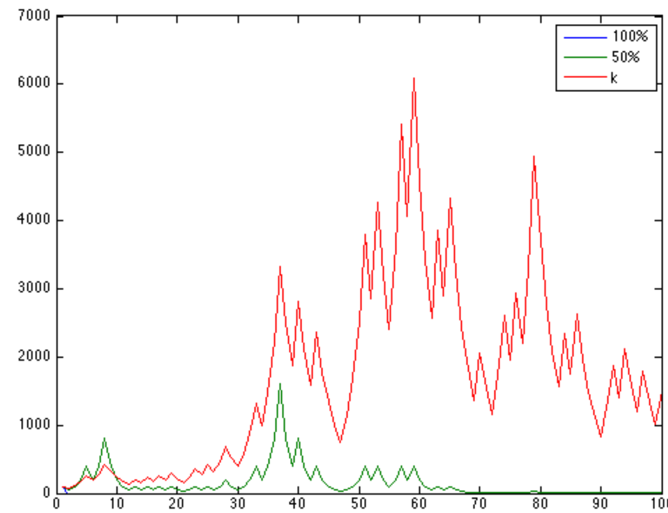


Econophysics

Kelly Criterion

Cristian Heredia

Physics 250 W2013
J. Rundle
March 20, 2013

Table of Contents

Cristian Heredia

Physics 250
J. Rundle
March 20, 2013

Econophysics

Kelly Criterion

The purpose of this project is to lay out the groundwork for a general money management system using the Kelly Criterion. Ideally, the variables used in the Kelly formula would come from back testing portfolio returns.

Background

$$f^* = \frac{bp - q}{b} = \frac{p(b+1) - 1}{b},$$

Kelly Formula

John Kelly, a scientist for Bell Laboratories, developed a formula to help with long distance telephone signal noise issues. Kelly published his findings, "A New Interpretation Of Information Rate" (1956); in it, Kelly optimized the equation Cn

$$C_N = (1 + fb)^{pN}(1 - fa)^{qN}.$$

In the above equation—for gambling purposes—Cn represents capital; f, fraction of capital to wager; 1+fb, the amount of possible winnings in wager; 1-fa, the amount possibly lost in wager; N, trials; pN, successes; qN, failures. By taking the log of Cn, and finding the maximum of the function by taking the first derivate, Kelly arrived at Kelly Formula, f*.

