APPLIED STOCHASTIC MODELS IN BUSINESS AND INDUSTRY
Appl. Stochastic Models Bus. Ind. 2010; **26**:639–658
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asmb.874

A modern Bayesian look at the multi-armed bandit

Steven L. Scott*†

Google

SUMMARY

A multi-armed bandit is an experiment with the goal of accumulating rewards from a payoff distribution with unknown parameters that are to be learned sequentially. This article describes a heuristic for managing multi-armed bandits called *randomized probability matching*, which randomly allocates observations to arms according the Bayesian posterior probability that each arm is optimal. Advances in Bayesian computation have made randomized probability matching easy to apply to virtually any payoff distribution. This flexibility frees the experimenter to work with payoff distributions that correspond to certain classical experimental designs that have the potential to outperform methods that are 'optimal' in simpler contexts. I summarize the relationships between randomized probability matching and several related heuristics that have been used in the reinforcement learning literature. Copyright © 2010 John Wiley & Sons, Ltd.

Received 9 August 2010; Revised 1 November 2010; Accepted 1 November 2010

KEY WORDS: probability matching; exploration vs exploitation; sequential design; Bayesian adaptive design

1. INTRODUCTION

A *multi-armed bandit* is a sequential experiment with the goal of achieving the largest possible reward from a payoff distribution with unknown parameters. At each stage, the experimenter must decide which arm of the experiment to observe next. The choice involves a fundamental trade-off between the utility gain from *exploiting* arms that appear to be doing well (based on limited sample information) vs *exploring* arms that might potentially be optimal, but which appear to be inferior because of sampling variability. The trade-off has also been referred to as 'earn vs learn.' This article reviews several techniques that have been used to manage the multi-armed bandit problem. Particular attention is paid to a technique known as *randomized probability matching*, which can be implemented quite simply in a modern Bayesian computing environment, and which can combine good ideas from both sequential and classical experimental designs.

*Correspondence to: Steven L. Scott, Google.
†E-mail: stevescott@google.com

Multi-armed bandits have an important role to play in modern production systems that emphasize 'continuous improvement,' where products remain in a perpetual state of feature testing even after they have been launched. Online software (such as a web site, an online advertisement, or a cloud service) is especially amenable to continuous improvement because experimental variation is easy to introduce, and because user responses to online stimuli are often quickly observed. Indeed, several frameworks for improving online services through experimentation have been developed. Google's *web site optimizer* [1] is one well-known example. Designers provide web site optimizer several versions of their web site differing in font, image choice, layout, and other design elements. Web site optimizer randomly diverts traffic to the different configurations in search of configurations that have a high probability of producing successful outcomes, or *conversions*, as defined by the web site owner. One disincentive for web site owners to engage in online experiments is the fear that too much traffic will be diverted to inferior configurations in the name of experimental validity. Thus, the exploration/exploitation tradeoff arises because experimenters must weigh the potential gain of an increased conversion rate at the end of the experiment with the cost of a reduced conversion rate while it runs. Treating product improvement experiments as multi-armed bandits can dramatically reduce the cost of experimentation.

The name 'multi-armed bandit' alludes to a 'one-armed bandit', a colloquial term for a slot machine. The straightforward analogy is to imagine different web site configurations as a row of slot machines, each with its own probability of producing a reward (i.e. a conversion). The multi-armed bandit problem is notoriously resistant to analysis [2], although optimal solutions are available in certain special cases [3, 4]. The optimal solutions are typically hard to compute, rely on artificial discount factors, and fail to generalize to realistic reward distributions. They can also exhibit *incomplete learning*, meaning that there is a positive probability of playing the wrong arm forever [5].

Because of the drawbacks associated with optimal solutions, analysts often turn to heuristics to manage the exploration/exploitation trade-off. Randomized probability matching is a particularly appealing heuristic that plays each arm in proportion to its probability of being optimal. Randomized probability matching is easy to implement, broadly applicable, and combines several attractive features of other popular heuristics. It is an old idea [6, 7], but modern Bayesian computation has dramatically broadened the class of reward distributions to which it can be applied.

The simplicity of randomized probability matching allows the multi-armed bandit to incorporate powerful ideas from classical design. For example, both bandits and classical experiments must face the exploding number of possible configurations as factors are added to the experiment. Classical experiments handle this problem using fractional factorial designs (see, e.g. [8]), which surrender the ability to fit certain complex interactions in order to reduce the number of experimental runs. These designs combat the curse of dimensionality by indirectly learning about an arm's reward distribution by examining rewards from other arms with similar characteristics. Bandits can use the fractional factorial idea by assuming that a model, such as a probit or logistic regression, determines the reward distributions of the different arms. Assuming a parametric model allows the bandit to focus on a lower-dimensional parameter space and thus potentially achieve greater rewards than 'optimal' solutions that make no parametric assumptions.

It is worth noting that there are also important differences between classical experiments and bandits. For example, the traditional optimality criteria from classical experiments (D-optimality, A-optimality, etc.) tend to produce balanced experiments where all treatment effects can be accurately estimated. In a multi-armed bandit, it is actually undesirable to accurately estimate treatment effects for inferior arms. Instead, a bandit aims to gather just enough information about a sub-optimal arm

DOI: 10.1002/asmb

to determine that it is sub-optimal, at which point further exploration becomes wasteful. A second difference is the importance placed on statistical significance. Classical experiments are designed to be analyzed using methods that tightly control the type-I error rate under a null hypothesis of no effect. But when the cost of switching between products is small (as with software testing), the type-I error rate is of little relevance to the bandit. A type-I error corresponds to switching to a different arm that provides no material advantage over the current arm. By contrast, a type-II error means failing to switch to a superior arm, which could carry a substantial cost. Thus when switching costs are small almost all the costs lie in type-II errors, which makes the usual notion of statistical significance largely irrelevant. Finally, classical experiments typically focus on designs for linear models because the information matrix in a linear model is a function of the design matrix. Designs for nonlinear models, such as probit or logistic regression, are complicated by the fact that the information matrix depends on unknown parameters [9]. This complication presents no particular difficulty to the multi-armed bandit played under randomized probability matching.

