CSE599s, Spring 2014, Online Learning Due: 05/27/2014

Programming Assignment 2

Instructors: Ofer Dekel, Brendan McMahan *TA: De (Dennis) Meng*

Submission: Programming assignments involve both mathematical derivation and coding. Your submission should include a write-up that describes your derivation and explains your code for each sub-problem. Your code should print out the final results as required by each problem. Please submit your write-up (pdf file) and source code files in a single compressed package named "Coding2_YourFirstName_YourLastName" to the dropbox
`https://catalyst.uw.edu/collectit/dropbox/summary/demeng/31445`.
If you have any questions, feel free to contact the TA or discuss on the discussion board
`https://catalyst.uw.edu/gopost/board/demeng/36570/`.

In this programming exercise, we will implement and experiment with *experts* algorithms using stock market data.

Data The file `prices.txt` contains high frequency stock prices in a tab-separated matrix format. Columns 1 and 2 indicate date and time, while each of the remaining 2770 columns corresponds to a stock in the NASDAQ stock market. To take care of splits and penny stocks, we eliminate those stocks that either dipped below \$1 or changed by more than 40% in any 5 minute period. The header row specifies the column names, and each of the other rows contains the stock prices at a given time. Below, we use $v_{t,i}$ to denote the price (value) of stock i at time t.

Setting Assume that we do not pay any transaction costs, and that our trading strategy is to reinvest our money once every five minutes. In other words, every five minutes we liquidate all of our assets for cash and decide how to reinvest our money for the next five-minute period.

Background We can make two types of investments:

1. Taking a *long* position in a stock means that we buy shares of that stock at the beginning of the five-minute trading period and sell them after five minutes. Therefore, if we invest α dollars in a long position in stock i on round t, our profit (i.e. change of the market value, possibly negative) after five minutes is $\alpha\big(\frac{v_{t,i}}{v_{t-1,i}} - 1\big)$ dollars. If the price of the stock goes up during the five minute trading period, we make a positive profit.

2. Taking a *short* position in a stock means that we borrow shares of that stock from the market and sell them at the beginning of the trading period. After five minutes, we buy back the same number of stocks and return them to the market. Therefore, if we take a short position in the value of α dollars in stock i on round t, our profit after five minutes is $\alpha\big(\frac{v_{t-1,i}}{v_{t,i}} - 1\big)$ dollars. If the price of the stock goes down during the five minute trading period, we make a positive profit.

1 Implement

1. Implement the EG (Exponentiated Gradient) algorithm for the experts problem. Recall that EG draws I_t from the distribution p_t, which is obtained by solving FTRL with the regularizer $R(p) = \frac{1}{\eta}\sum_{i=1}^{d} p_i \log(p_i) + I_{\Delta_d}(p)$. However, you should use the convenient closed form for p_t, which was derived in class. Your implementation should allow the user to specify an arbitrary learning rate and an arbitrary set of experts.

2. Implement the GD (Gradient Descent) algorithm for the experts problem. GD draws I_t from the distribution p_t, which is obtained by solving FTRL with the regularizer $R(p) = \frac{1}{2\eta}\|p\|_2^2 + I_{\Delta_d}(p)$. The solution to FTRL in this case is almost a closed form, and only requires a Euclidean projection onto Δ_d (see theoretical homework exercise 2). A simple algorithm that performs this projection is summarized in `http://faculty.ucmerced.edu/mcarreira-perpinan/papers/SimplexProj.pdf`, with Matlab code provided.

3. For any matrix of losses corresponding to a set of experts, you will also need to compute the loss of the best expert in order to compute regret. Another useful comparison point is to compute the loss of the average expert (the loss you would have obtained by always playing the uniform distribution).

   For both algorithms, you should compute two values for each run of the algorithm:

   - Compute the actual cumulative loss where on each round t an expert i_t is sampled according to the probability distribution p_t from the algorithm; your algorithm should return the sampled cumulative loss, $\sum_{t=1}^{T} \ell_t(i_t)$.
   - Directly compute the expected cumulative loss (the average sampled cumulative loss if you repeated the above many times). On each round, the expected loss can be computed as

   $$\sum_{i=1}^{d} p_{t,i}\ell_t(i) = p_t \cdot \ell_t.$$

   Comment: If you were investing a large amount of money on each round, say D dollars, then you could just buy Dp_i dollars worth of every stock i, and achieve exactly the expected cumulative loss. There are at least two problems with this approach: if there are transaction costs (which we aren't modeling), you will have many more transactions if you buy some of every stock, instead of just sampling a single stock to buy. Second, if D is small, you could easily end up with $Dp_i < \$0.01$, and it is likely impossible to buy stocks in such small quantities.

2 Experiment

1. For each stock in the data, define an expert that *always* recommends taking a *long* position in that stock. Namely, define 2770 experts, where expert i always recommends buying stock i. Define the loss of expert i on round t as

   $$\ell_{t,i} = \begin{cases} 0 & \text{if } \frac{v_{t,i}}{v_{t-1,i}} \geq 1.0001 \\ 1 & \text{otherwise.} \end{cases}$$

   In words, an expert gets a zero loss only if his recommended investment goes up by at least 0.01%. The goal is to maximize the number of trading periods (online rounds) in which our investment increases by at least 0.01%. Use EG (and GD) and follow the advice of a single expert on each round (for the sampled loss).

