COMP 510 – Computational Finance
Prof. Steven Skiena
Fall 2008

Homework 2 – Trading Strategies
Due Thursday, November 27, 2008

The development of algorithmic trading strategies for equities is a particularly exciting part of Computational Finance. In this assignment, you are asked to design, implement and evaluate trading strategies on real world stock price data. In particular, we have prepared a data set containing daily historical stock price quotes from 490 S&P 500 companies over the ten year period from October 1, 1998 to October 1, 2008.

You are required to do the following tasks:

- Implement at least three simple baseline trading strategies including:
  - *The Mattress Strategy*: hold all cash at all times, i.e. buy no stock
  - *Buy and Hold:* buy all stocks with equal weight at the beginning, and hold.
  - *Diversification with Rebalancing:* at regular intervals, sell overweighted stocks and use the proceeds to buy underweighted ones so you have equal dollar weights of each.
- Propose and implement at least three other more sophisticated strategies. These may based such ideas as (1) pairs trading, (2) momentum trading, i.e. pick stocks which have been successful recently, (3) risk-return trading, i.e. pick stocks with the highest volatility/risk, (4) technical analysis, etc.
- Evaluate each of your strategies using the simulator described below. Then write a five page report describing your strategies, how they performed, and any interesting results you find.

Our trading simulator/evaluator accepts stock trading orders with following format:

```
buy 100 YHOO 20010908 open
```

which means we want to buy 100 shares of the Yahoo stock (with symbol YHOO) at the open price on the date 09/08/2001. The Evaluator takes a file of trading orders as input, executes these orders and outputs an evaluation of the given trading strategy. Specifically, the evaluator will create a log file that indicates the execution errors and the final portfolio at the end of the input trading order sequence. It will also create a .csv file that traces through the orders month-wise and records the date, number of trades, cash, stock value, short liabilities, monthly portfolio, monthly return since last period, monthly return since beginning. This enables us to see how successful a strategy really is.

Your mission is to read the price data and emit buy and sell orders at any time without peaking into the future, producing output files (one per strategy) in a format suitable for the evaluator. The generated sequence of orders should be ordered with dates increasing in order of time, as a real stream of orders. Other rules:

- You start with a total of $1,000,000 at beginning of the simulation. You are not allowed to go negative.
- You are allowed to short sell without no limit. This permits you to cheat in certain ways, but do not to abuse the power. You must clear your accounts before the end of the simulation.
- Cash in an account earns interest at a 5% annual rate.

To serve as examples, there are two random strategies built into the generator. One is randomized buy/sell and the other is randomized buy/hold. The former means we generate a random sequence of buy and sell orders while the latter means we just randomly buy stocks without any selling until the final date in the run.

Data Set and Evaluation Programs

The full time series data has been divided into four quadrants by halving the company names and halving the date range. This provides you the opportunity to train your dataset on one quadrant and evaluate it on the others, with the alternate time and name quadrant probably being the most informative. Eventually, the TA may test your strategy generator on all sets.

The stock dataset 1 is contained in *dataset1.csv.* There is also file *category.csv* which contains information fields about each company, such as name, ticker symbol, industral sector, etc.

We have provided a sample input and output file. Confirm the simulator behaves similarly when you run it.

The source code for both the evaluator and an example random trade generator is contained in the directory *Source.* There is also a directory called *Excutable* which contains the excutable files. Instructions on usage of both programs are contained in the file *readme.txt.*

You may wish to compile these programs and get your own excutable file. These programs were written in C++ using VS9. Make sure that the *stockdata.csv* (under the directory Excutable) is correctly placed prior to running the program.

Rules of the Game

1. *The goal of this assignment is not to find the most profitable strategy.* The best grades will go to the most interesting / honest / rigorously evaluated trading strategies/report, not the one with the highest payoff.
2. *The goal of this assignment is not to find the most profitable strategy.* It would be easy but pointless to cheat by (1) looking ahead because the results for the next ten years already sit in the file, or only slightly less dishonestly (2) tune your strategy over several runs for the best results on the given dataset.
3. *The goal of this assignment is not to find the most profitable strategy.* We will take 5 points off the paper of whichever student reports the most profitable strategy, on general principles.

4. The simulator is student-built software, and likely very brittle. Start early to implement and evaluate at least the baseline strategies in the first week to shake down simulator problems.

5. In your report, clearly explain how your strategies work, how they performed (using graphs or plots to demonstrate). Also, try to explain why your strategies produced the following results, as grading will partially depend uponyour understanding/reasoning.

6. All data files are available from the course webpage *http://www.cse.ust/hk/∼skiena/510*.

7. Please submit your assignment through CASS at *https://course.cse.ust.hk/cass/submit.html*. You shall find the submit guides there.

8. You may write your programs in whatever language you wish. Submit your program code along with the report and any spreadsheets you happen to build.

9. Please submit only one zipped file, with the format: yourname(stu_id).

10. I encourage open-ended exploration. Experiment and have fun!