The remainder of this paper is structured as follows. Section 2 describes the principle of randomized probability matching in greater detail. Section 3 reviews other approaches for multi-armed bandits, including the Gittins index and several popular heuristics. Section 4 presents a simulation study that investigates the performance of randomized probability matching in the unstructured binomial bandit, where optimal solutions are available. Section 5 describes a second simulation study in which the reward distribution has low-dimensional structure, where 'optimal' methods do poorly. There is an important symmetry between Sections 4 and 5. Section 4 illustrates the cost savings that sequential learning can have over classic experiments. Section 5 illustrates the improvements that can be brought to sequential learning by incorporating classical ideas like fractional factorial design. Section 6 concludes with observations about extending multi-armed bandits to more elaborate settings.

2. RANDOMIZED PROBABILITY MATCHING

Let $\mathbf{y}_t = (y_1, \ldots, y_t)$ denote the sequence of rewards observed up to time t. Let a_t denote the arm of the bandit that was played at time t. Suppose that each y_t was generated independently from the reward distribution $f_{a_t}(y|\theta)$, where θ is an unknown parameter vector, and some components of θ may be shared across the different arms.

To make the notation concrete, consider two specific examples, both of which take $y_t \in \{0, 1\}$. Continuous rewards are also possible, of course, but I will focus on binary rewards because counts of clicks or conversions are the typical measure of success in e-commerce. The first example is the *binomial bandit*, in which $\theta = (\theta_1, \ldots, \theta_k)$, and $f_a(y_t|\theta)$ is the Bernoulli distribution with success probability θ_a. The binomial bandit is the canonical bandit problem appearing most often in the literature. The second example is the *fractional factorial bandit*, where a_t corresponds to a set of levels for a group of experimental factors (including potential interactions), coded as dummy variables in the vector $\mathbf{x}_t$. Let k denote the number of possible configurations of $\mathbf{x}_t$, and let $a_t \in \{1, \ldots, k\}$ refer to a particular configuration according to some labeling scheme. The probability of success is $f_a(y_t = 1|\theta) = g(\theta^{\mathrm{T}}\mathbf{x}_t)$, where g is a binomial link function, such as probit or logistic. I refer to the case where g is the CDF of the standard normal distribution as the *probit bandit*.

Let $\mu_a(\theta) = E(y_t|\theta, a_t = a)$ denote the expected reward from $f_a(y|\theta)$. If θ were known, then the optimal long run strategy would be to always choose the arm with the largest $\mu_a(\theta)$. Let $p(\theta)$

DOI: 10.1002/asmb

denote a prior distribution on θ, from which one may compute

$$w_{a0} = \Pr(\mu_a = \max\{\mu_1, \ldots, \mu_k\}). \tag{1}$$

The computation in Equation (1) can be expressed as an integral of an indicator function. Let $I_a(\theta) = 1$ if $\mu_a(\theta) = \max\{\mu_1(\theta), \ldots, \mu_k(\theta)\}$, and $I_a(\theta) = 0$ otherwise. Then

$$w_{a0} = E(I_a(\theta)) = \int I_a(\theta)\, p(\theta)\, \mathrm{d}\theta. \tag{2}$$

If *a priori* little is known about θ then the implied distribution on μ will be exchangeable, and thus w_{a0} will be uniform. As rewards from the bandit process are observed, the parameters of the reward distribution are learned through Bayesian updating. At time t, the posterior distribution of θ is

$$p(\theta|\mathbf{y}_t) \propto p(\theta) \prod_{\tau=1}^{t} f_{a_\tau}(y_\tau|\theta), \tag{3}$$

from which one may compute

$$\begin{aligned} w_{at} &= \Pr(\mu_a = \max\{\mu_1, \ldots, \mu_k\}|\mathbf{y}_t) \\ &= E(I_a(\theta)|\mathbf{y}_t), \end{aligned} \tag{4}$$

as in Equation (2).

Randomized probability matching allocates observation $t+1$ to arm a with probability w_{at}. Randomized probability matching is not known to optimize any specific utility function, but it is easy to apply in general settings, balances exploration and exploitation in a natural way, and tends to allocate observations efficiently from both inferential and economic perspectives. It is compatible with batch updates of the posterior distribution, and the methods used to compute the allocation probabilities can also yield useful quantities for reporting experimental results.

A particularly convenient feature of randomized probability matching is that it is free of arbitrary tuning parameters that must be set by the analyst. Tuning parameters can be introduced, if desired, by allocating observations in proportion to some monotonic function of w_{at}. For example, one can influence the amount of exploration by allocating observations with probability proportional to w_{at}^{γ}, for some $\gamma > 0$. The algorithm will be more cautious (tend to explore) with $\gamma < 1$, and it will be more aggressive if $\gamma > 1$. The remainder of the paper considers the obvious default case of $\gamma = 1$.

2.1. Computing allocation probabilities

For some families of reward distributions, it is possible to compute w_{at} either analytically or by quadrature. In any case, it is easy to compute w_{at} by simulation. Let $\theta^{(1)}, \ldots, \theta^{(G)}$ be a sample of independent draws from $p(\theta|\mathbf{y}_t)$. Then by the law of large numbers,

$$w_{at} = \lim_{G\to\infty} \frac{1}{G} \sum_{g=1}^{G} I_a(\theta^{(g)}). \tag{5}$$

Equation (5) simply says to estimate w_{at} by the empirical proportion of Monte Carlo samples in which $\mu_a(\theta^{(g)})$ is maximal. If f_a is in the exponential family and $p(\theta)$ is a conjugate prior distribution, then independent draws of θ are possible. Otherwise one may draw a sequence

DOI: 10.1002/asmb

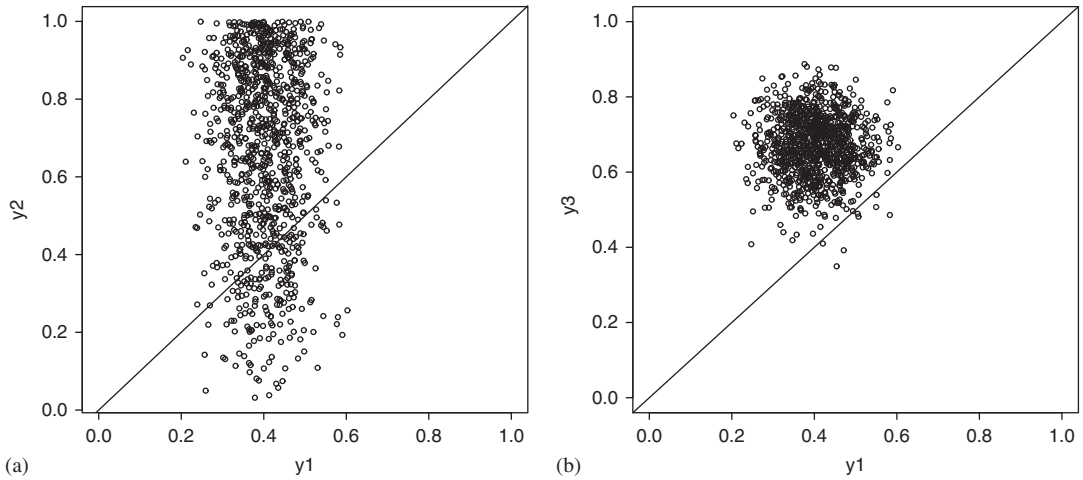


Figure 1. One thousand draws from the joint distribution of two independent beta distributions. In both cases, the horizontal axis represents a beta (20,30) distribution. The vertical axis is (a) beta(2,1) and (b) beta(20,10).