   Tune the learning rate η using the first 400 trading periods (think of this as historical data currently available).[1] You should choose the η based on the exact expected cumulative losses computed on the training data, not the sampled values (why?). We recommend using a multiplicative grid of possible η's, for example, finding the best value from the set $\{\eta_0 2^i \mid i \in \mathbb{N}\}$, with η_0 an initial guess of the learning rate (perhaps based on the theory); for a finer grained search you could replace 2 with 1.5 or 1.2, but this is not necessary. Note that the theory recommends an η of the form $\frac{\gamma}{\sqrt{T}}$; rather than optimizing η we will instead optimize γ. If η_0 is the value that works best on the training data, this is equivalent to choosing $\eta = \eta_0 \frac{\sqrt{399}}{\sqrt{1471}}$ on the test data.

[1] This means you should have $T = 399$ for the training data, since you are looking at the gains between prices. Similarly, you should have 1472 prices for each stock in the test data, leading to $T = 1471$ rounds for EG.

Once you have chosen a value of η, fix this value and run on the test period. Plot the exact cumulative expected loss as a function of t. Plot 10 additional lines showing the sampled cumulative loss for 10 runs (each using fresh randomization). Make one plot for EG, and one for GD. Repeat this, making two more plots (one for EG, one for GD), this time with Regret on the y-axis. Note that Regret at time t should be computed with respect to the best expert on rounds $1, \ldots, t$ (which may not be the best expert at the final time T).

You will also report the results after the final round ($T = 1471$) in a table, see instructions at the end of this section. For the remaining experiments, you will only report results in this table, but we encourage you to make the plots and examine them.

2. **Flat Market Experts** Add 2770 additional experts, one per stock: each of these new experts believes that the price of his stock simply oscillates around a constant value. The expert keeps track of a 5-round moving average of the stock price, $\frac{1}{5}\sum_{s=t-5}^{t-1} v_{s,i}$: if the most recent price is below the moving average, the expert recommends a long position in the stock; if the price is above the moving average, the expert recommends a short position in the stock. The loss of expert i in this setting can be defined as

$$\ell_{t,i} = \begin{cases} 0 & \text{if expert advised long position and } \frac{v_{t,i}}{v_{t-1,i}} \geq 1.0001 \\ 0 & \text{if expert advised short position and } \frac{v_{t-1,i}}{v_{t,i}} \geq 1.0001 \\ 1 & \text{otherwise} \end{cases}$$

Fill in the corresponding rows in the table shown at the end of the exercise for both algorithms using the 5540 experts defined so far.

3. **Statistical Arbitrage Experts** Add $\binom{2770}{2} = 3835065$ additional experts, one per each pair of stocks. The expert that corresponds to the pair (i, j) believes that stocks i and j follow the same general trend, but are not identical. Therefore, if the price of i increases more than the price of j, then j will soon catch up with similar gains; and if the price of i drops more than the price of j, then j will soon correct with a similar drop. This behavior is often true for correlated stocks (such as Coke and Pepsi) but it is not true in general - instead of figuring out exactly which stock pairs to follow, we just define an expert for each pair and let EG/GD compete with the best pair in hindsight.

   The expert that corresponds to the pair (i, j) makes one of two possible recommendations: if he thinks that the stock i is currently overpriced (compared to j), he recommends that we use *half* of our money to short i and half of our money to long j. Alternatively, if he thinks that the stock i is currently underpriced (compared to j), he recommends that we use half of our money to long i and half of our money to short j. Concretely, he compares $\frac{v_{t-1,i}}{v_{t-5,i}}$ (how much i gains in past 5 rounds) to $\frac{v_{t-1,j}}{v_{t-5,j}}$ (how much j gains in past 5 rounds) and recommends taking a long position in the lower one and a short position in the higher one. Define the loss for each of these experts. Again, fill in the corresponding rows in the table below.

Reporting Your Results We implemented 2 algorithms, refined the set of experts 3 times, which gives 6 experiments, corresponding to the 6 rows in the table below. You should fill in the values in each of the last 5 columns.

The experts column refers to the set of experts: the "long" experts are those used in part 2.1. The "flat" expert rows *add* the flat market experts defined in part 2.2, and the "arbitrage" rows corresponding to part 2.3 (where we have all three types of experts). Note that you should optimize η on the training set separately for each row, following the instructions in 2.1.

The "eta" column gives the learning rate you chose based on the training data; the remaining columns refer to results on the test set. The "loss" column gives the cumulative expected loss of the algorithm; the

"best expert" is the loss of the best expert. The "avg expert" gives the of the average expert, that is, the loss you would have obtained by playing the uniform distribution on every round.

alg	experts	eta	loss	best expert	avg expert	regret
EG	long					
EG	flat					
EG	arbitrage					
GD	long					
GD	flat					
GD	arbitrage					