Lecture 22: Competitive Analysis for Finance

Steven Skiena

Department of Computer Science
State University of New York
Stony Brook, NY 11794–4400

http://www.cs.sunysb.edu/∼skiena

Competitive Ratio

We say an online algorithm ALG is c-*competitive* if there is a constant α such that for all finite input sequences I,

$$ALG(I) \leq c \cdot OPT(I) + \alpha$$

Note that the additive constant α is a fixed cost that becomes unimportant as the size of the problem increases.
We do not particularly care about the run-time efficiency of ALG (except maybe that it is polynomial), but we do care about its competitive ratio c.

One-Way Trading

A generalization of the price searching problem is to sell my entire assets over the trading period, but to remove the constraint that I must sell it all at once.

Suppose I am trying to liquidate my position in a stock. I may be able to better optimize my expected performance by using this freedom.

This is particularly true in real markets, as my sales serve to depress prices by increasing supply.

For this problem, it turns out that there is no difference between what competitive ratio is achievable with and without randomization.

What is a reasonable strategy?

Threat-Based Strategies

Suppose we know that a competitive ratio of c can be obtained.

If the current price is high and I don't sell, my adversary can drop it to m and keep it there the rest of the period.

But if I do buy, the adversary can jack the price to M at least momentarily, and I will be in trouble if I have already sold everything.

The *threat-based strategy* sells only when the price hits a new maximum. It sells just enough to ensure that we achieve a competitive ratio of c if the price drops to m for the rest of the game.

Randomized Strategies

Analysis is needed to determine the optimal c value and also how much to buy in response to price changes.

Clearly, we can achieve $c = O(\lg \phi)$, since we can use the randomized strategy and sell all at once.

We can simulate the randomized strategy deterministically by putting $1/k$th of our money on each value of i.

The optimal threat-based strategy for one-way trading achieves a competitive ratio of $1/\ln 2 \approx 1.44$ times better the search bound of $EXPO$.

Assessing the Model

How useful are our competitive algorithms for price searching and one-way trading?

Assumptions of upper and lower bounds on possible price movements seem suspicious, although we can make safe over/under-estimates of possible movements over short periods of time using historical data.

Guaranteeing you do, say half, as well as optimal doesn't look so good when the difference between the best investor and an index fund is often only a few percentage points.

That said, the randomized and threat-based heuristics seem to suggest reasonable approaches for one-way trading.

Online Portfolio Selection

Suppose we can invest in a market with s types of assets, including cash.

How should partition our money among the assets, and how should we adjust our portfolio to changes in asset prices?

The *buy and hold* strategy (BAH) strategy does not attempt to modify the portfolio for long periods of time.

Buy and hold results in very low transaction costs, and is historically better for individual investors to do than *market-timing* strategies which switch among investments seeking the best return.

Rebalancing Algorithms

Market timing can yield much better returns *if* you do it right. Consider a stock that alternates returns of d and $1/d$. Buy and hold returns at most d over any period of length n, while the optimal market timing would yield a return of $d^{n/2}$.

A *constant rebalancing algorithm* always puts $1/s$ of our current wealth in each of the s securities.

Such a diversified strategy enables us to pickup exponential growth over the previous return sequence if it starts positively.

Implementing a constant rebalancing strategy requires daily trades, but provides a way to capitalize on boom periods.

Two-Way Trading

Two-way trading is a special case of online portfolio selection where you have only cash and one other security you can hold.

It differs from one-way trading in that we can shift back and forth between the two assets.

The optimal *offline* strategy is clear: put all your assets into the security on any day it offers positive returns. Otherwise, put everything into cash.

No Guaranteed Free Lunch

A trading strategy is said to be *money making* if it returns positive profit on every market sequence for which the optimal offline algorithm makes a profit.

The general adversary can ensure that *no* money making strategy exists.

If you are not initially invested in the non-cash asset, the next period will be the only one offering positive returns. If you are initially invested in the non-cash asset, offer such a negative return as to essentially wipe it out, then have a small enough positive return that you cannot recoup what you lost.