$\theta^{(1)}, \theta^{(2)}, \ldots$ from an ergodic Markov chain with $p(\theta|\mathbf{y}_t)$ as its stationary distribution [10]. In the latter case Equation (5) remains unchanged, but it is justified by the ergodic theorem rather than the law of large numbers. The method used to generate the posterior sample is not particularly important. For example, sequential Monte Carlo [11] could be used in place of Markov chain Monte Carlo, and may in fact be more natural.

Posterior draws of θ are all that is needed to apply randomized probability matching. Such draws are available for a very wide class of models through Markov chain Monte Carlo and other sampling algorithms, which means that randomized probability matching can be applied with almost any family of reward distributions.

2.2. *Implicit allocation*

The optimality probabilities in Equation (5) do not need to be computed explicitly. It will usually be faster to simulate $a \sim w_{at}$ by simulating a single draw of $\theta^{(t)}$ from $p(\theta|\mathbf{y}_t)$ and setting $a = \arg\max_a \mu_a(\theta^{(t)})$.

2.3. *Exploration, exploitation, and uncertainty*

Randomized probability matching naturally incorporates uncertainty about θ because w_{at} is defined as an integral over the entire posterior distribution $p(\theta|\mathbf{y}_t)$. To illustrate, consider the binomial bandit with $k = 2$ under independent beta priors. Figure 1(a) plots $p(\theta_1, \theta_2|\mathbf{y})$ assuming that we have observed 20 successes and 30 failures from the first arm, along with 2 success and 1 failure from the second arm. In panel (b), the second arm is replaced with an arm that has generated 20 successes and 10 failures. Thus it has the same empirical success rate as the second arm in panel (a), but with a larger sample size. In both plots, the optimality probability for the first arm is the probability a dot lands below the 45° line (which Equation (5) estimates by counting simulated dots). In panel (a), the probability that the first arm is optimal is around 18%, despite having a

DOI: 10.1002/asmb

lower empirical success rate than the second arm. In panel (b), the larger sample size causes the posterior distribution to tighten, which lowers the first arm's optimality probability to 0.8%.

This example shows how the amount of exploration decreases as more is learned about the parameters of the reward distribution. If the two largest values of $\mu_a(\theta)$ are distinct, then $\max_a w_{at}$ eventually converges to 1. If the $k_1 \leqslant k$ largest values of $\mu_a(\theta)$ are identical, then $\max_a w_{at}$ need not converge, but w_{at} may drift on a subset of the probability simplex that divides 100% of probability among the k_1 optimal alternatives. This is obviously just as good as convergence from the perspective of total reward accumulation.

Notice the alignment between the inferential goal of finding the optimal arm and the economic goal of accumulating reward. From both perspectives, it is desirable for superior arms to rapidly accumulate observations. Downweighting inferior arms leads to larger economic rewards, while larger sample sizes for superior arms means that the optimal arm can be more quickly distinguished from its close competitors.

3. REVIEW OF OTHER METHODS

This section reviews a collection of strategies that have been used with multi-armed bandit problems. I discuss pure exploration strategies, purely greedy strategies, hybrid strategies, methods based on upper confidence bounds (UCBs), and Gittins indices. Of these, only Gittins indices carry an explicit optimality guarantee, and then only under a very particular scheme of discounting future rewards. Some of the other methods can be shown to converge to asymptotically optimal behavior.

3.1. The Gittins index

Gittins [3] provided a method of computing the optimal strategy in certain bandit problems. The method assumes a geometrically discounted stream of future rewards with present value $\mathrm{PV} = \sum_{t=0}^{\infty} \gamma^t y_t$, for some $0 \leqslant \gamma < 1$. Gittins provided an algorithm for computing the expected discounted present value of playing arm a, assuming optimal play in the future, a quantity that has since become known as the 'Gittins index.' Thus, by definition, playing the arm with the largest Gittins index maximizes the expected present value of discounted future rewards. The Gittins index has the further remarkable property that it can be computed separately for each arm, in ignorance of the other arms. A policy with this property is known as an *index policy*.

Logical and computational difficulties have prevented the widespread adoption of Gittins indices. Powell [12] notes that 'Unfortunately, at the time of this writing, there do not exist easy to use software utilities for computing standard Gittins indices'. Sutton and Barto [13] add 'Unfortunately, neither the theory nor the computational tractability of [Gittins indices] appear to generalize to the full reinforcement learning problem ...'.

Although it is hard to compute Gittins indices exactly, Brezzi and Lai [14] have developed an approximate Gittins index based on a normal approximation to $p(\theta|\mathbf{y})$. Figure 2 plots the approximation for the binomial bandit with two different values of γ. Both sets of indices converge to $a/(a+b)$ as a and b grow large, but the rate of convergence slows as $\gamma \to 1$. The Brezzi and Lai approximation to the Gittins index is as follows. Let $\theta_n = E(\theta|\mathbf{y}_n, \mathbf{a}_n)$, $v_{an} = \mathrm{Var}(\mu_a|\mathbf{y}_n)$, $\sigma_a^2(\theta) = \mathrm{Var}(y_t|\theta, a_t = a)$, and $c = -\log\gamma$. Then

$$\tilde{v}_a = \mu_a(\theta_n) + v_{an}^{1/2}\psi\left(\frac{v_{an}}{c\sigma^2(\theta_n)}\right), \tag{6}$$

