivity, resulting in a return of 0% on each of those days, but the Posterro+ has several catastrophic daily losses ranging into the double digits.
The Markowitz strategy gives a distribution fitting well enough to the Student t-distribution (with 20 degrees of freedom) that the goodness of fit test fails to reject the null hypothesis that the data come from such a distribution. It also presents a slightly-negative skewness of -0.1540.
The daily S&P500 returns are clustered very tightly, not quite matching either normal or Student t and having a heavy skewness of -0.5789.

**INFLUENCE OF VOLATILITY**

Lastly, a comparison of the time series of returns to the time series of volatilities (i.e. the volatilities for each day at 3:29pm) in the new strategy to the portfolio is presented in the figures below.

The chart seems to suggest a highly inconsistent pattern of higher volatility days having higher returns, but the next chart shows that the slightly positive correlation between the two variables is not statistically significant, having an $R^2$ value of only 0.0811.

*** Inactive days (16) excluded from graph.
While this correlation proves little, subtracting the S&P500 returns from the new strategies returns, and then comparing the difference to new strategy volatilities, gives an improved, albeit still weak, $R^2$ value of 0.1044. The purpose of this is to try to remove the typical negative correlation in the market between volatility and return. Below is a chart showing the slightly improved correlation.

**Chart 8: Comparison of return vs. variance for new strategy for Sept. 27th, 2012 - April 5th, 2013†††**

††† Inactive days (16) excluded from graph.
Chart 9: Comparison of difference in returns of new strategy and S&P500 vs. variance for new strategy for Sept. 27th, 2012 - April 5th, 2013

As an alternative way to examine the relationship between volatility and the new strategy’s performance, a volatility index called the VIX is compared to the cumulative return of the strategy for both periods. The VIX, sometimes known as the “fear index,” is based on the implied volatility of options on stocks in the S&P500 index, and thus it serves as an excellent measure of market risk.
The two charts above again suggest an inconsistent, but nevertheless present, positive correlation between how the new strategy performs and its daily volatility, which is directly affected in large part by the volatility of the market.
The attempt to improve upon Posterro’s strategy for leveraged ETF trading proved to be quite successful. The quartic optimization was able to create daily portfolios from the ETFs fitting the “discount and up” or “premium and down” conditions to significantly tighten the distribution of the returns through a reduction of the strategy’s volatility and curbing of heavier losses, as well as obtain an impressively high Sharpe ratio, even during a period when the strategy otherwise resulted in a loss.

Compared to the standard “buy-and-hold” strategies, the new strategy showed a reasonable daily return volatility of around 1%, not much higher than the S&P500 and much lower than Markowitz daily rebalancing, while maintaining a significantly higher return than either. The new strategy’s Sharpe ratio showed to be much higher than the Sharpe ratios of either versions of the Posterro strategy (albeit just outside the range of statistically significant for the original).

We were able to confirm evidence that market volatility affects the performance of the strategy, as a steady and heavy decline in the VIX in the earlier dataset led to increasingly lower returns, but the optimization algorithm successfully kept the strategy afloat, preventing the mean daily loss of over a quarter of a percent observed in the Postermo+ and instead giving a positive return of almost twice that of the equivalent Markowitz strategy for the same period.

Furthermore, while we are unaware of another obvious use for such a quartic optimization to maximize volatility, if such a problem arises, perhaps a strategy based on the VIX or taking advantage of the tracking error of another financial instrument, this tool can easily be implemented therein.
REFERENCES


