Appl. Statist. (2007)
56, *Part* 5, *pp.* 607–623

Algorithms for optimal allocation of bets on many simultaneous events

Chris Whitrow

Imperial College London, UK

[Received September 2006. Revised June 2007]

Summary. The problem of optimizing a number of simultaneous bets is considered, using primarily log-utility. Stochastic gradient-based algorithms for solving this problem are developed and compared with the simplex method. The solutions may be regarded as a generalization of 'Kelly staking' to the case of many simultaneous bets. Properties of the solutions are examined in two example cases using real odds from sports bookmakers. The algorithms that are developed also have wide applicability beyond sports betting and may be extended to general portfolio optimization problems, with any reasonable utility function.

Keywords: Gambling; Kelly staking; Log-utility; Portfolio optimization; Sports betting; Stochastic gradient ascent

1. Introduction

Two algorithms are presented to solve the problem of maximizing long-term return from a series of bets on events, which may occur in simultaneous groups. The solution to this problem has widespread applications in investment strategy and gambling games. In particular, this paper considers the problem that is faced by those betting on fixed odds sports events. An investor may identify many bets on a given day, all of which offer value in the odds (i.e. positive expectation). For example, over 40 matches are played, mostly at the same time, on a typical Saturday of the English football league season. It is possible that many of these will offer the chance to make a profitable bet. For example, the investor may have a statistical model which gives good estimates of the probabilities of each result $\{p_i\}$. For models relating to football, see for example Dixon and Coles (1997), Crowder *et al.* (2002) or Goddard (2005). The result probabilities from such models can be compared with the best odds that are offered on each result, represented as the return on a unit stake $\{r_i\}$, which is referred to as the 'decimal' odds by bookmakers. The set of 'value' bets, V, is $\{i : r_i p_i > 1\}$. These are the events for which the odds are more generous than they should be, giving the gambler a positive expectation.

In what follows, we must assume that we have a good model in the sense that it gives unbiased estimates for the event probabilities $\{p_i\}$. This may be difficult in practice but, with sufficient data and care, it should be possible to reduce the bias to a sufficiently small level. This still does not guarantee that the model will be sufficiently powerful to identify many good bets, given that the odds are never 'fair'. In the author's own experience, however, it is possible to develop good models for football betting and to use these in conjunction with the algorithms that are presented in this paper to produce substantial profits in the course of a season. Of course, sports betting is not the only context in which these methods are of interest.

Address for correspondence: Chris Whitrow, Institute for Mathematical Sciences, Imperial College London, 53 Princes Gate, South Kensington, London, SW7 2PG, UK.
E-mail: c.whitrow@imperial.ac.uk

 0035–9254/07/56607

More generally, there are many situations in which gamblers or investors may identify numerous good (though not risk-free) investments and need to consider how to allocate their available resources in order both to maximize their returns and to minimize risk. Faced with numerous simultaneous positive expectation bets, how should a gambler allocate the stakes, given some assumed known probabilities $\{p_i\}$, along with the known returns $\{r_i\}$? In this paper, we shall usually assume that the events are independent, although the methods for dealing with these can be extended to include various types of dependence, such as disjoint events (for example a home win and a draw are disjoint and therefore dependent events on a single football match although different matches are assumed independent).

A novel algorithm will be presented which solves the problem of allocating stakes on simultaneous events, to maximize expected log-utility (the so-called 'Kelly criterion'). This is a form of 'portfolio choice' problem in investment theory which, in many dimensions, appears to have no closed form solution. Even good approximations are difficult to find. Although the problem has been considered before (Hakansson, 1970), no general method of exact solution appears to have been presented in the case where many possible investments can be combined. A stochastic gradient ascent method is therefore developed to solve the problem and is demonstrated by using real data from football games together with simulated outcomes. Despite its stochastic nature, the algorithm is both fast and accurate. Dozens of simultaneous bets may be optimized in a few seconds on a personal computer. The method may also be adapted easily to deal with disjoint as well as independent events. It should also be possible to extend the method to cope with other kinds of dependence between different events, with the addition of Markov chain Monte Carlo methods such as Gibbs sampling.

The ability to allocate resources across a range of investments simultaneously allows the investor to reduce the risk at the same time as increasing expected returns. One of the objections that sports bettors often raise regarding 'Kelly betting' is the high volatility of the resulting returns. This is particularly marked if one can bet only on one event at a time from a pool of simultaneous events. The algorithms that are presented here, by generalizing the 'Kelly staking method', offer the prospect of enhanced returns with lower volatility. This is achieved by spreading the risk among many positive expectation investments simultaneously.

2. Review

2.1. Utility functions

Several special cases of the general problem have been examined and solved previously. Kelly (1956) used an information theoretic formulation to show how to maximize the expected log-return on a series of single events. Kelly recommended that the gambler should attempt to maximize the expectation of the logarithm of his or her overall bankroll. When faced with a favourable game, this seems best in the long run. For example, if we attempt to maximize the expected bankroll itself at each step, the strategy would be to bet the entire bankroll on any game with positive expectation. Eventually, the gambler will lose and be instantly ruined. Ruin is certain if we follow this strategy. Even if we do not bet the entire bankroll, it is still true in general that playing too aggressively leads to ruin in the long run *even when the game is in the player's favour*. A more conservative utility function is therefore required.

The Kelly criterion is to maximize log-utility. This ensures that ruin is avoided and leads to a proportional betting strategy in which one always bets a proportion of the bankroll, which in turn depends on the size of the gambler's 'edge'. Breiman (1961) analysed the log-utility criterion and found that it gave a rise to the best strategy according to some well-defined mathematical criteria. Nevertheless, some controversy remains regarding the correct choice of utility function.

This is understandable, since the question is ultimately not a mathematical one. For example, how sad would you be if your wealth were reduced by half? How happy would you be to double your wealth? Many people have a preference for the log-utility function. Unlike some functions, it does not allow any chance of ruin (since the associated utility is $-\infty$) and is in this sense conservative. Breiman (1961) showed that in the long term, though, it leads to higher growth than any other strategy.

Although log-utility is optimal for growth, it can lead to a high volatility of returns, to which many investors are averse. MacLean *et al.* (1992) showed that an efficient trade-off between growth and security could be achieved by using the so-called 'fractional Kelly' strategy in which a fixed fraction of the player's bankroll is set aside in a riskless asset on each iteration. In effect, this means that the log-optimal bet (or bets) is reduced by some fixed proportion. This not only leads to lower volatility of returns but can also guard against the overbetting which might follow from the use of a misspecified (i.e. biased) model for the event probabilities.

In this paper, results will be presented using only the log-utility function. This is because of its optimality in terms of growth, at the same time as avoiding ruin and being capable of easy adaptation to achieve a combination of growth and security. It is also seemingly the most studied of utility functions for investment purposes, as attested by the literature cited. This allows us to appeal to some theoretical results. In addition, the Kelly criterion has been used by many gamblers and investors in practice with great success. See Thorp (1984, 1997) or Poundstone (2005) for examples.

