Statistical Arbitrage 237M – Section 1
Professor Ledoit – Anderson Graduate School of Management - UCLA

Problem Set #3 : Backtest

WARNING: This problem set is very long. You have one week to do it. It is due on Thursday, November 1st 2012 before class (10am). If you start two days before you'll never make it. You should start today.

Objective: The goal is to run a backtest simulation using some classical alphas. The backtest is intended to be realistic due to the inclusion of transaction costs and maximum trade limits.

1) Loading the Data: Download the official solution to Problem Set 2 from the class intranet. The Matlab database contains the following variables:

1. `allstocks   (1×n)`
2. `myday       (T×1)`
3. `price       (T×n)`
4. `tri         (T×n)`
5. `volume      (T×n)`
6. `mtbv        (T×n)`
7. `rec         (T×n)`
8. `isactivenow (T×n)`
9. `tcost       (T×n)`

Do not take the versions of these variables that you produced: everybody must start from the same data set. Let `T` denote the number of weekdays and `n` the number of stocks.

2) Risk Model: Every month, for the stocks in the universe where `isactivenow` is equal to one on the first day of the month, compute the shrinkage estimator of the covariance matrix using the past year of daily returns. By "returns", I mean specifically the arithmetic returns computed using DataStream's Total Return Index (variable named `tri`). As shrinkage target, take the identity matrix properly scaled. This is detailed in Lecture 4. Do not bother demeaning the returns. You will use this matrix for all the portfolio optimizations conducted that month. Store the optimal shrinkage intensities that you have estimated in a vector called `shrink` of dimension `(60×1)`.

3) Alphas: The alpha we will inject into the optimizer will be a weighted blend of alphas drawn from the four main categories of alphas:

a) SHORT-TERM CONTRARIAN (TECHNICAL): the short-term mean-reversion alpha with triangular decay detailed in Lecture 8. Store it in a Matlab variable called `alpharev` of dimension `(T×n)`. Assign to it a weight of 50%.
b) SHORT-TERM PROCYCLICAL (FUNDAMENTAL): the recommendations revision alpha detailed in Lecture 8. Store it in a Matlab variable called `alpharec` of dimension `(T×n)`. Assign to it a weight of 25%.
c) LONG-TERM CONTRARIAN (FUNDAMENTAL): the value alpha detailed in Part 1) of Lecture 5. Specifically, we will take the market-to-book ratio whose data is contained in the

matrix named `mtbv`. Store it in a Matlab variable called `alphaval` of dimension `(T×n)`. Assign to it a weight of 15%.

d) LONG-TERM PROCYCLICAL (TECHNICAL): the momentum alpha detailed in Part 2) of Lecture 5. We will use the last 12 months, kicking out the last month. Take what I called the “straight momentum”, which is the original and most basic version. Store it in a Matlab variable called `alphamom` of dimension `(T×n)`. Assign to it a weight of 10%.

You must cross-sectionally demean, standardize and windsorize every day each of these four alphas individually before blending them up with the specified weights, and you must also cross-sectionally demean, standardize and windsorize every day the blended alpha afterwards. Store the blended alpha in a Matlab variable called `alphablend` of dimension `(T×n)`.

<u>4) Optimizer:</u> You will use Matlab’s `quadprog.m` as detailed in Lecture 7. Choose the scalar parameter λ (Matlab variable name `lambda` of dimension `1×1`) that controls turnover and the scalar parameter μ (Matlab variable name `mu` of dimension `1×1`) that controls book size so that you trade approximately €15M (M stands for million) per day on average (excluding the first trading day of the period when you start from zero position), and that your average book size is approximately €50M long by €50M short. The backtest simulation runs for five years from the first business day of 1998 to the last business day of 2002. Since our databases start on January 1st, 1997, create a scalar `t0` that contains the index of the first day of trading. Make sure you use alphas, transaction costs and risk model computed from data available at the close of business on day *t-1* in order to compute the trade executed on day *t*. Store the trade expressed in € produced by the optimizer in a Matlab variable called `trade` of dimension `(T×n)`. *Important Note:* for the matrix `H` which is the first parameter passed to the `quadprog.m` function, use the corrected formula given in **Lecture07_v06.pdf** currently available on the intranet, which has a factor 2μ in front.

<u>5) Backtest Results:</u> Compute the stock positions in € at the end of day *t* as the stock positions in € at the end of day *t-1* multiplied by [one plus the stock returns on day *t*] plus the trade executed on day *t*. Store them in a Matlab variable called `back_weight` of dimension `(T×n)`. Compute your profit or loss on day *t* as the stock positions in € held at the close of business on day *t-1* multiplied by the stock returns on day *t* minus the transaction cost paid on the trade of day *t*. Store the *cumulative* P&L (profit and loss) in a Matlab variable called `pnl` of dimension `(T×n)`. Also compute your book size every day (long side plus short side) and store it in a Matlab variable called `booksize` of dimension `(T×1)`. Compute your trade size every day and store it in a Matlab variable called `tradesize` of dimension `(T×1)`.

<u>6) Performance Statistics:</u>

a) Compute your annualized Sharpe ratio assuming the risk-free rate is zero and store it in a Matlab variable called `sharpe` of dimension `(1×1)`.

b) Compute your longest drawdown, which is the longest time you spent without setting a new high-water mark and store it in a Matlab variable called `longest_dd` of dimension `(1×1)`. The high-water mark at any point in time is the highest cumulative P&L you have achieved up to that time.

c) Compute your deepest drawdown, which is the biggest loss you've taken from the previous high-water mark and store it in a Matlab variable called `deepest_dd` of dimension `(1×1)`.

Problem Set Output: You must e-mail Kyle Matoba a Matlab database containing the variables:

1.	`shrink`	`(60×1)`
2.	`alpharev`	`(T×n)`
3.	`alpharec`	`(T×n)`
4.	`alphaval`	`(T×n)`
5.	`alphamom`	`(T×n)`
6.	`alphablend`	`(T×n)`
7.	`lambda`	`(1×1)`
8.	`mu`	`(1×1)`
9.	`t0`	`(1×1)`
10.	`trade`	`(T×n)`
11.	`back_weight`	`(T×n)`
12.	`pnl`	`(T×n)`
13.	`booksize`	`(T×n)`
14.	`tradesize`	`(T×n)`
15.	`sharpe`	`(1×1)`
16.	`longest_dd`	`(1×1)`
17.	`deepest_dd`	`(1×1)`

Grading: You will be judged by how complete and correct these variables are.

Very Important: Due to the limited time that the T.A. can devote to this class, every student is allowed at most one e-mail question to the T.A. – so use it wisely! In addition, the T.A. will hold one office hour which will be devoted to this problem set.

<u>7) Optional Bonus Question:</u> For extra points, find ways to improve this strategy in the sense of more cumulated profit, higher Sharpe ratio, shorter longest drawdown, smaller maximum drawdown, a combination of the above, or (ideally) all of the above. Explain what you have modified and save the results in the same format as the original backtest.