DOI: 10.1002/asmb

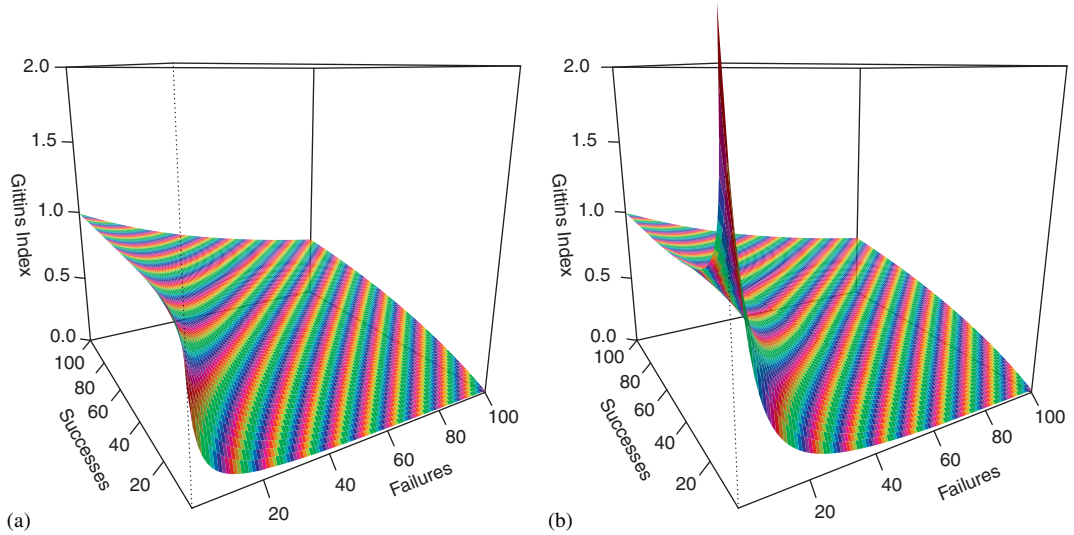


Figure 2. Brezzi and Lai's approximation to the Gittins index for the binomial bandit problem with (a) $\gamma=0.8$ and (b) $\gamma=0.999$.

where

$$\psi(s)=\begin{cases}\sqrt{s/2} & \text{if } s\leqslant 0.2,\\ 0.49-0.11s^{-1/2} & \text{if } 0.1<s\leqslant 1,\\ 0.63-0.26s^{-1/2} & \text{if } 1<s\leqslant 5,\\ 0.77-0.58s^{-1/2} & \text{if } 5<s\leqslant 15,\\ (2\log s-\log\log s-\log 16\pi)^{1/2} & \text{if } s>15.\end{cases} \tag{7}$$

Computing aside, there are three logical issues that challenge the Gittins index (and the broader class of index polices). The most important is known as *incomplete learning*, which means that the Gittins index is an inconsistent estimator of the location of the optimal arm. The Gittins policy eventually chooses one arm on which to continue forever, and there is a positive probability that the chosen arm is sub-optimal [5].

A second issue (which applies to any index policy) is that the arms must have distinct parameters. The Gittins index can be far from optimal when applied to problems where the arms are described by covariates or structured experimental factors. It is useful only for sequential versions of the 'one-way layout.' The final issue is that if the discounting scheme is anything other than geometric, then the Gittins index is no longer optimal [15, 16]. In real applications, arms are rarely played at equally spaced time intervals as would be necessary for true geometric discounting. Thus ,it should be possible to find reasonable competitors to the Gittins index in applied problems.

3.2. UCB algorithms

Lai and Robbins [17] and Lai [18] introduced a class of algorithms based on UCBs for the mean reward of each arm. Each arm is assigned an UCB for its mean reward, and the arm with the

DOI: 10.1002/asmb

largest bound is to be played. The bound is not the conventional upper limit for a confidence interval, and is difficult to compute. However under some basic assumptions described below, the expected number of times suboptimal arm a would be played by time t is

$$E(n_{at}) \leqslant \left(\frac{1}{K(a, a^*)} + o(1) \right) \log t, \tag{8}$$

where $K(a, a^*)$ is the Kullback–Leibler divergence between the reward distributions for arm a and the optimal arm a^*. This bound essentially says that the optimal arm will be played exponentially more often than any of the suboptimal arms, for large t. Lai and Robbins showed that this bound is asymptotically optimal as $t \to \infty$. Agrawal [19] and Auer *et al.* [20] suggested more computationally tractable versions of the UCB algorithm with regret bounds that are also of order $O(\log t)$, with proportionality constants close to the Lai and Robbins bound.

Lai [18] assumes that the arm-specific reward distributions are single parameter exponential family models (in which case the mean is a sufficient statistic). Likewise, the UCB algorithms introduced by Agrawal and Auer *et al.* only require knowledge of the sample mean of each arm (and the sample variance, for normally distributed rewards). Furthermore, the regret bounds that Auer *et al.* [20] obtain hold even if the reward distributions are dependent (as they would be if rewards were determined by a regression model, for example). However the optimality of Equation (8) assumes independent error distributions, which means that lower-regret solutions might be obtained in shared-parameter settings. Thus, UCB algorithms mentioned above might be far from optimal in the presence of dependent rewards. UCB algorithms are an active research area in computer science, and it is likely that effective UCB algorithms can be derived for more general payoff distributions.

3.3. Heuristic strategies

3.3.1. Equal allocation. One naive method of playing a multi-armed bandit is to equally allocate observations to arms until the maximum optimality probability exceeds some threshold, and then play the winning arm afterward. This strategy leads to stable estimates of θ for all the arms, but Section 4 demonstrates that it is grossly inefficient with respect to the overall reward. Of the methods considered here, equal allocation most closely corresponds to a non-sequential classical experiment (the one-way layout).

3.3.2. Play-the-winner. Play-the-winner [21] is a well-known strategy in which arm a is played at time $t+1$ if it resulted in a success at time t. If a failure is observed at time t, then the next arm is either chosen at random or the arms are cycled through deterministically. Play-the-winner can be nearly optimal when the best arm has a very high success rate. Berry and Fristedt [16] show that if arm a is optimal at time t, and it results in a success, then it is also optimal to play arm a at time $t+1$. However the converse is not true: it is not necessarily desirable to switch away from an arm in the event of a failure. It is straightforward to see that play-the-winner tends toward equal allocation, and thus over-explores, as the success rate of the optimal arm tends to zero.

3.3.3. Deterministic greedy strategies. An algorithm that focuses purely on exploitation is said to be *greedy*. A textbook example of a greedy algorithm is to always choose the arm with the highest sample mean reward. This approach has been shown to do a poor job of producing large overall rewards because it fails to adequately explore the other arms [13]. A somewhat better greedy

DOI: 10.1002/asmb

algorithm is deterministic probability matching, which always chooses the arm with the highest optimality probability w_{at}.