Although there are many good reasons to focus on the log-utility function, it must be emphasized that the methods that are presented here are in no way limited to using log-utility. They can be adapted very easily for use with any reasonable utility function. In particular, there is a class of utility functions, which is concave on the interval $(0, \infty]$, which are closely related to log-utility and for which the algorithms require the addition of only a single parameter. These are the functions $U(x) = -(x^{-\alpha})$ with $\alpha > 0$ which also have the property of avoiding ruin. Although these functions are not optimal for growth, some investors may prefer their security properties. Later, it will be shown how closely these functions relate to log-utility and how the latter may be regarded as a special case. In particular, the algorithms that are given later will handle all these functions with minimal modification. Their concavity also guarantees that an optimal solution can be found by gradient ascent.

2.2. The Kelly staking strategy

Here, we give a reminder of the staking strategy for maximizing long-term growth rate over a series of single bets. Let the initial bankroll be B_0. This is the gambler's starting capital. It represents his entire financial resources available for betting. If he loses this amount, the whole game is over: no more bets may be placed and the player is ruined. We assume that the player is in the fortunate position of having found a positive expectation game and is able to play this game repeatedly for n iterations, after which the bankroll is B_n. On any given iteration t, he places a bet (which may be 0) on a single event with two outcomes. The probability of winning is known to be p, and the return on a unit stake is r, where we assume $rp > 1$ (for a positive expectation game). Note then that r is equivalent to the 'decimal odds' that are used by European bookmakers, so $r > 1$ always. 'English' odds of 'evens' correspond to $r = 2$, for example. In general, r = English odds + 1.

This is a very simple but important scenario, which will illustrate some general points, as well as having direct application to real problems. Generalization of the solution to games with numerous disjoint discrete outcomes (such as in horse-racing) was also achieved in the original paper by Kelly but is slightly more complicated. Approximate solutions have also been obtained for games where the return r is a (possibly continuous) random variable, as in spread betting (Haigh, 2000; Browne and Whitt, 1996). We ignore these possibilities for now but will consider them again later. Ultimately, we also want to address the problem of what the player should do when there is a choice of events to bet on at each iteration. For now, consider only a single event at each iteration and the player must decide how much of his bankroll to risk.

There must be some fraction x of the player's bankroll which can be bet at iteration t, so that the expected growth of the bankroll will be greatest. To see this, define

$$Q_n = \log\left(\frac{B_n}{B_0}\right) = \log(B_n) - \log(B_0). \tag{1}$$

Our aim is to maximize $E[Q_n]$ or, if we define $\delta Q_t = Q_t - Q_{t-1} = \log(B_t) - \log(B_{t-1})$,

$$E[Q_n] = \sum_{t=1}^{n} E[\delta Q_t]. \tag{2}$$

Hence, one should maximize the expected value of δQ_t at each iteration separately, where

$$E[\delta Q_t] = p\log(1 - x + rx) + (1-p)\log(1-x). \tag{3}$$

Here, the quantities on the right-hand side are treated as constant, since we are only concerned with a single iteration of the game. In general, p (the probability of winning the bet) and r (the return per unit stake) may vary with each iteration t. Consequently, the optimal proportion of player's capital to wager, x, will also vary. However, the optimal value of x depends only on p and r at each t. It is easy to find by setting the derivative of the expectation, equation (3), to 0 so that

$$x = \frac{pr-1}{r-1}. \tag{4}$$

The numerator is the expected profit, whereas the denominator can be regarded as a measure of risk, or dispersion of possible returns. Naturally, it only makes sense to place a bet when $pr > 1$ (expectation of gain is positive). Another expression for the optimum fraction x is obtained by using $\tilde{p} = 1/r$, the probability that is implied by the odds. Hence, $\tilde{p}$ is the probability corresponding to a 'fair' game (zero expectation of gain) when the odds are r. Then we have

$$x = \frac{p - \tilde{p}}{1 - \tilde{p}}. \tag{5}$$

Substituting this back into the expression for $E[\delta Q_t]$ yields an interesting formula for the optimal expected log-growth rate:

$$E[\delta Q_t] = p\log\left(\frac{p}{\tilde{p}}\right) + (1-p)\log\left(\frac{1-p}{1-\tilde{p}}\right) = D_{\mathrm{KL}}(P||\tilde{P}). \tag{6}$$

Thus, the mean log-growth rate is equal to the Kullback–Leibler information divergence, $D_{\mathrm{KL}}(P||\tilde{P})$, between the distributions that are implied by the probabilities, for $p > \tilde{p}$ (Kullback and Leibler, 1951; Cover and Thomas, 1991). Edelman (2000) has also shown that this is the maximum achievable growth rate in the most general case, when there is a range of possible pay-outs for different outcomes (i.e. when r is regarded as a random variable). Another point to note is that there is some critical value of x for which the expected log-return becomes negative, even when $pr > 1$.

The 'Kelly stake', then, is simply a proportion x of the player's capital (bankroll) at each iteration, given by equation (4). This is fine when there is only one game to bet on each time. Complications arise when there is a choice of games, with different odds and probabilities, or if r itself can assume a spectrum of values for each such game. Stochastic optimization methods can be applied to optimize multiple stakes, as will be demonstrated. The strategy is to perform approximate gradient ascent to maximize the utility function. Unfortunately, exact calculation of the gradient is infeasible, owing to the potentially huge number of terms which must be calculated, but an arbitrarily precise estimate of the gradient is easily computed by Monte Carlo simulation, as shown in the next section.

3. Optimization of simultaneous bets

Now we consider the general problem of allocating stakes across a number of possible events, with outcomes G_i, each with 'win' probability p_i and an associated random variable z_i representing the profit that is obtained on that bet per unit stake, i.e. $z_i = r_i - 1$ (decimal odds minus 1) for a winning bet and $z_i = -1$ for a losing bet on game i when the bet has only two outcomes. The restriction to binary outcomes is not necessary in what follows, however. The random variable z_i could even be continuous. The problem is to allocate stakes x_i so as to maximize the expected log-return over all possible joint outcomes, $S = \{G_1, \ldots, G_n\}$. For compactness, let us introduce the function

$$R(S) = 1 + \sum_{i=1}^{n} z_i(S)x_i \tag{7}$$

where S is a particular set of outcomes of the n events. Thus $R(S)$ is the total return from the joint outcome S and the dependence of z_i on S has been made explicit. Now the utility is

$$U(x_1, \ldots, x_n) = \sum_S \Pr(S) \log\{R(S)\}. \tag{8}$$

Here, $\Pr(S)$ is simply the joint probability of a particular set of outcomes S and U is the log-utility function, as before. Taking the derivative with respect to a particular stake, say x_k, gives

$$\frac{\partial U}{\partial x_k} = \sum_S \Pr(S) \frac{z_k(S)}{R(S)} \tag{9}$$

or in more compact notation

$$\frac{\partial U}{\partial x_k} = \left\langle \frac{z_k}{R} \right\rangle_S \tag{10}$$

where $\langle \ldots \rangle_S$ signifies the expectation over all joint outcomes (of the n events). There is an implied inequality constraint: $\Sigma_{i=1}^{n} x_i \leqslant 1$. Later, this constraint will also be expressed as an equality, although much progress can be made by considering only the unconstrained problem. It is also worth noting here that if we use a utility function of the form $U(x) = -(1/\alpha)x^{-\alpha}$ then equation (10) becomes

$$\frac{\partial U}{\partial x_k} = \left\langle \frac{z_k}{R^{\alpha+1}} \right\rangle_S. \tag{11}$$

Hence, we see that maximizing log-utility is equivalent to using equation (11) with $\alpha = 0$. Intuitively, larger values of α would give greater weight to the terms where R is small, when calculating the gradient. The resulting allocation of stakes can be expected to be more cautious in some sense. In general, varying α provides another mechanism for adjusting the growth–security trade-off. We could even contemplate the use of any $\alpha > -1$. This raises many questions, which are beyond the scope of this paper. In the simulations that are presented later, only equation (10) is used, as we restrict our attention to log-utility.