APPENDIX

A. MATLAB CODE

I. MAIN BODY OF DATA TESTING

```matlab
function [weights, ret, vols] = thdataproc( markowitz, GARCHoff, posterro, SP, dataset, maxvoltarget, dispOn, shortsell)

%% Initialize variables
% index variables
hr = 4;
mn = 5;
dyc = 3; % day count (different than date)

% program parameters
{%
markowitz = false;
GARCHoff = false;
posterro = false;
SP = false;
dispOn = false;
dataset = 2;
maxvoltarget = .9;
%}
maxcount = 37927;
if(dataset == 2)
    maxcount = 48093;
end

% storage variables
covars = 0; % ('day count','hr','mn','1.ETF#1','1.ETF#2')=covar(1,2)
dmom = 0; % daily momentums, ('day count','1.ETF#')=return of 1.ETF @ 15:29
terr = 0; % tracking error, ('day count','1.ETF#')=actual-target @ 15:29
ETFrettemp = 0; % to hold day's returns for GARCHscript
retcount = 1;
vols = 0; % to hold daily volatility of portfolio

%results
weights = 0; % ('day count','1.ETF#')=weight for 1.ETF
ret = 0; % ('day count'-1)=return for day
totalRet = 1;

%% Read in file
data = 0;
if(dataset == 1)
    data = csvread('ETFformatted.csv',1,1);
else
    data = csvread('ETFformatted2.csv',1,1);
end

n = size(data);
% ETF data starts @ col 6, index then +ETF then -ETF
if(posterro)
```

31
disp('Running Posterrob algorithm with equal weights...')
elseif(markowitz)
    disp('Running basic Markowitz algorithm...')
elseif(SP)
    disp('Running S&P500 benchmark...')
elseif(GARCHoff)
    disp('Running new strategy...')
else
    disp('Running new strategy with GARCH forecasting...')
end
if(shortsell)
    disp('Short selling allowed.')
else
    disp('Short selling not allowed.')
end
disp(['Using data set ', num2str(dataset)])

for i=2:maxcount
    % Prep data
    if(data(i,hr) == 16)
        ETFrettemp = 0;
        SP500 = 0;
        retcount = 1;
    elseif((data(i,hr) < 15 || data(i,mn) < 31) && (data(i,mn) == 30 ||
        data(i,hr) > 10 || data(i,mn) >= 30))
        for j=1:6
            ETFrettemp(retcount,j*2-1) = data(i,6+(j-1)*3+1)-1;
            ETFrettemp(retcount,j*2) = data(i,6+(j-1)*3+2)-1;
        end
        if(dataset == 2)
            SP500(retcount) = data(i,24)-1;
        end
        retcount = retcount + 1;
    end
if(data(i,hr) > 9 && (data(i,hr) < 15 || data(i,mn) < 30))
    halfhourdatatemp = [(data((i-29):i,6+(l-1)*3+1))';(data((i-
        29):i,6+((l-1)*3+2))'';
        (data((i-29):i,6+(2-1)*3+1))';(data((i-29):i,6+(2-1)*3+2))'';
        (data((i-29):i,6+(3-1)*3+1))';(data((i-29):i,6+(3-1)*3+2))'';
        (data((i-29):i,6+(4-1)*3+1))';(data((i-29):i,6+(4-1)*3+2))'';
        (data((i-29):i,6+(5-1)*3+1))';(data((i-29):i,6+(5-1)*3+2))'';
        (data((i-29):i,6+(6-1)*3+1))';(data((i-29):i,6+(6-1)*3+2))''];
    covtemp = cov(halfhourdatatemp);
    for j=1:12
        for k=1:12
            covars(data(i,dyc),data(i,hr),data(i,mn)+1,j,k) =
            covtemp(j,k);
            % creates the covariance matrix for each half hour period
        end
    end
    %
    % only for heteroskedasticity test
    if(data(i,hr) == 14 && data(i,mn) > 30 && data(i,mn) < 60) ||
    data(i,hr) > 10 || data(i,mn) >= 30)
    12 = 0;
for t=1:30
    l2(t) = covars(data(i,dyc),data(i,hr),t+data(i,mn)-30,1,1);
end
templ = cov(l2,l2);
secCovs(data(i,dyc),data(i,hr)+(data(i,hr)-14.5)*60) =
l2(end);  % templ(1,1);  %(data(i,6+(l-1)*3+1))' ;
end
%
if(data(i,hr)==15 && data(i,mn) == 29)
    for j=1:6
        dmom(data(i,dyc),j*2-1) = data(i,6+(j-1)*3+1)-1;
        dmom(data(i,dyc),j*2) = data(i,6+(j-1)*3+2)-1;
        if(j~=4)  % for the x2 ETF
            terr(data(i,dyc),j*2-1) = (data(i,6+(j-1)*3+1)-1)-3*(data(i,6+(j-1)*3)-1);
            terr(data(i,dyc),j*2) = (data(i,6+(j-1)*3+2)-1)+3*(data(i,6+(j-1)*3)-1);
        else
            terr(data(i,dyc),j*2-1) = (data(i,6+(j-1)*3+1)-1)-2*(data(i,6+(j-1)*3)-1);
            terr(data(i,dyc),j*2) = (data(i,6+(j-1)*3+2)-1)+2*(data(i,6+(j-1)*3)-1);
        end
    end
end
if(data(i,hr)==15 && data(i,mn) == 59)
%}
% Equal weighting
for j=1:6
    weights(data(i,dyc),j*2-1)=0/6;
    weights(data(i,dyc),j*2)=1/6;
end
count=1;
%
if(markowitz)
% Markowitz optimization for control group
expRet = 0;
count = 1;
for j=1:6
    expRet(j*2-1)=data(i-30,6+(j-1)*3+1);
    expRet(j*2)=data(i-30,6+(j-1)*3+2);
end
target = 1.03;
covMat = 0;
for j=1:12
    for k=1:12
        covMat(j,k) = covars(data(i,dyc),data(i,hr),data(i-30,mn),j,k);
    end
end
weighttemp = quadraticsolver(target, covMat, expRet, 0);
for j=1:12
    weights(data(i,dyc),j) = weighttemp(j);
end
vols(data(i,dyc)-1) = (weights(data(i,dyc),1:12)*covMat*weights(data(i,dyc),1:12)');