One practicality to keep in mind with greedy strategies is that they can do a poor job when batch updating is employed. For logistical reasons, the bandit may have to be played multiple times before it can learn from recent activity. This can occur if data arrive in batches instead of in real time (e.g. computer logs might be scraped once per hour or once per day). Greedy algorithms can suffer in batch updating because they play the same arm for an entire update cycle. This can add substantial variance to the total reward relative to randomized probability matching. Greedy strategies perform especially poorly in the early phases of batch updating because they only learn about one arm per update cycle.

3.3.4. Hybrid strategies. Hybrid strategies are greedy strategies that have been modified to force some amount of exploration. One example is an ε-greedy strategy. Here each allocation takes place according to a greedy algorithm with probability $1-\varepsilon$, otherwise a random allocation takes place. The equal allocation strategy is an ε-greedy algorithm with $\varepsilon=1$. One can criticize an ε-greedy strategy on the grounds that it has poor asymptotic behavior, because it continues to explore long after the optimal solution becomes apparent. This leads to the notion of an ε-decreasing strategy, which is an ε-greedy strategy where ε decreases over time. The performance of ε-decreasing strategies can be quite good. Auer *et al.* [20] showed that, for a particular decay schedule, the ε-decreasing algorithm selects a sub-optimal arm on play t with probability $O(1/t)$ as $t\to\infty$. The total regret after t plays is thus near the optimal bound of $O(\log t)$.

Both ε-greedy and ε-decreasing strategies are wasteful in the sense that they use simple random sampling as the basis for exploration. A more fruitful approach would be to use stratified sampling that under-samples arms that are likely to be sub-optimal. Softmax learning [22] is a randomized strategy that allocates observation $t+1$ to arm a with probability

$$\tilde{w}_{at}=\frac{\exp(\mu_{at}/\tau)}{\sum_{j=1}^{k}\exp(\mu_{jt}/\tau)}, \tag{9}$$

where τ is a tuning parameter to be chosen experimentally. Softmax learning with fixed τ shares the same asymptotic inefficiency as ε-greedy strategies, which can be eliminated by gradually decreasing τ to zero.

3.4. Comparison with randomized probability matching

Randomized probability matching combines many positive aspects of the strategies mentioned in Section 3.3. It is ε-greedy in the sense that it employs deterministic probability matching with probability $\max_a w_{at}$, and a random (though non-uniform) exploration with probability $\varepsilon=1-\max_a w_{at}$. It is ε-decreasing in the sense that in non-degenerate cases $\max_a w_{at}\to 1$. However, it should be noted that $\max_a w_{at}$ can sometimes decrease in the short run if the data warrant. The regret bounds for ε-decreasing algorithms proved by Auer *et al.* [20] offer hope that randomized probability matching can do even better, but I know of no theorem to that effect. The stratified exploration provided by softmax learning matches that used by randomized probability matching to the extent that w_{at} is determined by μ_{at} through a multinomial logistic regression with coefficient $1/\tau$. The benefit of randomized probability matching is that the tuning parameters and their decay schedules evolve in principled, data-determined ways rather than being arbitrarily set by the analyst.

DOI: 10.1002/asmb

The cost is the need to sample from the posterior distribution, which can require substantially more computing than the other heuristics.

Randomization is an important component of a bandit allocation strategy because deterministic strategies can fail to consistently estimate which arm is optimal. Yang and Zhu [23] proved that an ε-decreasing randomization can restore consistency to a greedy policy. I conjecture that randomized probability matching does consistently estimate the optimal arm. If a^* denotes the index of the optimal arm, then a sufficient condition for the conjecture to be true is for $w_{a^*t} > \varepsilon$ for some $\varepsilon > 0$, where ε is independent of t.

4. THE BINOMIAL BANDIT

This section describes a collection of simulation studies comparing randomized probability matching to various other learning algorithms in the context of the binomial bandit, where there are ample competitors. The binomial bandit assumes that the rewards in each configuration are independent Bernoulli random variables with success probabilities $(\theta_1, \ldots, \theta_k)$. For simplicity, assume the uniform prior distribution $\theta_a \sim U(0,1)$, independently across a. Let Y_{at} and N_{at} denote the cumulative number of successes and trials observed for arm a up to time t. Then the posterior distribution of $\theta = (\theta_1, \ldots, \theta_k)$ is

$$p(\theta|\mathbf{y}_t) = \prod_{a=1}^{k} \mathrm{Be}(\theta_a | Y_{at}+1, N_{at}-Y_{at}+1), \tag{10}$$

where $\mathrm{Be}(\theta|\alpha,\beta)$ denotes the density of the beta distribution for random variable θ with parameters α and β. The optimality probability

$$w_{at} = \int_0^1 \mathrm{Be}(\theta_a | Y_{at}+1, N_{at}-Y_{at}+1) \prod_{j \neq a} \Pr(\theta_j < \theta_a | Y_{jt}+1, N_{jt}-Y_{jt}+1)\, \mathrm{d}\theta_a \tag{11}$$

can easily be computed either by quadrature or simulation (see Figures 3 and 4).

Our simulation studies focus on *regret*, the cumulative expected lost reward, relative to playing the optimal arm from the beginning of the experiment. Let $\mu^*(\theta) = \max_a\{\mu_a(\theta)\}$, the expected

```
compute.probopt <- function(y, n){
  k <- length(y)
  ans <- numeric(k)
  for(i in 1:k){
    indx <- (1:k)[-i]
    f <- function(x){
      r <- dbeta(x, y[i]+1, n[i]-y[i]+1)
      for(j in indx)
        r <- r * pbeta(x, y[j]+1, n[j]-y[j]+1)
      return(r) }
    ans[i] = integrate(f,0,1)$value }
  return(ans) }
```

Figure 3. R code for computing Equation (11) by quadrature.