The space of joint outcomes S is potentially enormous. Exact calculation of each $\partial U/\partial x_k$ involves m^n terms, for n games with m outcomes each. However, Monte Carlo estimates can be made by using K simulations, each using just n simulated outcomes (one for each event) using the probabilities p_i, assuming that the events are independent. Dependent events may be more difficult to simulate, although Markov chain Monte Carlo methods could be applied.

Some further insight may be gained by attempting an approximate solution to equation (10) without resorting to a Monte Carlo method. We must first assume that event k is independent of the others and note that

$$R(S) = R_k + z_k x_k \tag{12}$$

where R_k is the return on all events other than k:

$$R_k = 1 + \sum_{i \neq k} z_i x_i.$$

We may therefore express equation (10) as a Taylor series expansion in x_k:

$$\frac{\partial U}{\partial x_k} = \left\langle \frac{z_k}{R_k} - \frac{z_k^2}{R_k^2} x_k + \frac{z_k^3}{R_k^3} x_k^2 - \frac{z_k^4}{R_k^4} x_k^3 + \ldots \right\rangle_S . \tag{13}$$

This assumes, quite reasonably, that $\|z_k x_k\| < R_k$. If we make the further assumption that $z_k x_k$ is sufficiently small for every outcome in S, we may ignore terms of $O(x_k^2)$ or higher order. In many cases we may also assume that $R_k \approx 1$, or at least that it is approximately constant. Thus, taking R_k constant, putting $\partial U/\partial x_k = 0$ and solving for x in first order yields

$$\frac{x_k}{R_k} = \frac{\langle z_k \rangle}{\langle z_k^2 \rangle}. \tag{14}$$

This is quite a well-known approximation, which is used among the blackjack community (Thorp, 1997). It is only valid for 'small expectation' games and often the variance is used in place of the second moment in the denominator, since, when $\langle z_k \rangle$ is small, $\mathrm{var}(z_k) \approx \langle z^2 \rangle$. In blackjack, where the player's edge is usually small (less than 2%) and one is usually playing only one or two hands, the formula is reasonable. However, when the number of independent events rises and the expected profit on any one of them becomes more than about 5%, the 'small expectation' formula becomes increasingly inadequate. Moreover, the formula tends to overestimate the stakes, thus 'erring on the side of incaution', which could lead to disaster. For the sports bettor hoping to bet on many games simultaneously, it is effectively useless. One of the conclusions of this paper is that approximation is neither adequate nor necessary. Instead, a Monte Carlo solution to the exact problem is quite easily found and is far preferable.

Two approaches to the Monte Carlo solution will be described: one is an unconstrained optimization in which the stakes are kept within the valid range by simple renormalization. For example, if any parameter update would force a stake x_k to take a negative value, it is set to 0 instead. Similarly, if the sum of all the stakes becomes greater than 1, they are immediately rescaled so that their sum is just below 1.

The other approach is to constrain the stakes by the use of auxiliary variables. We may assume that $\Sigma_i x_i = 1$ without loss of generality, since we may consider an event θ in S with $p_\theta = 1$ and $z_\theta = 0$. This event corresponds to withholding a proportion of the bankroll x_θ, effectively placing no bet with it. With the stakes now constrained, we may express the x_i via the multinomial logit transformation in terms of auxiliary variables t_i:

$$x_i = \frac{\exp(t_i)}{\sum_j \exp(t_j)}. \tag{15}$$

This ensures that $x_i > 0, \forall i$, and that $\Sigma_i x_i = 1$. The constrained optimization of log-utility (equation (8)) now becomes an unconstrained optimization in the space of auxiliary parameters t_i and the gradient can be expressed in terms of the new variables:

$$\Delta t_l = \frac{\partial U}{\partial t_l} = \sum_k \frac{\partial U}{\partial x_k} \frac{\partial x_k}{\partial t_l} \tag{16}$$

where Δt_l simply denotes $\partial U/\partial t_l$ (this notation is justified since the gradient is proportional to the step change in t_l that is required to increase U). Similarly, if $\partial U/\partial x_k$ is denoted by Δx_k, differentiating equation (15) with respect to t_l gives

$$\Delta t_l = \sum_k \Delta x_k \frac{\delta_{kl} \exp(t_l) \sum_i \exp(t_i) - \exp(t_l) \exp(t_k)}{\left\{ \sum_j \exp(t_j) \right\}^2}. \tag{17}$$

Now we may also constrain the t_i so that $\Sigma_i \exp(t_i) = 1$. This can always be done by the addition of a constant to each t_i, for example. In practice, we shall calculate the appropriate values for t_i at each iteration from the existing x_i:

$$t_i = \log(x_i). \tag{18}$$

The expression (17) for the gradient now simplifies:

$$\begin{aligned} \Delta t_l &= \exp(t_l) \sum_k \Delta x_k \{\delta_{kl} - \exp(t_k)\} \\ &= x_l (\Delta x_l - \sum_k \Delta x_k x_k). \end{aligned} \tag{19}$$

The auxiliary variables have been eliminated from the right-hand side in equation (19), so the gradient with respect to the t_i has been re-expressed in terms of the original gradient and stakes. This allows the t_i to be calculated first by using equation (18), then updated by using equation (19) and finally the x_i are updated by using equation (15). In effect, the auxiliary variables serve only to modify the update rule and can be eliminated entirely from the solution if desired. Enforcement of the constraint by equation (15) leads to a more effective optimization procedure with good convergence, as will be demonstrated.

Before looking at the results, one final point must be addressed: are we finding a local or global optimum? Fortunately, Breiman (1991) showed that, although there may be more than one optimum, they are all global, in the sense that all optima have the same value of U. This follows from the concavity of the log-function. So, there may for example be symmetries among the odds and event probabilities which define the problem space and these will lead to symmetries in the solution (and hence multiple optima) but these are all equivalent in practice. Our algorithms, although stochastic, should therefore converge if we are careful about the step size. However, no more formal consideration of convergence matters is given here. Instead, we now discuss some examples and demonstrate performance in practice.