elseif (~SP)
    % New method
    % determine "tradability"
    totrade = 0;
    totrade2 = zeros(12,1);
    count = 0;
    lower = 0;
    upper = 0;
    for j=3:4
        if(shortsell)
            if(sign(dmom(data(i,dyc),j))==sign(terr(data(i,dyc),j))
                && ((abs(dmom(data(i,dyc),j)) > .01 && abs(terr(data(i,dyc),j)) > .0001))
                    count=count+1;
                    totrade(count)=j;
                    totrade2(j)=1;
                    if(dmom(data(i,dyc),j) > 0)
                        upper(count) = 1;
                        lower(count) = 0;
                    else
                        upper(count) = 0;
                        lower(count) = -0.5;
                    end
        end
    end
    else
        if(sign(dmom(data(i,dyc),j)) == sign(terr(data(i,dyc),j))
            && ((dmom(data(i,dyc),j)) > .01 && abs(terr(data(i,dyc),j)) > .0001))
                count=count+1;
                totrade(count)=j;
                totrade2(j)=1;
                upper(count) = 1;
                lower(count) = 0;
        end
    end
end

if(count > 0)
    nonGARCHcovMat = 0;
    % construct covariance matrix:
    for j=1:count
        for k=1:count
            nonGARCHcovMat(j,k)=covars(data(i,dyc),data(i,hr),data(i-30,mn),totrade(j),totrade(k));
        end
    end
    if(posterro)
        if(sum(lower)==0)
            m = sum(upper)-sum(lower./min(lower));
            if(m == 0)
                m = 2;
            end
            count2 = 1;
            for j=1:12
                if(totrade2(j)==0)
                    weights(data(i,dyc),j) = 0;
                end
            end
        end
    end
end

34
else
    if (upper(count2)==0)
        weights(data(i,dyc),j) = -1/m;
    else
        weights(data(i,dyc),j) = 1/m;
    end
    count2 = count2 + 1;
end
else
    for j=1:12
        if (totrade2(j)==0)
            weights(data(i,dyc),j) = 0;
        else
            weights(data(i,dyc),j) = 1/count;
        end
    end
end
vols(data(i,dyc)-1) =
((weights(data(i,dyc),1:count)*nonGARCHcovMat*weights(data(i,dyc),1:count)'))
^(0.5);
else
    covMat = 0;
    if (GARCHoff)
        covMat = nonGARCHcovMat;
    else
        %GARCHscript call
        ETFreturns = 0;
        for j=1:count
            for k=1:length(ETFrettemp)
                ETFreturns(k,j) = ETFrettemp(k,totrade(j));
            end
        end
        covMat = GARCHscript(ETFreturns,SP500);
    end
    % calculate 2nd order covariances:
    secCovMat = 0;
    for j=1:count
        for k=1:count
            jkc = 0;
            for t=1:30
                jkc(t) =
                covars(data(i,dyc),data(i,hr),t,totrade(j),totrade(k));
            end
        end
        for l=1:count
            for m=1:count
                lmc = 0;
                for t=1:30
                    lmc(t) =
                    covars(data(i,dyc),data(i,hr),t,totrade(l),totrade(m));
                end
                tempc = cov(jkc,lmc);
                secCovMat(j,k,l,m) = tempc(1,2);
            end
        end
    end
end
weighttemp = 
quarticsolver2(max(diag(covMat))*maxvoltarget,secCovMat,covMat,lower,upper);