DOI: 10.1002/asmb

```
sim.post <- function(y, n, ndraws){
  k <- length(y)
  ans <- matrix(nrow=ndraws, ncol=k)
  no <- n-y
  for(i in 1:k)
    ans[,i]<-rbeta(ndraws,y[i]+1,no[i]+1)
  return(ans)}
prob.winner <- function(post){
  k <- ncol(y)
  w <- table(factor(max.col(post), levels=1:k))
  return(w/sum(w))}
compute.win.prob <- function(y, n, ndraws){
  return(prob.winner(sim.post(y, n, ndraws)))}
```

Figure 4. R code for computing Equation (11) by simulation.

reward under the truly optimal arm, and let n_{at} denote the number of observations that were allocated to arm a at time t. Then the expected regret at time t is

$$L_t = \sum_a n_{at}(\mu^*(\theta) - \mu_a(\theta)), \tag{12}$$

and the cumulative regret at time T is $L = \sum_{t=1}^{T} L_t$. The units of L_t are the units of reward; hence, with $\frac{0}{1}$ rewards L_t is the expected number of lost successes relative to the unknown optimal strategy.

The simulation study consisted of 100 experiments, each with $k = 10$ 'true' values of θ independently generated from the $\mathcal{N}(0, \frac{1}{10})$ distribution. The first such value was assumed to be the current configuration and was assigned 10^6 prior observations, so that it was effectively a known 'champion' with the remaining nine arms being new 'challengers'. I consider both batch and real-time updates.

4.1. Batch updating

In this study, each update contains a Poisson(1000) number of observations allocated to the different arms based on one of several allocation schemes. I ran each experiment until either the maximal w_{at} exceeded a threshold of 0.95, or else 100 time periods had elapsed.

4.1.1. RPM vs equal allocation. Our first example highlights the substantial gains that can be had from sequential learning relative to a classical non-sequential experiment. Figure 5 compares the cumulative regret for the randomized probability matching and equal allocation strategies across 100 simulation experiments. The regret from the equal allocation strategy is more than an order of magnitude greater than under probability matching. Figure 6 shows why it is so large. Each boxplot in Figure 6 represents the distribution of L_t for experiments that were still active at time t. The regret distribution at time 1 is the same in both panels, but the expected regret under randomized probability matching quickly falls to zero as sub-optimal arms are identified and excluded from further study. The equal allocation strategy continues allocating observations to sub-optimal arms for the duration of the experiment. This produces more accurate estimates of θ for the sub-optimal arms but at great expense.

The inferential consequences of the multi-armed bandit can be seen in Figure 7, which shows how the posterior distribution of θ_a evolves over time for a single experiment. Notice how the

DOI: 10.1002/asmb

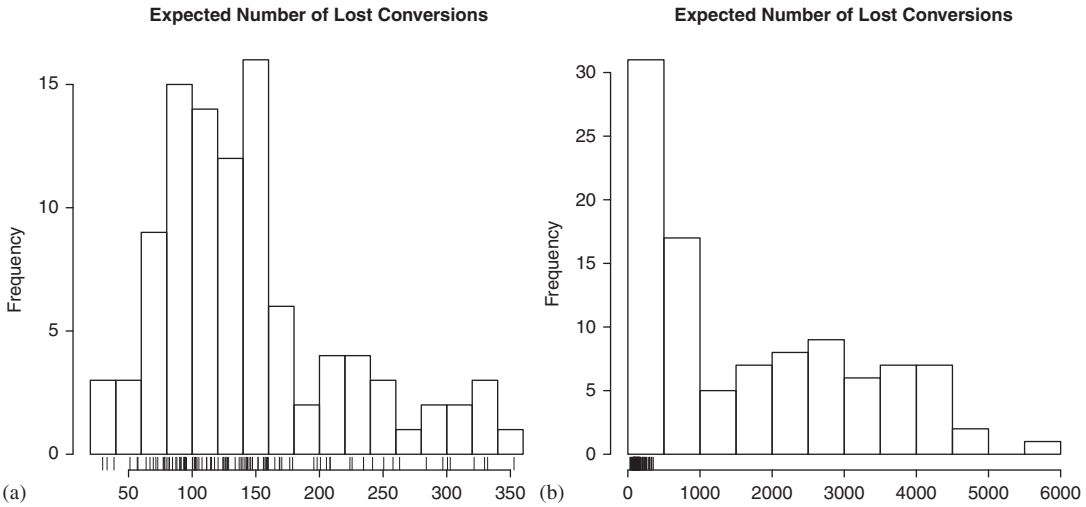


Figure 5. Cumulative regret in each of 100 simulation experiments under (a) randomized probability matching and (b) equal allocation. Note that the scales differ by an order of magnitude. For comparison purposes, both panels include a rug-plot showing the regret distribution under probability matching.

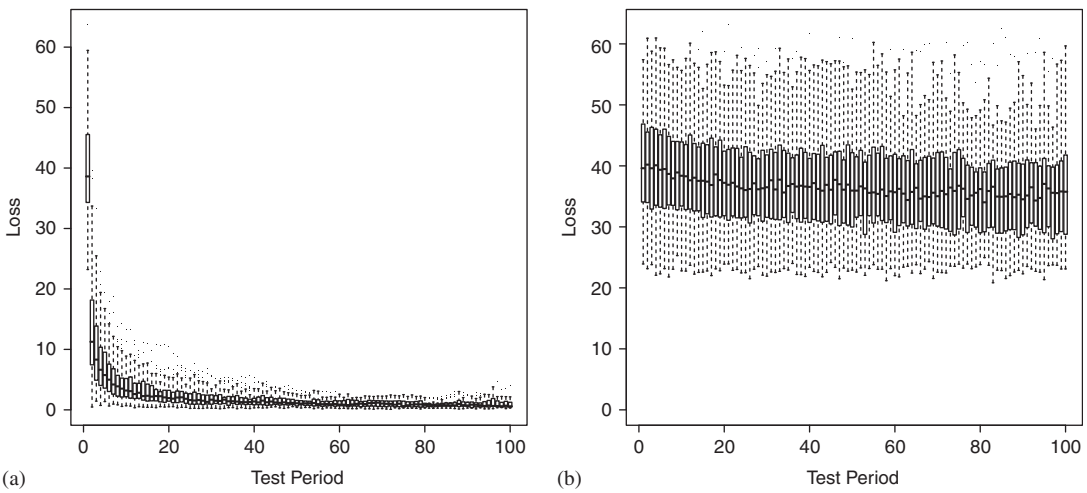


Figure 6. Expected regret per time period under (a) randomized probability matching and (b) equal allocation. Boxplots show variation across 100 simulated experiments.

95% credibility interval for the optimal arm (in the upper left plot of each panel) is smaller in panel (a) than in panel (b), despite the fact that the experiment in panel (b) ran longer. Likewise, notice that the 95% credible intervals for the sub-optimal arms are much wider under probability matching than under equal allocation.