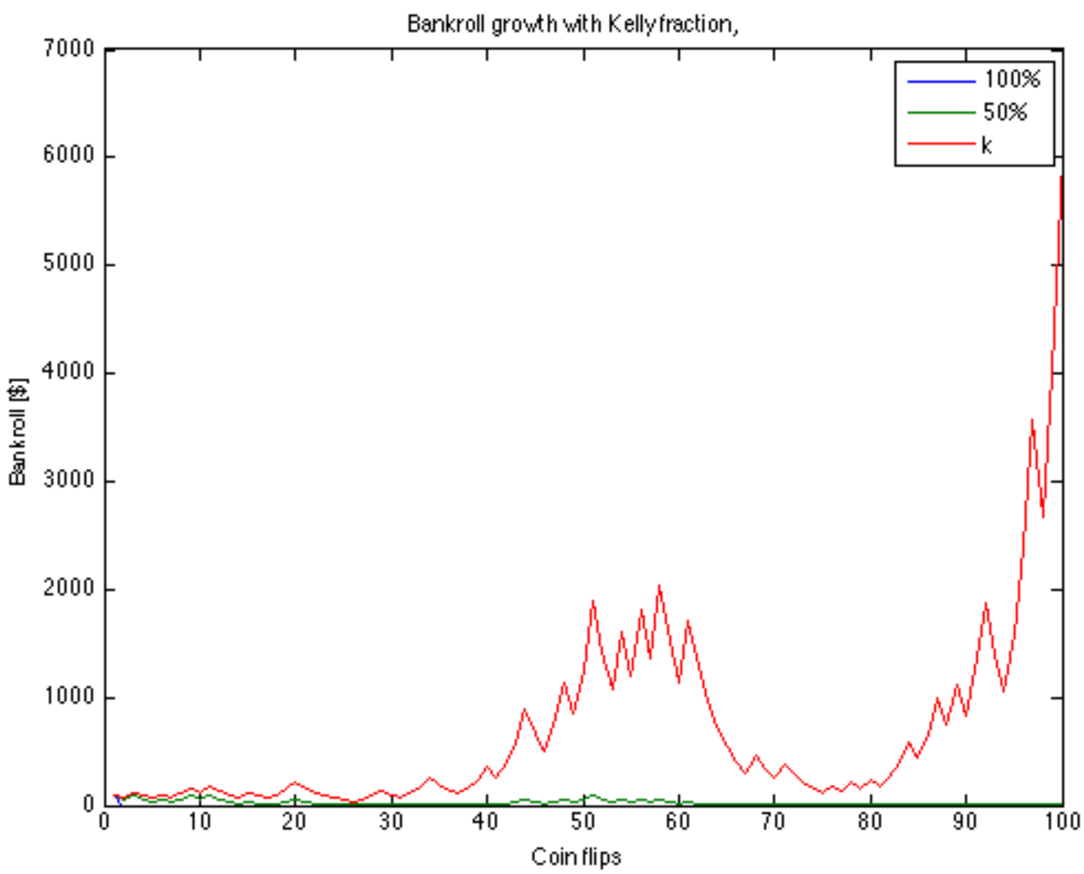
- p = winning probability
- b = win/loss ratio
- q = losing probability