% Test for Markowitz on only strategy selected ETFs
expRett = 0;
for j=1:6
    expRett(j*2-1)=data(i-30,6+(j-1)*3+1);
    expRett(j*2)=data(i-30,6+(j-1)*3+2);
end

countR = 1;
expRet = 0;
for j=1:12
    if(totrade2(j))
        expRet(countR) = expRett(j);
        countR = countR+1;
    end
end

weighttemp = quadraticsolver(1.02,covMat,expRet,1);

vols(data(i,dyc)-1) =
((weighttemp(1:count)*nonGARCHcovMat*weighttemp(1:count)')).^(0.5);/%max(diag(covMat))

count = 1;
for j=1:12
    if(totrade2(j)==0)
        weights(data(i,dyc),j) = 0;
    else
        weights(data(i,dyc),j) = weighttemp(count);
        count = count + 1;
    end
end

elseif(SP)
    count = 0;
end

ret(data(i,dyc)-1)=0;
if(count > 0)
    % Calculate results from weights
    for j=1:6
        ret(data(i,dyc)-1)=ret(data(i,dyc)-1)+data(i,6+(j-1)*3+1)/data(i-29,6+(j-1)*3+1)*weights(data(i,dyc),j*2-1);
        ret(data(i,dyc)-1)=ret(data(i,dyc)-1)+data(i,6+(j-1)*3+2)/data(i-29,6+(j-1)*3+2)*weights(data(i,dyc),j*2);
    end
    %sum(weights(data(i,dyc),:))
end

if(ret(data(i,dyc)-1)==0)
    if(~SP)
        ret(data(i,dyc)-1)=1;
    else
        ret(data(i,dyc)-1)=ret(data(i,dyc)-1)+data(i,24)/data(i-29,24);
    end
elseif(posterro == markowitz && sum(weighttemp(1:count-1)) < 1)
\[ \text{ret}(\text{data}(i,dyc)-1) = \text{ret}(\text{data}(i,dyc)-1) + 1 - \text{sum}(\text{weights}(\text{data}(i,dyc),:)); \]
\[
\text{elseif} (\text{posterror} \&\& \text{sum}(\text{weights}(\text{data}(i,dyc),:)) < 1)
\]
\[
\text{ret}(\text{data}(i,dyc)-1) = \text{ret}(\text{data}(i,dyc)-1) + 1 - \text{sum}(\text{weights}(\text{data}(i,dyc),:)); 
\]
\[
\text{end} \\
\text{if}(\text{dispOn}) \\
\text{disp}([\text{num2str}((\text{ret}(\text{data}(i,dyc)-1)-1)*100,'\%0+6.3f'), '\% on day ', \text{num2str}(\text{data}(i,dyc)-1)])
\]
\[
\text{end} \\
\text{end}
\]
\[
\text{disp}('\text{Total return: '} \text{num2str}((\text{totalRet}-1)*100,'\%06.3f'))
\]
\[
\text{disp}('\text{Mean of returns: '} \text{num2str}((\text{mean}(\text{ret})-1)*100,'\%06.3f'))
\]
\[
\text{disp}('\text{Standard deviation for returns: '} \text{num2str}((\text{std}(\text{ret}))*100,'\%06.3f'))
\]