4. Results

We compare principally two variants of the gradient following algorithm that was sketched above. The first ignores the constraint on stakes and uses equation (10) to estimate the gradient. When the stakes are updated, it is possible for the total to exceed 1. This is avoided by merely rescaling the stakes (if necessary) after any such update so that they sum to a value that is very close to 1. We shall refer to this as the 'naïve' gradient algorithm. It is almost equivalent to introducing a Lagrange multiplier but avoids the problem of stakes becoming negative by resetting such stakes to 0. The second algorithm is the full algorithm that was described above in which the constraint is enforced by using auxiliary variables. This we shall call the 'constrained' algorithm.

Both algorithms use a dynamically adjusted learning rate (i.e. step size for changing the stakes) to speed up convergence, i.e. the learning rate is increased (multiplied by 1.05 say) if the gradient does not change sign but is decreased (multiplied by 0.95) if it does. $K = 100$ simulated joint outcomes are used to estimate the gradient in each case. This is a remarkably small number, yet it seems to produce good convergence every time. Using more simulations does improve the gradient estimate but takes longer and does not seem to improve the final solution.

Initialization is an important issue but will not be discussed at length here. Several simple schemes were used. In the main, stakes were initialized to the simple Kelly stake as given by equation (5). The amount to set aside (not for betting) was initialized to 1. All stakes were then rescaled so that the total (including the proportion that was not risked) equals 1. Thus, the stake for a given event is initialized to be proportional to the Kelly stake for that event. This may be varied by initializing the amount that is set aside to be proportional to the mean Kelly stake (over all events) or to a very small value (0.01 or 0.001). As we shall see, the appropriate scheme seems to depend on the number of events.

We compare results by using two sets of plausible events with associated probabilities and odds, taken from English football league fixtures. The odds are actual bookmakers' odds and the probabilities are estimated by using a logistic regression model and game data (details of which are omitted). It is sufficient to assume that we can establish an advantage compared with the official odds, on each event in the set.

The first set of data (data set 1) contains 12 events with probabilities ranging from 0.26 to 0.53 and an average of 0.36. The odds range from 2 to 4.33, with an average of 3.13. The average 'edge' (the difference between actual probability and implied probability based on the odds) is 2.2%. The average expectation per event is $pr - 1$ (as in equation (4)) and this is 6.6% (the expected return on a unit stake) over the 12 events.

The second set of data (data set 2) contains 37 events with probabilities ranging from 0.13 to 0.78 (average 0.41) and odds from 1.3 to 8.0 (average 3.25). The edge is 3.0% on average and the mean expected return is 9.3%.

4.1. Data set 1

Convergence to a solution is fast for the naïve algorithm. No further optimization occurs after about 300 iterations. This is despite the fact that the learning rates have not decayed very much (a separate learning rate is employed for each stake and they are modified independently). This suggests that the utility function has reached a good optimum, which is confirmed when the estimated gradients are examined. The mean gradient from iteration 300 onwards is less than 3×10^{-4} for all stakes.

Convergence is noticeably slower for the more sophisticated 'constrained' algorithm. A close examination shows that after about 5000 iterations the gradients are all less than or equal to

Table 1. Algorithm comparison (data set 1)

Data set 1	*Results from the following algorithms:*	
	Naïve	*Constrained*
E[log-return] $\pm 6 \times 10^{-5}$	0.0196	0.0195
E[return] $\pm 6 \times 10^{-3}$	3.9%	3.8%
Standard deviation (log-return)	0.197	0.190
Standard deviation (return)	19.9%	19.3%
Sharpe ratio	0.197	0.197

5×10^{-3}. The solution can be improved by running the algorithm for a longer time but it takes a large number of iterations to approach the solution that is found by the naïve algorithm. The total stakes committed by the constrained algorithm amount to 38% of the bankroll, whereas the naïve algorithm commits 40%.

The actual quality of the two solutions can be investigated by means of a further Monte Carlo simulation, but using a great many more sampled outcomes (100 is nowhere near enough to estimate the actual log-utility). After optimizing the stakes, 10 million simulated outcomes were used to estimate the actual expected return (on the bankroll) as well as the expected log-return. These are shown in Table 1 along with the estimated standard errors. The standard deviations of both quantities were also calculated. We also quote the Sharpe ratio (Sharpe, 1994) as a measure of return *versus* risk. This is the ratio of the mean to the standard deviation of the return. In practice, there is little difference between the solutions that are provided by the two algorithms on this small number of bets.

The actual optimized stakes for both algorithms are shown in Table 2, where each row corresponds to an event (in this case a football match). Also given are the odds, the implied probability (according to the bookmakers equal to 1/odds), the actual probability (as estimated from our forecasting model) and the Kelly stake. It is interesting to note that both solutions are very close to the Kelly stakes on each bet taken separately, calculated according to equation (4) and shown in descending order. This is perhaps unsurprising.

Table 2. Sizes of bet for each algorithm (data set 1)

Event	*Actual probability*	*Bookmaker's probability*	*Odds*	*Kelly stake (%)*	*Constrained bet size (%)*	*Naïve bet size (%)*
1	0.470	0.400	2.50	11.7	11.2	11.5
2	0.530	0.500	2.00	6.0	5.6	5.8
3	0.480	0.455	2.20	4.7	4.4	4.5
4	0.310	0.286	3.50	3.4	3.2	3.3
5	0.255	0.231	4.33	3.2	2.9	3.1
6	0.270	0.250	4.00	2.7	2.5	2.6
7	0.270	0.250	4.00	2.7	2.5	2.6
8	0.311	0.294	3.40	2.4	2.2	2.3
9	0.310	0.294	3.40	2.3	2.1	2.2
10	0.466	0.455	2.20	2.2	1.7	2.1
11	0.371	0.370	2.70	0.2	0.1	0.2
12	0.304	0.303	3.30	0.1	0.1	0.1

Although the results on only two data sets have been presented, the author notes that, in all the other cases he has observed where the number of bets is low and the total committed is considerably less than 100% of the bankroll, the optimal stakes are closely proportional to the Kelly stake for each bet. Note that the constant of proportionality is less than 1, although in this case it is quite high at 0.98. A proof of this assertion has not been attempted and it is only mentioned as an observation concerning certain sets of events, only one of which is presented in detail here. In any case, this relationship certainly breaks down when the number and size of bets increases. However, it may lead to useful approximations when there are few events available for betting.

4.2. Data set 2

The second set of data reveals the usefulness of the constrained algorithm, although the naïve algorithm still performs well. Both were found to be considerably superior to the Nelder–Mead simplex algorithm (Nelder and Mead, 1965), as we might expect, given the use of gradient information. As before, the naïve algorithm converges very quickly, as can be seen from the evolution of the stakes over time (the number of iterations) in Fig. 1. By contrast, the constrained algorithm takes longer (Fig. 2). The estimated gradients (with respect to each stake) do diminish during optimization, as expected, although they do not reach 0 for either algorithm. For the naïve algorithm, the mean gradient settled at about 0.034, with the maximum at 0.085. The constrained algorithm resulted in a slightly higher mean gradient at 0.035 but the maximum was appreciably lower at 0.055.