The (n, ϕ) Adversary

Provably money making strategies are only possible against weaker adversaries.

An (n, ϕ) trading sequence is a $n-1$ day sequence of returns for which the optimal offline strategy generates a return of at least ϕ.

$$\phi = \prod_{i=1}^{n-1} \max\{1, r_i\}$$

An (n, ϕ) adversary is constrained to produce (n, ϕ) trading sequences.

We assume that you are given n and ϕ in advance to help you plan your trading strategy.

The Basis Case

Can you devise a provably money making strategy against an (n, ϕ) adversary?

How about when $n = 2$?

For $n = 2$, we have only one trading period. Since this must produce a profit ϕ, we should be fully invested in the non-cash asset.

How about when $n = 3$?

Making Money from the Adversary

For $n = 3$, we must look ahead to the case of $n = 2$. If we initially bet only on cash, the adversary will make that the only period of positive return.

If we are initially out of cash, the adversary can wipe us out now and show a ϕ in the next period.

We must bet something but not everything on the non-cash asset in the first round. If it shows positive return, we can quit the game with our holdings. If it shows negative return, we still have money and know there must be a positive return of at least ϕ in the remaining time.

Through such reasoning, for large n we can work backwards from $n - 1$ to figure out the best first move to make.

The Money Making Strategy

If $n = 2$, invest in the non-cash asset for return $R_2(\phi)$.
Otherwise, invest the fraction b of your wealth in the non-cash asset, where

$$b = argmax_{b=0}^{1} \inf_{x \leq \phi} \{(bx + (1-b))R_{n-1}(\phi_{n-1})\}$$

$argmax$ returns the b which maximizes the value, as opposed to $\max$ which returns the value.
Once you pick a b, your adversary will pick a return x such as to minimize your wealth.
Your wealth after this event is your initial wealth times the returns on the cash and non-cash portions, i.e. $(bx + (1-b))$.
After the return x is revealed to you, you can figure out the guaranteed return for the remaining period:

$$\phi_{n-j-1} = \min\{\phi_{n-j}, \phi_{n-j}/r_{j+1}\}$$

The best value of b can be determined by dynamic programming.
Although this strategy is provably money-making, it can yield poor returns for large n,

$$R_n(\phi) \le \frac{1}{1 - (1 - 1/\phi)^{n-1}}$$

For large n, it can initially only afford to put a small amount of money into non-cash assets.
Since the optimal offline return is ϕ, we get a not-inspiring competitive ratio of $\ge \max\{n-1, \phi\}$.

The Fixed Fluctuation Model

In the fixed fluctuation model, all returns are either α or $1/\alpha$. Such a model is consistent with our random walk model, although we picture the return sequences as being generated by a hostile rather than random adversary.

It can also be thought of *time scaling* model, where we consider each return of α or $1/\alpha$ as one step, regardless of how long it took to take that step.

An (α, n, k)-adversary generates length-n binary sequences on $(\alpha, 1/\alpha)$ where exactly k individual returns are profitable. Thus the optimal offline return is α^k.

Strategy by Dynamic Programming

Let FMM denote the optimal money making algorithm against this adversary and $R_\alpha(n,k)$ be its return. Then:

$$R(n,k) = \max_{b=0}^{1} \quad \min\{(b/\alpha + (1-b))R(n-1,k), \\ (\alpha b + (1-b))R(n-1,k-1)\}$$

for boundary conditions $R(n,0) = 1$ and $R(n,n) = \alpha^n$.

Note the similarities to Binomial Trees!

Results Against Fixed Fluctuation Adversaries

It can be proven that FFM is always better than optimal offline buy and hold.

For the constant rebalancing strategy $(b, 1-b)$, the optimal rebalancing constant

$$b = \frac{(n/k)(\alpha+1)-1}{\alpha-1}$$

This constant rebalancing strategy is also better than buy and hold – however we assume no transaction costs.