II. QUARTIC OPTIMIZATION ROUTINE:

\[
\text{function } [\text{WnV}] = \text{quarticsolver2}(\text{targetVal}, \text{fourMat}, \text{expVals}, \text{lower}, \text{upper})
\]
\[
\% \text{min } f(w) = (w(x)w)^Tc:(w(x)w) \text{ (minimal variance)}
\]
\[
\% \text{constraints:}
\]
\[
\% 1^T*w = 1 \text{ (weights add up to one)}
\]
\[
\% w^T*expVals*w \geq \text{targetVal} \text{ (target return is met)}
\]
\[
\text{n} = \text{length}(\text{expVals}(1,:));
\]
\[
\text{Aeq} = [\text{ones}(1,n)];
\]
\[
\text{beq} = [1];
\]
\[
\text{starts} = 0;
\]
\[
\text{if}(\text{sum}(\text{lower}) \neq 0)
\]
\[
\text{m} = \text{sum}(\text{upper})-\text{sum}(\text{lower}/\text{min}(\text{lower})) ;
\]
\[
\text{if}(m == 0)
\]
\[
\text{m} = .5;
\]
\[
\text{end} \\
\text{for i=1:n}
\]
\[
\text{if}(\text{upper}(i)==0)
\]
\[
\text{starts}(i)=-1/\text{m};
\]
\[
\text{else}
\]
\[
\text{starts}(i)=1/\text{m};
\]
\[
\text{end}
\]
\[
\text{end}
\]
\[
\text{else}
\]
\[
\text{starts} = \text{ones}(1,n)/n;
\]
\[
\text{end}
\]
\[
\text{w} = 0;
\]
\[
\text{fval} = 0;
\]
\[
\text{if}(\text{sum}(\text{upper}) \neq 0)
\]
\[
[\text{w},\text{fval},\text{exi}] = \text{fmincon (@(w) fourMatMult(w, fourMat), starts, [], [], \text{Aeq}, \text{beq}, \text{lower}, \text{upper}, @(w) \text{const}}(\text{w, expVals, targetVal}), \text{optimset('Algorithm', 'interior-point', 'Disp', 'off')));
\]
\[
\text{else}
\]
\[
[\text{w},\text{fval},\text{exi}] = \text{fmincon (@(w) fourMatMult(w, fourMat), starts, [], [], \text{Aeq},-0.5*\text{beq}, \text{lower}, \text{upper}, @(w) \text{const}}(\text{w, expVals, targetVal}), \text{optimset('Algorithm', 'interior-point', 'Disp', 'off')));
\]
\[
\text{end}
\]
\[
\%\text{disp variance}
\]
%w*expVals*w'
%disp second order variance
%fval
%disp exit code

if (exi ~= 1)
  %exi
end
%returns vector of weights with value, variance, and "exit flag" concatenated to end.
WnV = [w w*expVals*w' fval*2 exi];
end

% multiplies (w(x)w):C:(w(x)w)

function [ val ] = fourMatMult(w,fourMatr)
val = 0;
for i=1:length(fourMatr)
    for j=1:length(fourMatr)
        for k=1:length(fourMatr)
            for l=1:length(fourMatr)
                val = val + w(i)*w(j)*w(k)*w(l)*fourMatr(i,j,k,l);
            end
        end
    end
end
end

% multiplies w:C:w for minimization constraints

function [c,ceq] = const(w,mat,target)
    c = -(w*mat*w'-target);
ceq = 0;
end

III. QUADRATIC OPTIMIZATION ROUTINE (FOR MARKOWITZ)

function [WnV] = quadraticsolver(targetRet, covMat, expRets, shortsell)
% min_w w^T*covMat*w (minimal variance)
% constraints:
% 1^T*w = 1 (weights add up to one)
% expRets^T*w = targetRet (target return is met)
n = length(expRets);
Aineq = [expRets];
Aeq = [ones(1,n)];
bineq = [targetRet];
beq = [1];
if (shortsell == 1)
    [w,fval,exi] = quadprog(covMat,zeros(1,n),-Aineq,-bineq,Aeq,beq,-0.5*ones(1,n),ones(1,n),[],[],optimset('Algorithm','active-set','Display','off'));
else
    [w,fval,exi] = quadprog(covMat,zeros(1,n),-Aineq,-bineq,Aeq,beq,zeros(1,n),[],[],optimset('Algorithm','active-set','Display','off'));
end
IV. GARCH FORECASTING

function [ covMatFor ] = GARCHscript( ETFreturns, factor )