The constrained algorithm was compared, as before, with the naïve algorithm. This time, the performance of each was also compared with that obtained by using the simplex method to optimize the utility function without the use of gradient information. The latter ran for around 8000 iterations before reaching a stable solution: further iterations did not improve performance. From Table 3, it is clear that the constrained algorithm achieves the highest value of the expected log-utility and that even the naïve algorithm achieves a much better optimum than the simplex method. However, we must note that the difference in expected returns spans only 2% across all three methods and that the Sharpe ratio is slightly better for the least optimal solution. These statistics were estimated by using 10 million simulated outcome sets, as before.

The actual solutions of each algorithm are given in Table 4. It can be seen that the constrained algorithm produces the most unequal sizes of bet, with a greater proportion of stakes committed to the most advantageous bets. The naïve algorithm shows a similar general pattern but gives more equal sizes of bet, whereas the simplex algorithm again gives similar stakes although these are more irregular.

Table 3. Algorithm comparison (data set 2)

Data set 2	*Results from the following algorithms:*		
	Naïve	*Constrained*	*Simplex*
E[log-return] $\pm 9 \times 10^{-5}$	0.0845	0.0869	0.0820
E[Return] $\pm 9 \times 10^{-3}$	12.5%	13.4%	11.3%
Standard deviation (log-return)	0.263	0.286	0.231
Standard deviation (return)	28.0%	30.2%	24.3%
Sharpe ratio	0.44	0.44	0.47

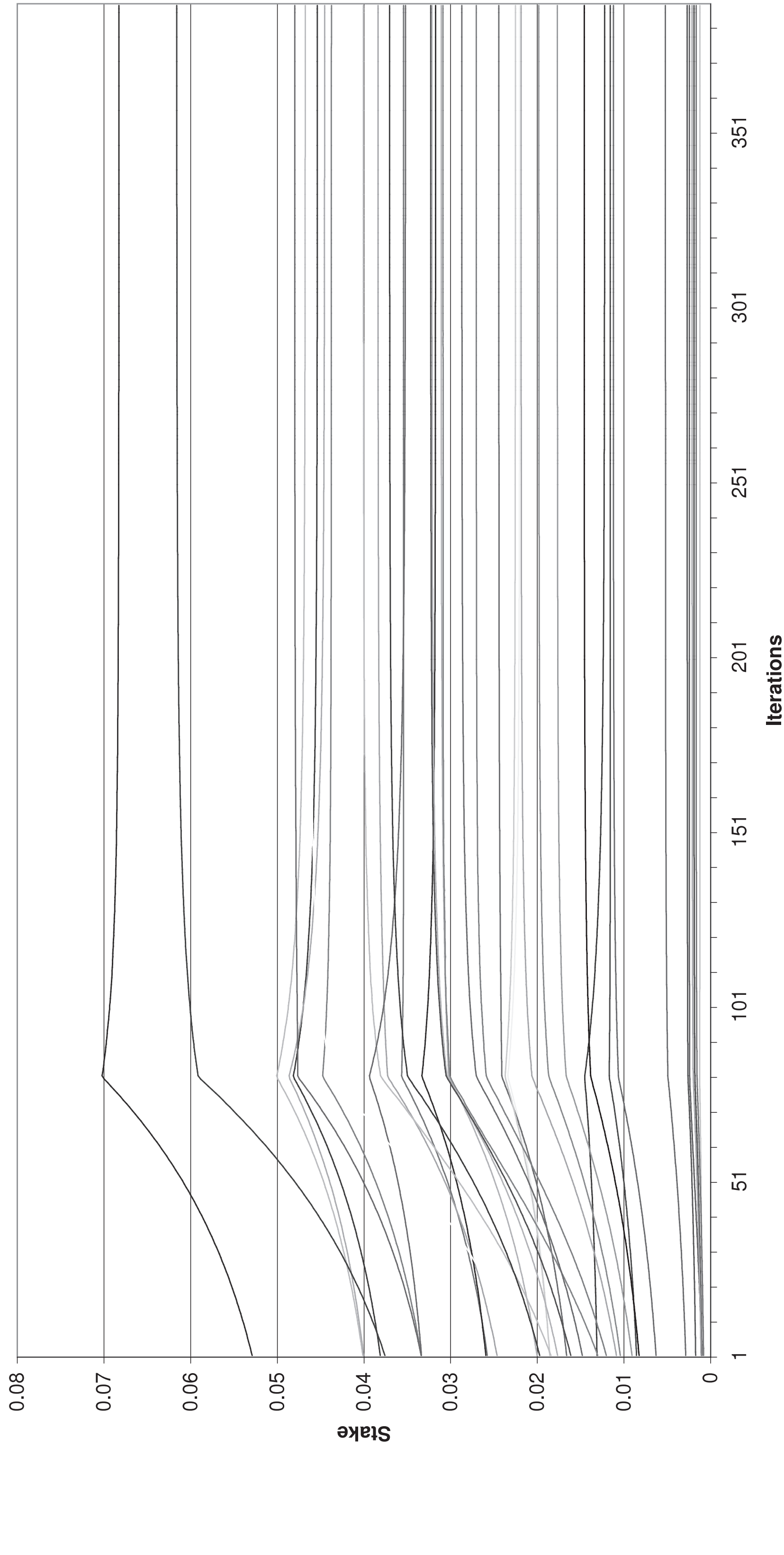

Fig. 1. Naïve algorithm stakes (data set 2)