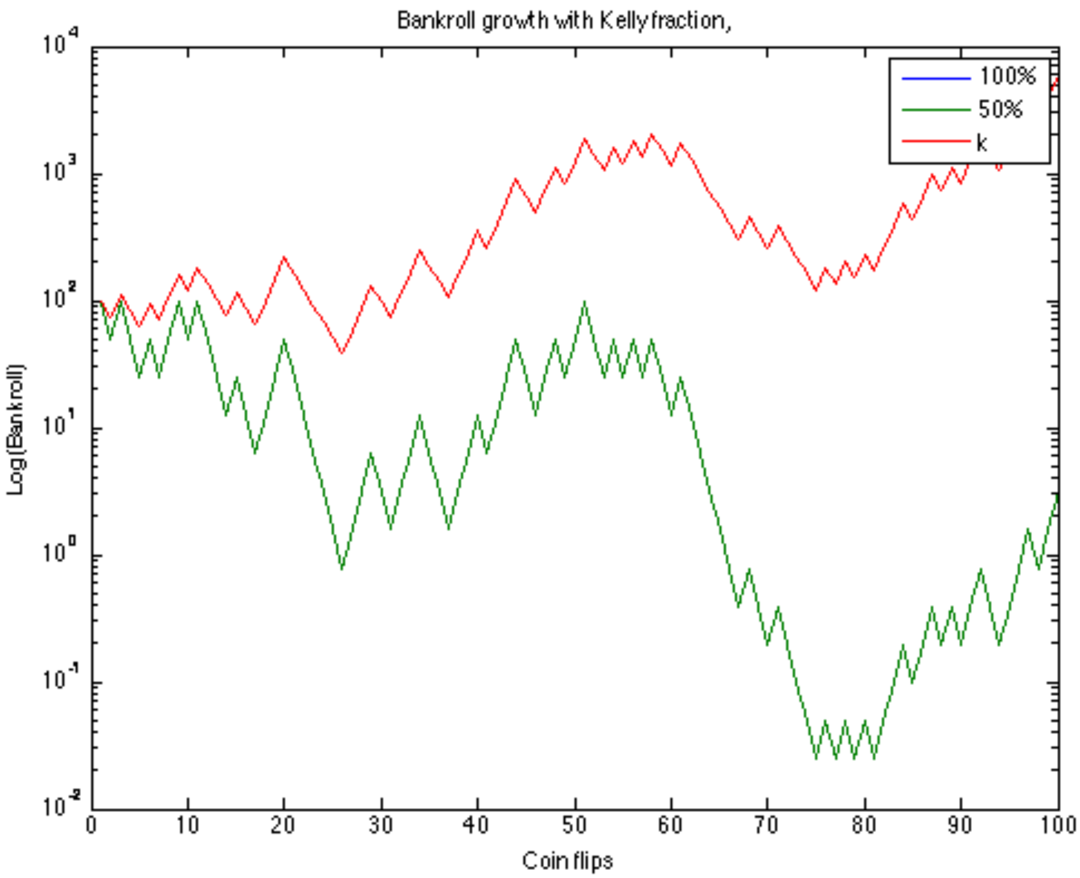
Coin Flip Example

As a proof of concept let us examine the scenario where a coin is tossed. If the coin shows heads we win the wager and double the initial bet; for tails we lose the wager.

Three different betting strategies are analyzed: betting all of bankroll for every toss, betting half of bankroll for every toss, and finally betting the Kelly fraction, f. The following charts, graphed in Matlab, show that betting the Kelly fraction will yield the greatest bankroll.

- p = 0.5
- b = 2
- q = 0.5





Betting all of your money at once will not keep you in the game for long, because all it takes is one loss to be out of the game. Betting half of your bankroll will insure you stay in the game forever, assuming infinitesimal bets are allowed. However, betting the Kelly fraction yields the best returns over many iterations. For the Coin Flips figures f = 0.25. Matlab code is found in Appendix 1.

Portfolio Management with Kelly Criterion

The difficulty in using the Kelly Criterion comes with choosing the appropriate variables. Back testing previous trades is necessary for accurate and portfolio pertinent variables.

I developed a Matlab code that finds the dynamic Kelly Criterion variables for any stock as it evolves in time. The program takes the average number of positive returns and divides them by the number of total returns; this value is saved as, p, the probability of winning. The win/loss ratio is found taking the average positive returns and dividing by the absolute value of the average negative returns. The program solves these values for every value t, where t is a trading day. See Appendix 2.

- p: Found taking the number of trades that yielded a positive return divided by the total number of trades.
- b: Found by dividing the average gain of the positive trades by the average loss of the negative trades.
- q = 1-p