X = ETFreturns;
F = factor';

t = 0;
A = 0;
B = 0;

% regression
n = size(X);
for i=1:n(2)
    [b a] = polyfit(F,X(:,i),1);
    A(i) = b(2);
    B(i) = b(1);
    for j=1:n(1)
        err(j,i) = X(j,i) - (A(i)+B(i)*F(j));
    end
end

% GARCH forecast for F (30 min)
[junk Ffor] = garchfor(F,'GARCH','T',0,0,1,1,30);

% GARCH forecast for err's (30 min)
Errfor = 0;
for i=1:n(2)
    [junk Errtemp] = garchfor(err(:,i),'GARCH','T',0,0,1,1,30);
    for j=1:30
        Errfor(j,i) = Errtemp(j);
    end
end

% calculate variances
for i=1:n(2)
    for j=1:n(2)
        if (i==j)
            covMatFor(i,j) = Ffor(30)*B(i)*B(j)+Errfor(30,i);
        else
            covMatFor(i,j) = Ffor(30)*B(i)*B(j);
        end
    end
end
public class DataCleaner
{
    private HashMap<String, HashMap<Date, Return>> returns = new HashMap<String, HashMap<Date, Return>>()
    private ArrayList<Date> dates = new ArrayList<Date>();

    public void printReturns()
    {
        try
        {
        PrintStream out = new PrintStream(new FileOutputStream("ETFformatted2.csv"));
        for (String key : returns.keySet())
        {
            out.print(”,"+returns.get(key).get(dates.get(0)).getName());
        }
        out.println();
        for (int i = 0; i < dates.size(); i++)
        {
            if (!((dates.get(i).getMonth() == 10 && (dates.get(i).getDate() == 8 ||
                dates.get(i).getDate() == 25 || dates.get(i).getDate() == 28)))
            {
            out.print(dates.get(i)+",";
            for (String key : returns.keySet())
            {
                Return ret = returns.get(key).get(dates.get(i));
                if (ret != null)
                {

```
out.print(ret.getReturn()+",");
}
else if(dates.get(i).getHours()==9)
{
    int count = i;
    while(returns.get(key).get(dates.get(count)) == null)
    {
        count++;
    }
    out.print(returns.get(key).get(dates.get(count)).getReturn()+" ");
} //print extrapolated value
else
{
    out.print("BLANK,");
}
//System.out.println(ret.getDate() + " " + ret.getReturn());
out.println();
out.close();
}
}
catch(FileNotFoundException e)
{
e.printStackTrace();
}
}
private void printPrices()
{
    try
    {
        PrintStream out = new PrintStream(new FileOutputStream("ETFformatted2.csv"));
        for(String key : returns.keySet())
        {
            out.print(","+returns.get(key).get(dates.get(0)).getName());
        }
        out.println();
        for(int i = 0; i < dates.size(); i++)
        {
            out.print(dates.get(i)+",");
            for(String key : returns.keySet())
            {
                Return ret = returns.get(key).get(dates.get(i));
                if(ret != null)
                {
                    out.print(ret.getPrice()+" ");
                }
                else if(ret.getDate().getHours()==9)
                {
                    int count = i;
while (returns.get(key).get(dates.get(count)) == null) {
    count++;
}
out.print(returns.get(key).get(dates.get(count)).getPrice() + ",");

//print extrapolated value
else {
    out.print("BLANK,");
}

//System.out.println(ret.getDate() + " " + ret.getReturn());
out.println();
out.close();
}
catch (FileNotFoundException e) {
    e.printStackTrace();
}
}

private void importExcel() {
    String fileName = "ETFdata2.xls";
    File file = new File(fileName);
    Workbook workbook = null;
    try {
        workbook = Workbook.getWorkbook(file, new WorkbookSettings());
    } catch (Exception e) {
    }
    Sheet sheet = workbook.getSheet(0);
    int maxrowcount = 100000;
    for (int col = 0; col < 20; col++) {
        Return last4pm = null;