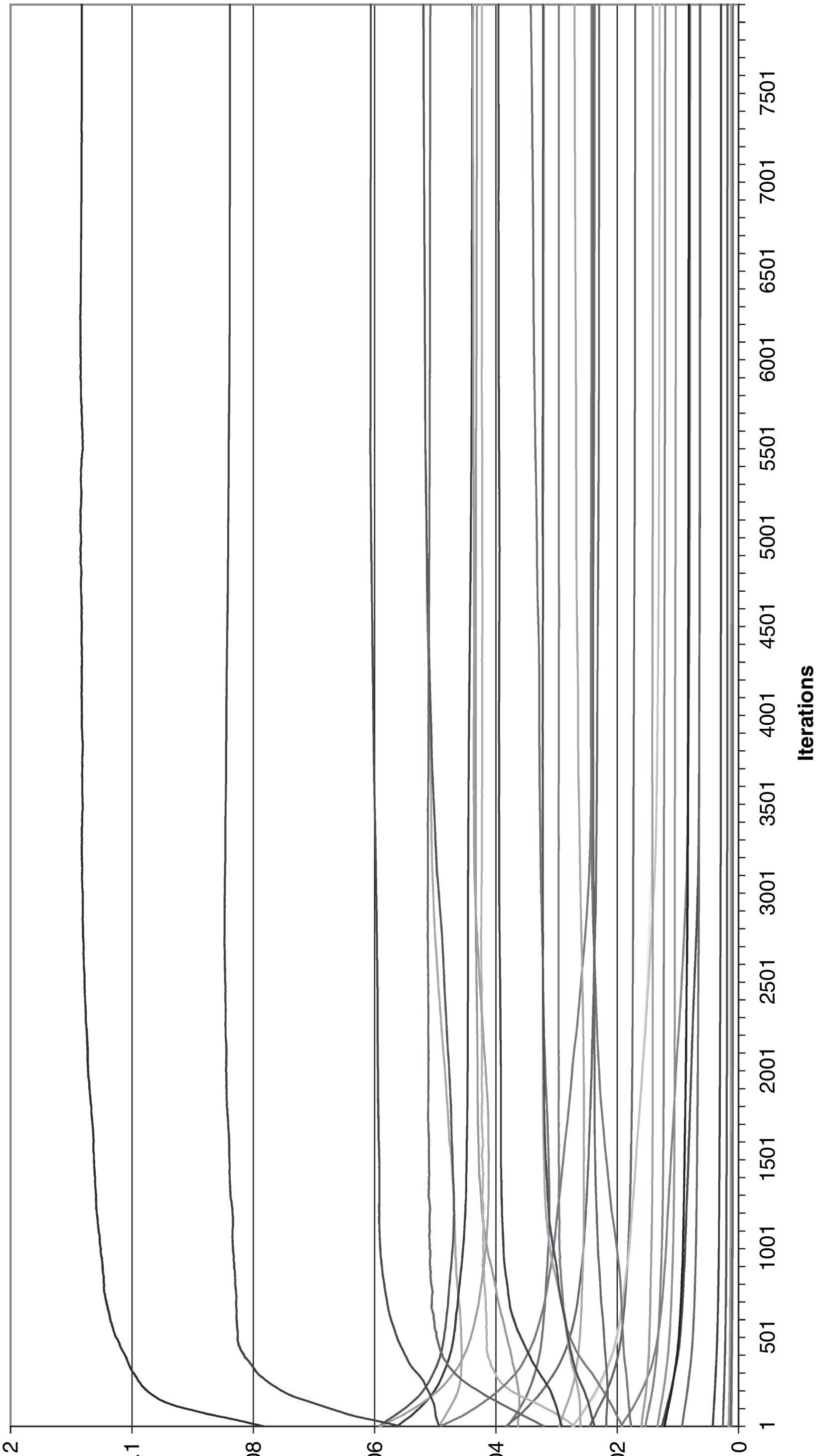

Fig. 2. Constrained algorithm stakes (data set 2)

Table 4. Sizes of bet for each algorithm (data set 2)

Event	*Actual probability*	*Bookmaker's probability*	*Odds*	*Kelly stake (%)*	*Constrained bet size (%)*	*Naïve bet size (%)*	*Simplex bet size (%)*
1	0.590	0.513	1.95	15.8	10.8	6.8	8.0
2	0.638	0.588	1.70	12.0	5.2	4.7	6.0
3	0.670	0.625	1.60	12.0	4.4	4.5	6.0
4	0.620	0.571	1.75	11.4	4.4	4.5	5.7
5	0.421	0.347	2.88	11.2	8.4	6.2	5.6
6	0.550	0.500	2.00	10.0	5.2	4.4	5.0
7	0.700	0.667	1.50	10.0	2.4	3.5	4.7
8	0.491	0.435	2.30	10.0	6.1	4.8	5.0
9	0.585	0.550	1.82	7.8	2.3	3.2	3.9
10	0.506	0.465	2.15	7.7	3.4	3.5	3.9
11	0.427	0.382	2.62	7.4	4.3	3.8	3.7
12	0.245	0.192	5.20	6.5	5.1	4.6	3.3
13	0.422	0.385	2.60	6.0	2.7	3.1	3.0
14	0.310	0.267	3.75	5.9	4.0	3.7	3.0
15	0.595	0.571	1.75	5.6	1.4	2.2	2.8
16	0.580	0.556	1.80	5.6	1.3	2.3	2.8
17	0.227	0.182	5.50	5.5	4.2	4.0	2.8
18	0.323	0.286	3.50	5.3	3.2	3.2	2.5
19	0.449	0.420	2.38	5.0	1.7	2.4	2.5
20	0.267	0.230	4.35	4.8	3.2	3.2	2.4
21	0.283	0.250	4.00	4.4	2.4	2.9	2.2
22	0.778	0.769	1.30	3.9	0.8	1.2	2.0
23	0.167	0.133	7.50	3.9	3.0	3.1	1.9
24	0.197	0.167	6.00	3.6	2.4	2.7	1.8
25	0.256	0.231	4.33	3.2	1.4	2.2	1.6
26	0.286	0.263	3.80	3.1	1.2	2.0	1.6
27	0.270	0.250	4.00	2.7	1.0	1.8	1.4
28	0.490	0.476	2.10	2.6	0.6	1.2	1.3
29	0.325	0.308	3.25	2.5	0.8	1.5	1.2
30	0.316	0.303	3.30	1.9	0.6	1.1	1.0
31	0.300	0.294	3.40	0.9	0.3	0.5	0.4
32	0.388	0.385	2.60	0.5	0.2	0.3	0.3
33	0.514	0.513	1.95	0.3	0.1	0.1	0.2
34	0.288	0.286	3.50	0.3	0.1	0.2	0.2
35	0.128	0.125	8.00	0.3	0.1	0.2	0.2
36	0.266	0.263	3.80	0.3	0.1	0.2	0.2
37	0.232	0.230	4.35	0.2	0.1	0.2	0.1

Note that the proportionality between the Kelly stake and the solution of the naïve algorithm is now much less precise. The Kelly stake is an even worse approximation for the (more optimal) stakes of the constrained algorithm. However, there is a better relationship between the stakes and the gambler's advantage or edge for each bet, which we define as the difference between actual and bookmaker's probabilities $p-\tilde{p}$. Fig. 3 shows this for the naïve algorithm. One could try other functions of p and $\tilde{p}$ but $p-\tilde{p}$ is the only linear combination that is consistent with the requirement that there should be no bet whenever $p=\tilde{p}$. The relationship between the edge and the stakes that is given by the constrained algorithm is better described by a quadratic function (Fig. 4). These relationships are also obtained if one tries a multivariate fit using powers of p and $\tilde{p}$. Although interesting, it is not easy to use these relationships to approximate the optimal bet sizes since the coefficients will depend on the data and there is no way to determine them other than empirically. There is also no guarantee that these relationships would continue to hold with radically different sets of events and odds.