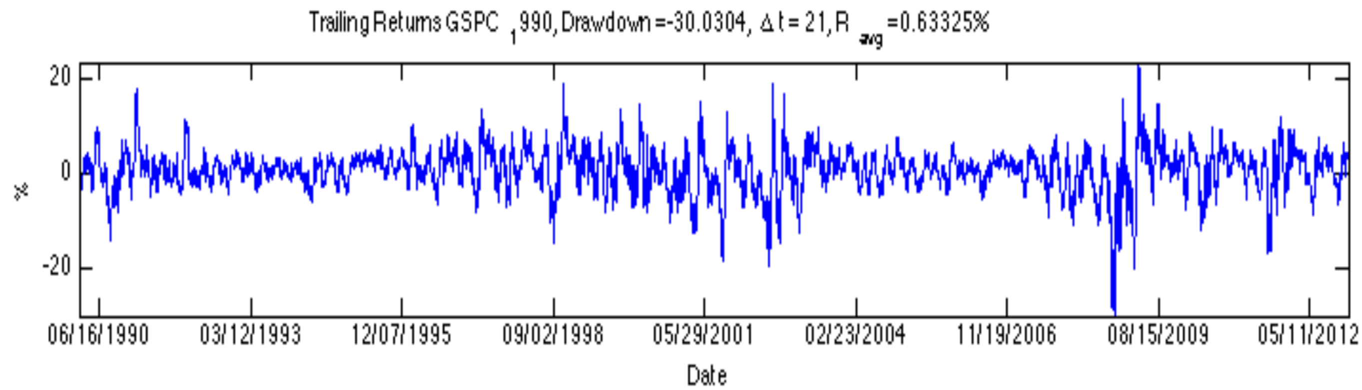
Limiting bankroll

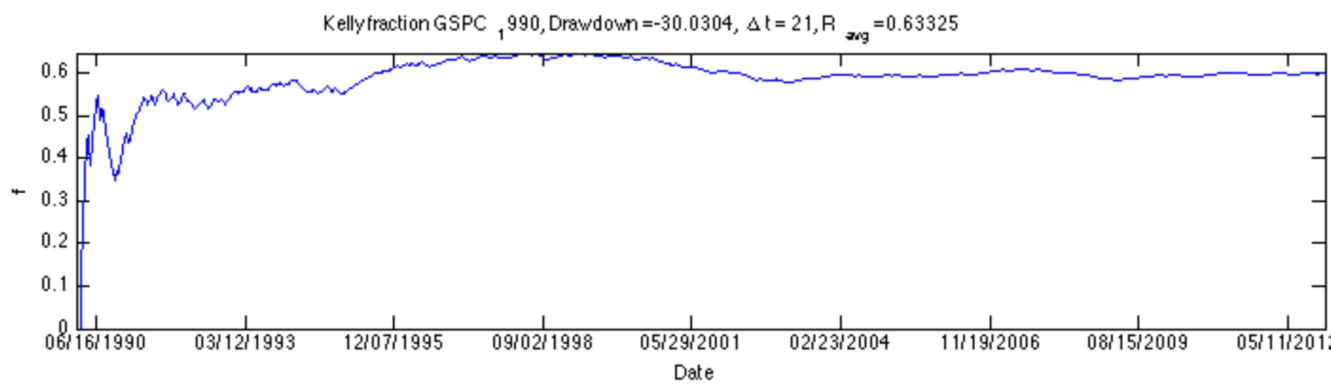
From the simulation it was found that starting with an initial small bankroll, ~$1000, leads to a depleted purse that doesn't recover despite betting the Kelly factor. This could be because betting the Kelly fraction assumes wins over many iterations, i.e., winnings accumulate over an asymptotic process.