        Return last = null;
        int count = 1;
        while (count < maxrowcount) {
            String time = sheet.getCell(col*3, count).getContents();
            if (time.length() == 0) break;
            //System.out.println(time);
            int m = -1;
            try {
                m = Integer.parseInt(time.substring(0, time.indexOf("/") - 1);
            } catch (StringIndexOutOfBoundsException e) {
            }
        }
    }
}
System.out.println(time);
System.out.println(last.getDate() + " " + last.getName());
}
int d = Integer.parseInt(time.substring(time.indexOf("/"))+1,time.lastIndexOf("/"))); int y = Integer.parseInt(time.substring(time.lastIndexOf("/")+1,time.lastIndexOf("/)+3))+100;
System.out.println(m+" "+d+" "+y);
int h = Integer.parseInt(time.substring(time.indexOf(" ")+1,time.indexOf(":")));
int mi = Integer.parseInt(time.substring(time.indexOf(":")+1));
//System.out.println(m+" "+mi);
Date date = new Date(y,m,d);
date.setHours(h);
date.setMinutes(mi);
double returnFrom4 = 1;
double value = ((NumberCell)sheet.getCell(col*3+1,count)).getValue();
String ticker = sheet.getCell(col*3,0).getContents().substring(0,sheet.getCell(col*3,0).getContents().indexOf(" "));
if(last == null){
returns.put(ticker,new HashMap<Date,Return>());
}
if((h == 9 && mi == 30 && !(last.getDate().getMinutes() == 0 && last.getDate().getHours() == 16)) || ( h == 9 && mi > 29 && last.getDate().getMinutes() == 59 && last.getDate().getHours() == 15))
//if it's 9:30/next morning and there was no value for 4pm previous day
{
//System.out.println(last.getDate()+" "+last.getName());
Date ld = new Date(last.getDate().toString());
ld.setMinutes(ld.getMinutes()+1);
Return lr = new Return(ld,last.getPrice(),last.getReturn(),last.getName());
returns.get(ticker).put(ld,lr);
if(col == 0)
dates.add(ld);
last = lr;
last4pm = lr;
}
if(last4pm != null)
{
returnFrom4 = 1+(value-last4pm.getPrice())/last4pm.getPrice();
}
Return ret = new Return(date,value,returnFrom4,ticker);
if(h == 16 && mi == 0)//if 4pm
{
last4pm = ret;
}
if((h != 16 || mi == 0) && (h != 9 || mi > 29))//if not after 4pm and
not before 9:30am
{
if(last == null || (mi-last.getDate().getMinutes() == 1 || mi-
last.getDate().getMinutes() == -59) || (h == 9 && mi == 30))
}{
    returns.get(ticker).put(date, ret);
    if (col == 0)
        dates.add(date);
    last = ret;
}
else{
    //System.out.println(last.getDate()+" "+h+" "+mi);
    ArrayList<Return> rets = interpolate(last, ret);
    for (int i = 0; i < rets.size(); i++)
    {
        returns.get(ticker).put(rets.get(i).getDate(), rets.get(i));
        if (col == 0)
            dates.add(rets.get(i).getDate());
        //if(ret.getName().equals("SOXS") &&
        ret.getReturn().getMonth()==1 && ret.getReturn().getDay()==9)
            //System.out.println(rets.get(i)+" "+last+" "+ret);
    }
    returns.get(ticker).put(date, ret);
    if (col == 0)
        dates.add(date);
    last = ret;
}
}
class Return
{
    private Date date;
    private double price, returnFrom4;
    private String name;

    public Return(Date date, double price, double returnFrom4, String name)
    {
        this.date = date;
        this.price = price;
        this.returnFrom4 = returnFrom4;
        this.name = name;
    }

    public double getReturn()
    {
        return returnFrom4;
    }

    public double getPrice()
    {
        return price;
    }

    public String getName()
    {
        return name;
    }

    public Date getDate()
    {
        return date;
    }

    public String toString()
    {
        return (name + " " + date + " " + price + " " + returnFrom4);
    }
}