4.1.2. RPM vs greedy algorithms. The experiment described above was also run under the purely greedy strategies of deterministic probability matching and playing the arm with the largest mean.

DOI: 10.1002/asmb

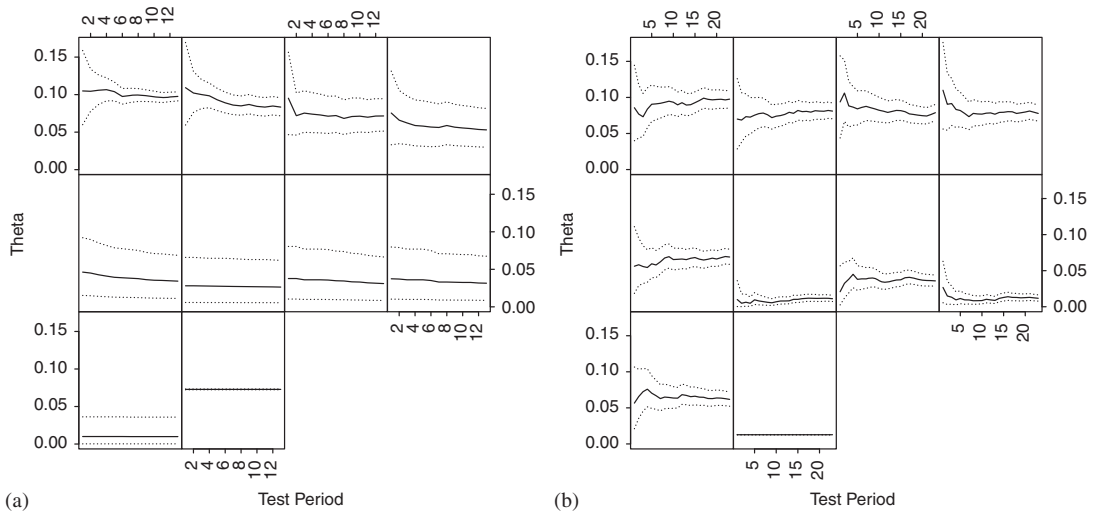


Figure 7. Evolution of the posterior distribution (means and upper and lower 95% credibility bounds) of θ under (a) randomized probability matching and (b) equal allocation. Each panel corresponds to an arm. Arms are sorted according to optimality probability when the experiment ends.

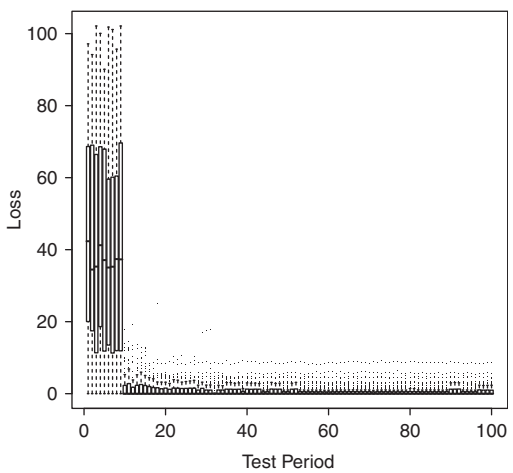
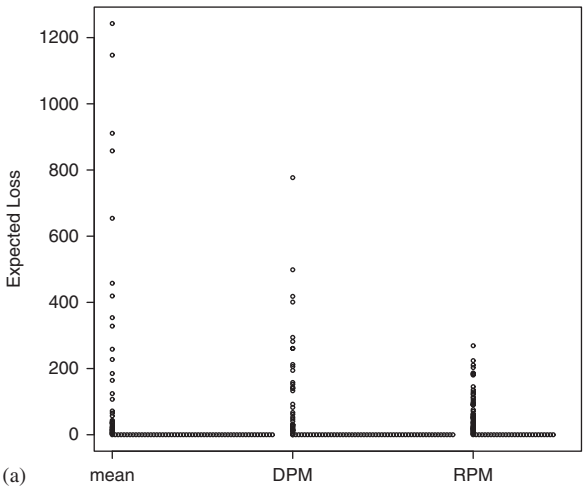


Figure 8. Regret from deterministic probability matching. The first ten periods were spent learning the parameters of each arm.

Neither purely greedy strategy is suitable for the batch updating used in our experiment. Because a $\mathcal{U}(0, 1)$ prior was used for each θ_a and because the 'true' values were simulated from $\mathcal{U}(0, \frac{1}{10})$, an arm with no observations will have a higher mean and higher optimality probability than an arm with observed rewards. Thus a purely greedy strategy will spend the first k batches cycling through the arms, assigning all the observations to each one in turn (see Figure 8). This form of exploration has the same expected regret as equal allocation, but with a larger variance, because all bets are placed on a single arm.

DOI: 10.1002/asmb

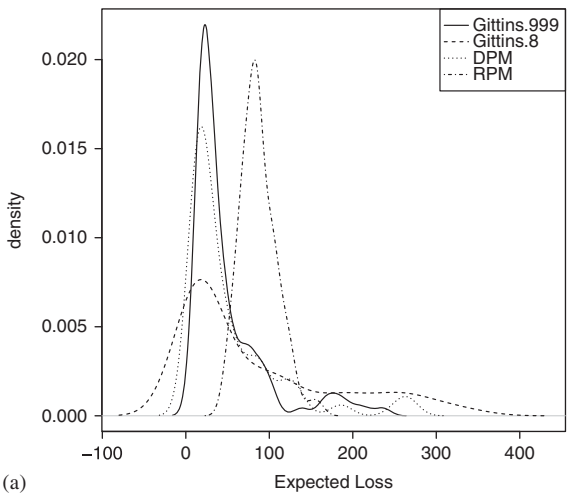


(a)

Method	Mean	SD
RPM	50.1	62.2
DPM	54.7	122.3
Largest Sample Mean	81.4	225.7

(b)

Figure 9. (a) Stacked stripchart showing cumulative regret after excluding the first 10 test periods. Panel (b) shows means and standard deviations of the expected losses plotted in panel (a). Greedy methods have a higher chance of zero regret. Randomized probability matching has a lower variance.



(a)

Method	Mean	SD	%-Correct
Gittins(.999)	49.0	47.0	63
Gittins(.8)	84.0	94.2	48
DPM	51.9	58.4	58
RPM	87.3	21.7	76

(b)

Figure 10. (a) Expected regret under real-time sampling across 100 experiments, each simulated for 10 000 time steps and (b) mean and standard deviation of the expected losses plotted in panel (a), along with the percentage of experiments for which the optimal arm was selected at time 10 000.