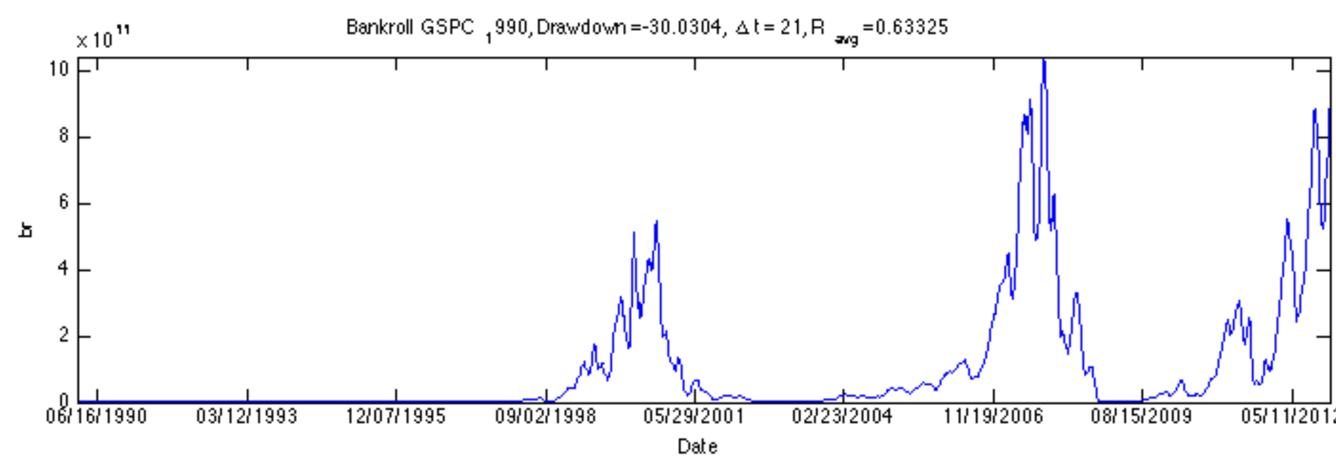
Trading Fees

I also built in the notion of trading fees: For every iteration, t, a transaction fee or trading fee is assessed to the running bankroll.

Large fees, on the order of $20, also quickly deplete the bankroll early in the trading. This bankroll crash looks very much like the dampening exhibited when starting with a limited bankroll.


Trailing Returns GSPC 1990, Drawdown =-30.0304, Δ t = 21, R avg =0.63325%
%
Date
06/16/1990 03/12/1993 12/07/1995 09/02/1998 05/29/2001 02/23/2004 11/19/2006 08/15/2009 05/11/2012


Kelly fraction GSPC 1990, Drawdown =-30.0304, Δ t = 21, R avg =0.63325
f
Date
06/16/1990 03/12/1993 12/07/1995 09/02/1998 05/29/2001 02/23/2004 11/19/2006 08/15/2009 05/11/2012


Bankroll GSPC 1990, Drawdown =-30.0304, Δ t = 21, R avg =0.63325
x 10^11
br
Date
06/16/1990 03/12/1993 12/07/1995 09/02/1998 05/29/2001 02/23/2004 11/19/2006 08/15/2009 05/11/2012

Conclusion

The Kelly Criterion is a proven system for making systematic bets. As seen from the simulations the formula could be used to diversify a money management portfolio. Its strength lies (at least the way it was implemented in this code) in the absence of human opinion, by that I mean variables were determined by back testing stocks.

When entered in the simulation, the DIA stock is not a good one to follow with the Kelly Criterion bet. The DIA bankroll drops off early in the life of the stock and never recovers. This needs to be further investigated.

Finally, the program could be further improved by using another metric for winning probability and win/loss ratio.

Appendix 1

```
%Let's solve earnings for a coin flip bet: heads you double your bet; tails
%you lose.
clear, clc
%First let's generate a matrix with random heads and tails, 1's and -1's.
flips = 100;
r = rand(flips,1);
r = r<0.5; % This generates a binary array
r = 3*r - 1; %This simple procedure produces a matrix with +2, and -1
m = mean(r) %calculate the mean of flips, should be 0.5

%Define Kelly Criterion
 b = 2; %net odds you could win on wager
 p = .5; %is the probability of winning
 q = 1-p; %probability of losing
 k = (b*p-q)/b;

%Let's simulate different betting strategies

%(1) Assume no strategy, bet 100% assets
a_purse(1) = 100;  %initial bank roll
for n = 2:flips %Steps through all the coin tosses
a_purse(n) = a_purse(n-1) + a_purse(n-1)*r(n);
end
a_purse(flips);


%(2) bet 50% assets
b_purse(1) = a_purse(1);  %initial bank roll
for n = 2:flips %Steps through all the coin tosses
b_purse(n) = b_purse(n-1) + .5*b_purse(n-1)*r(n);
```

```
end
b_purse(flips);


%(3) bet k assets
c_purse(1) = a_purse(1);  %initial bank roll
for n = 2:flips %Steps through all the coin tosses
c_purse(n) = c_purse(n-1) + k*c_purse(n-1)*r(n);
end
c_purse(flips);

figure(1)
tosses = (1:1:flips); %generates number of tosses
plot(tosses,a_purse,tosses,b_purse,tosses,c_purse)
title('Bankroll growth with Kelly fraction,')
ylabel('Bankroll [$]')
xlabel('Coin flips')
legend('100%','50%','k');

figure(2)
tosses = (1:1:flips); %generates number of tosses
plot(tosses,a_purse,tosses,b_purse,tosses,c_purse)
title('Bankroll growth with Kelly fraction,')
ylabel('Log(Bankroll)')
xlabel('Coin flips')
set(gca, 'yscale', 'log')
```