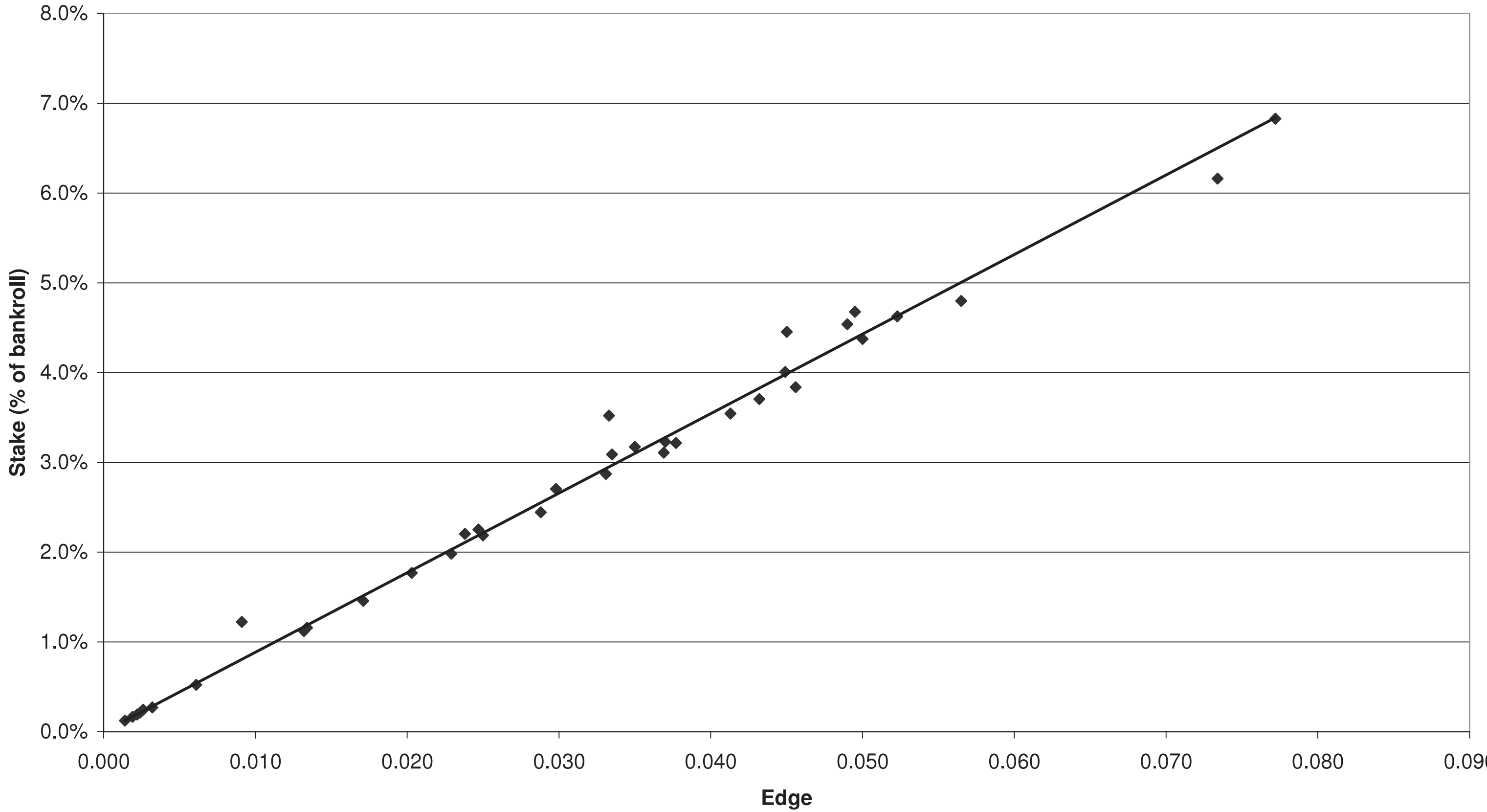

Fig. 3. Naïve algorithm stakes *versus* edge (data set 2): $y = 0.8858x$; $R^2 = 0.9892$

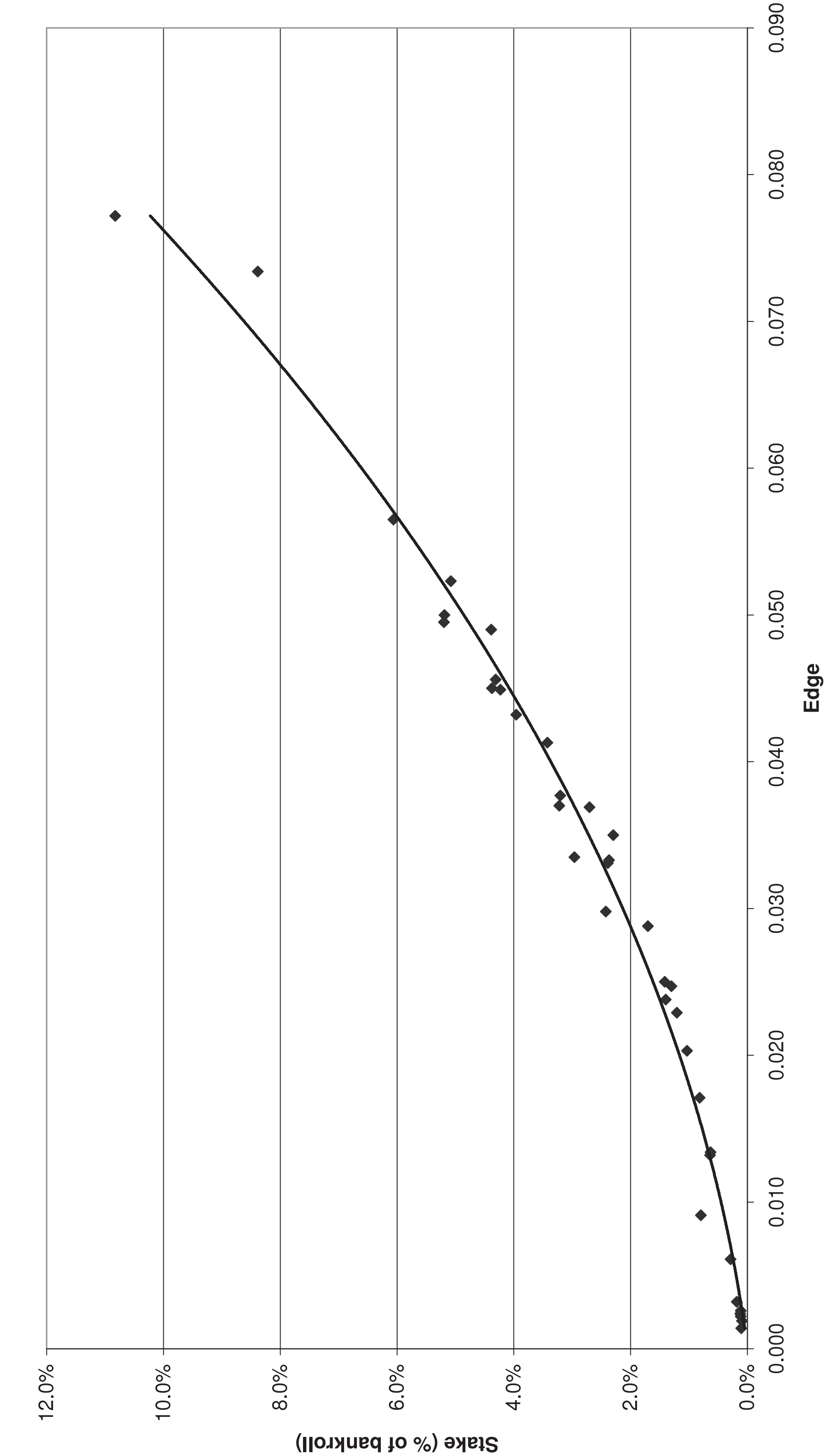

Fig. 4. Constrained algorithm stakes *versus* edge (data set 2): $y = 12.994x^2 + 0.3216x; R^2 = 0.9859$

5. Conclusions

The problem of optimizing log-utility, for multiple simultaneous bets or investments, can be solved satisfactorily by using a stochastic optimization technique. The solution can be obtained quickly, in a matter of seconds, even when dozens of bets need to be optimized. This eliminates the need for approximations which do not work well when there are many bets or when the gambler's advantage is large. The method of stochastic gradient ascent that is proposed here can easily be applied to other utility functions, although the log-utility has many desirable properties, including a unique optimum value. The method may also be applied to investments with continuous distributions of possible returns (e.g. 'spread' bets). It is even possible to take account of correlations between investments, as long as the outcome distribution can be simulated.