Appendix 2

```
%READ CSV FILE
%This creates a matrix without the headings
clear, clc
str = 'SPY';
fileID = fopen([str '.csv']);
C = textscan(fileID, '%s%*f%*f%*f%*f%*f%f','HeaderLines',1,'Delimiter',',');
fclose(fileID);

date = C{1,1}; %First cell contains dates
date_format= 'yyyy-mm-dd';  %used to convert to datenum
date=datenum(date,date_format);


closing = C{1,2}; %Second cell contains closing values

date=flipud(date); %reverse the order of date
closing=flipud(closing); %reverse the order of date


%need first date in date array: date_1=date(1,1)
% give the number of elements in array numel(date, :, 1)

delta_t = 21;
%Define returns function. This For function steps through the date array
% and performs the returns alogoithm, saving it to an array called Returns
bankroll = 5000; %Assume a bankroll of $5000
for m = delta_t+1:numel(date, :, 1) %starts with delta t and ends with number
of elements in date array
    Returns(m) = (closing(m)-closing(m-delta_t))/closing(m-delta_t)*100;

    pos = Returns(Returns>0);%returns matrix with positive values
    avgpos(m) = mean(pos);%avg of positive values
    p(m) = numel(pos, 1, :)/numel(Returns, 1, :); %wins / (total returns)
    q(m) = 1-p(m); %probability of losing

    neg = Returns(Returns<0);
    avgneg(m) = mean(neg);

    b(m) = (1+avgpos(m)/100)/abs(avgneg(m)/100); %win to loss ratio
    f(m) = (b(m)*p(m)-q(m))/b(m);
    f(f<0)=0;%make negative values of f equal to zero.
    f(isnan(f))=0;%make NaN values of f equal to zero.

    bankroll = (bankroll -f(m)*bankroll) + f(m)*bankroll*(1+Returns(m)/100)-
9;%need to annualize, trading fee assessed
    br(m) = bankroll;
    br(br<0)=0;%make negative values of br equal to zero.
```

```
end


avgreturns = mean(Returns)

drawdown=min(Returns(:)) % returns the minimum value in array:

figure(1)
plot(date, Returns)
title(['Trailing Returns ' str ', Drawdown =' num2str(drawdown) ',\Delta t =
' num2str(delta_t), ', R_{avg} =', num2str(avgreturns), '%'])
ylabel('%')
xlabel('Date')
datetick('x','mm/dd/yyyy', 'keepticks')
axis ([date(delta_t) max(date(:)) min(Returns(:)) max(Returns(:))])

figure(2)
plot(date, f)
title(['Kelly fraction ' str ', Drawdown =' num2str(drawdown) ',\Delta t = '
num2str(delta_t), ', R_{avg} =', num2str(avgreturns)])
ylabel('f')
xlabel('Date')
datetick('x','mm/dd/yyyy', 'keepticks')
axis ([date(delta_t) max(date(:)) min(f(:)) max(f(:))])

figure(3)
plot(date, br)
title(['Bankroll  ' str ', Drawdown =' num2str(drawdown) ',\Delta t = '
num2str(delta_t), ', R_{avg} =', num2str(avgreturns)])
ylabel('br')
xlabel('Date')
datetick('x','mm/dd/yyyy', 'keepticks')
axis ([date(delta_t) max(date(:)) min(br(:)) max(br(:))])
%set(gca, 'Yscale', 'log')
```

References

Kelly, J. L., Jr. (1956), "A New Interpretation of Information Rate", Bell System Technical Journal 35: 917–926

R. Mantegna and H.E. Stanley, An Introduction to Econophysics, Cambridge (2000)