When the number of bets is small, the optimal sizes of bet seem to be almost exactly proportional to the Kelly stakes on individual bets. This appears to be true for all such sets of data that the author has so far examined although only one example is presented here. A proof of this hypothesis should perhaps be attempted as part of future work. At present, it is left as an interesting but tentative observation. As the number of bets increases and the sum of stakes approaches 100% of the bankroll, the optimal bet sizes are somewhat different, although roughly monotonically related to the Kelly stakes. In fact, a better approximation to the optimal size of bet for a given event in this case is to use the edge $p-\tilde{p}$ of that bet alone. These intriguing relationships were obtained empirically and it is unclear yet whether these or other relationships may be deduced rigorously.

Some readers may be surprised that the algorithms can recommend risking almost the entire bankroll. However, this is not inconsistent with optimizing log-utility. The sum of stakes is always less than 100%, so total ruin is always avoided. None-the-less, there is a very small probability of losing almost everything. This is also true if one uses traditional Kelly staking on a sequence of events and one should use a fractional Kelly strategy to reduce the risk (MacLean *et al.*, 1992). Another possibility is to use a different utility function, such as $U(x)=-(x^{-\alpha})$. A subject for future investigation would be to compare the optimized stakes, using such functions, with those which are obtained by using log-utility and to examine the effect on risk and return.

It is interesting to note that the overall Sharpe ratio of the solutions (from Table 3) does not vary by much. Indeed, the best Sharpe ratio was obtained by using the suboptimal simplex method. It seems that we cannot obtain a better return without a proportionate increase in risk.

For practical purposes, the naïve algorithm appears to give the best results when the number of events is small (a dozen or less, say). This has been observed on many data sets, although only two are presented here for illustration. It also gives satisfactory results as the number of events increases. The practical investor–gambler may prefer the sizes of bet that it generates, compared with the constrained algorithm, since they tend to be more equal and profits are therefore less dependent on a few 'big games'. It was also notable that the gradient methods gave consistently repeatable results, regardless of initialization. This was not so for the simplex method.

Interestingly, the theoretical optimum log-growth rate on data set 1 is 0.0199. This follows from the result of Edelman (2000), together with the additivity of Kullback–Leibler divergence over independent events, and is equivalent to the expected growth rate that is achievable if we could bet sequentially on these games. Our naïve algorithm comes surprisingly close (0.0196) to this optimum, despite the fact that bets must be made simultaneously. In theory, the optimum growth rate would be achievable if we could bet on every single joint outcome, although this is infeasible in practice. For data set 2, the theoretical optimum log-growth rate is 0.1121, which is somewhat higher than the 0.0869 that our best algorithm achieved (when limited to simultaneous 'singles' betting). The shortfall could be reduced by including possible bets on 'multiples' (joint outcomes) where these offer the greatest potential for profit. Although it would still be infeasible to bet on every conceivable joint outcome, certain multiples could be included if the corresponding edge $p-\tilde{p}$ were sufficiently large to suggest that an improvement in the overall expected log-return could be achieved. The bets would no longer be independent, but this is not a fundamental problem for our algorithms, based on equation (10). All that matters is that the full joint outcome distribution can be simulated. This could be done by using Gibbs sampling, for instance.

The question remains about how the returns are affected by errors in the model for the event probabilities p_i. We have been assuming that we know these probabilities. In spite of substantial differences between the stakes, though, all three algorithms achieved reasonable performance, as shown in Table 3, suggesting that the utility function around the region of optimality in 'stakes space' is probably fairly 'flat'. This means that we have some margin for error in the allocation of stakes, especially if we adopt a fractional staking strategy. Still, it would be better to know precisely how the utility function varies with the p_i as well as with the stakes. This question could be addressed in future by both theoretical and empirical studies.

Future work could also address the tuning of these optimization algorithms to improve accuracy and speed, although both appear adequate for current purposes. Extension of the algorithm to continuous returns (spread betting) and to take account of more complex dependences between events would also be interesting.

Acknowledgements

Thanks go to the referees, as well as Martin Crowder and Niall Adams of Imperial College, for numerous helpful comments on earlier drafts. This work was partially supported by a grant from the Institute for Mathematical Sciences, Imperial College London.

References

Breiman, L. (1961) Optimal gambling systems for favourable games. In *Proc. 4th Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, pp. 65–78. Berkeley: University of California Press.

Browne, S. and Whitt, W. (1996) Portfolio choice and the Bayesian Kelly criterion. *Adv. Appl. Probab.*, **28**, 1145–1176.

Cover, T. M. and Thomas, J. A. (1991) *Elements of Information Theory*. New York: Wiley.

Crowder, M., Dixon, M., Ledford, A. and Robinson, M. (2002) Dynamic modelling and prediction of English Football League matches for betting. *Statistician*, **51**, 157–168.

Dixon, M. J. and Coles, S. G. (1997) Modelling association football scores and inefficiencies in the football betting market. *Appl. Statist.*, **46**, 265–280.

Edelman, D. (2000) On the financial value of information. *Ann. Ops Res.*, **100**, 123–132.

Goddard, J. (2005) Regression models for forecasting goals and results in professional football. *Int. J. Forecast.*, **21**, 331–340.

Haigh, J. (2005) The Kelly criterion and bet comparisons in spread betting. *Statistician*, **49**, 531–539.

Hakansson, N. H. (1970) Optimal investment and consumption strategies under risk for a class of utility functions. *Econometrica*, **38**, 587–607.

Kelly, Jr, J. L. (1956) A new interpretation of information rate. *Bell Syst. Tech. J.*, **35**, 917–926.

Kullback, S. and Leibler, R. A. (1951) On information and sufficiency. *Ann. Math. Statist.*, **22**, 79–86.

MacLean, L. C., Ziemba, T. and Blazenko, G. (1992) Growth versus security in dynamic investment analysis. *Mangmnt Sci.*, **38**, 1562–1585.

Nelder, J. A. and Mead, R. (1965) A simplex method for function minimization. *Comput. J.*, **7**, 308–313.

Poundstone, W. (2005) *Fortune's Formula: the Untold Story of the Scientific Betting System that Beat the Casinos and Wall Street*. New York: Hill and Wang.

Sharpe, W. F. (1994) The Sharpe ratio. *J. Prtfol. Mangmnt*, **21**, 49–58.

Thorp, E. O. (1984) *The Mathematics of Gambling*. Secaucus: Lyle Stuart.

Thorp, E. O. (1997) The Kelly criterion in blackjack, sports betting and the stock market. Edward O. Thorp & Associates, Newport Beach. (Available from `http://www.bjmath.com/bjmath/thorp/paper.htm`.)