Figure 9 compares the cumulative expected regret for the two greedy strategies to random probability matching after excluding the first 10 time steps, after the greedy algorithms have explored all 10 arms. The reward under randomized probability matching has a much lower standard deviation than under either of the greedy strategies. It also has the lowest sample mean, although the difference between its mean and that of deterministic probability matching is not statistically significant. Notice that the frequency of exact zero regret is lowest for randomized probability matching, but its positive losses are less than those suffered by the greedy methods.

DOI: 10.1002/asmb

4.2. Real-time updating

A third simulation study pitted randomized probability matching against the Gittins index, in the setting where Gittins is optimal. This time the experiment was run for 10 000 time steps, each with a single play of the bandit. Again the simulation uses $k=10$ and independently draws the true success probabilities from $\mathscr{U}(0, \frac{1}{10})$. Figure 10 compares randomized and deterministic probability matching with Gittins index strategies where $\gamma=0.999$ and $\gamma=0.8$. Of the four methods, randomized probability matching did the worst job of accumulating total reward, but it had the smallest standard deviation, and it selected the optimal arm at the end of the experiment the largest number of times. The Gittins index with $\gamma=0.999$ gathered the largest reward, but with a larger standard deviation and lower probability of selecting the optimal arm. Deterministic probability matching did slightly worse than the better Gittins policy on all three metrics. Finally, the Gittins index with $\gamma=0.8$ shows a much thicker tail than the other methods, illustrating the fact that you can lower your overall total reward by too heavily discounting the future.

5. FRACTIONAL FACTORIAL BANDIT

The binomial bandit described in Section 4 fails to take advantage of potential structure in the arms. For example, suppose the arms are web sites differing in font family, font size, image location, and background color. If each characteristic has five levels, then there are $5^4=625$ possible configurations to test. One could analyze this problem using an unstructured binomial bandit with 625 arms, but the size of the parameter estimation problem can be dramatically reduced by assuming additive structure. In the preceding example, suppose each 5-level factor is represented by four indicator variables in a probit or logistic regression. If one includes an intercept term and assumes a strictly additive structure, then there are only $1+(5-1)\times 4=17$ parameters that need estimating. Interaction terms can be included if the additivity assumption is too restrictive. Choosing a set of interactions to allow into the bandit is analogous to choosing a particular fractional factorial design in a classical problem.

Let $\mathbf{x}_t$ denote the vector of indicator variables describing the characteristics of the arm played at time t. For the purpose of this Section I will assume that the probability of a reward depends on $\mathbf{x}_t$ through a probit regression model, but any other model can be substituted as long as posterior draws of the model's parameters can be easily obtained. Let $\Phi(z)$ denote the standard normal cumulative distribution function. The probit regression model assumes

$$\Pr(y_t=1)=\Phi(\theta^{\mathrm{T}}\mathbf{x}_t). \tag{13}$$

The probit regression model has no conjugate prior distribution, but a well-known data augmentation algorithm [24] can be used to produce serially correlated draws from $p(\theta|\mathbf{y})$. The algorithm is described in Section A.1 of the Appendix. Each iteration of Albert and Chib's algorithm requires a latent variable to be imputed for each y_t. This can cause the posterior sampling algorithm to slow as more observations are observed. However, if $\mathbf{x}$ is small enough to permit all of its possible configurations to be enumerated then Albert and Chib's algorithm can be optimized to run much faster as $t\to\infty$. Section A.2 of the Appendix explains the modified algorithm. Other modifications based on large sample theory are possible.

A simulation study was conducted to compare the effectiveness of the fractional factorial and binomial bandits. In the simulation, data were drawn from a probit regression model based on

DOI: 10.1002/asmb

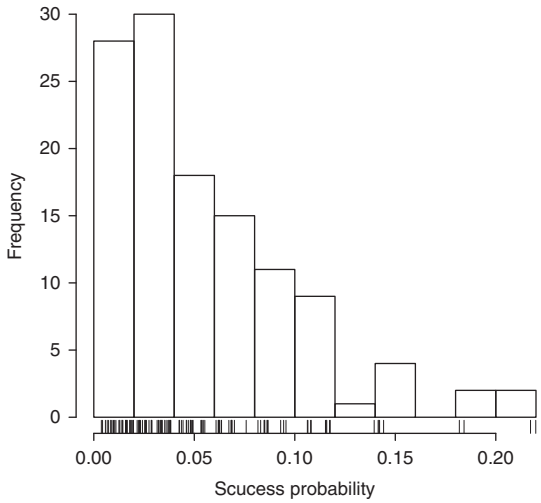


Figure 11. True success probabilities on each arm of the fractional factorial bandit.

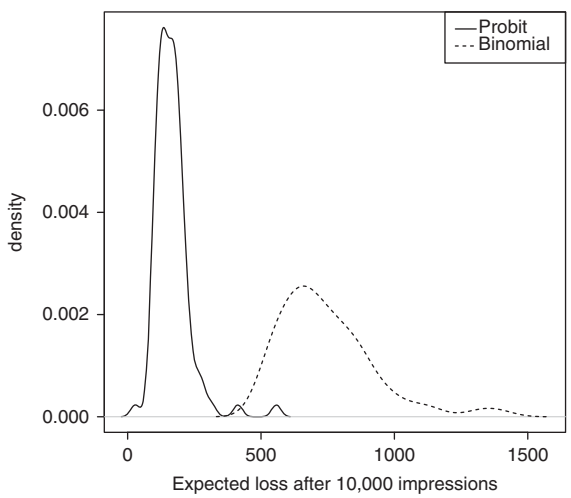


Figure 12. Cumulative regret after 10,000 trials for the fractional factorial (solid) and binomial (dashed) bandits.

four discrete factors with two, three, four, and five levels. This model has 120 possible configurations, but only 11 parameters, including an intercept term. The intercept term was drawn from a normal distribution with mean $\Phi^{-1}(0.05)$ and variance 0.1. The other coefficients were simulated independently from the $\mathcal{N}(0, 0.1)$ distribution. These levels were chosen to produce arms with a mean success probability of around 0.05, and a standard deviation of about 0.5 on the probit scale. Figure 11 shows the 120 simulated success probabilities for one of the simulated experiments. The simulation was replicated 100 times, producing 100 bandit processes on which to experiment.

DOI: 10.1002/asmb

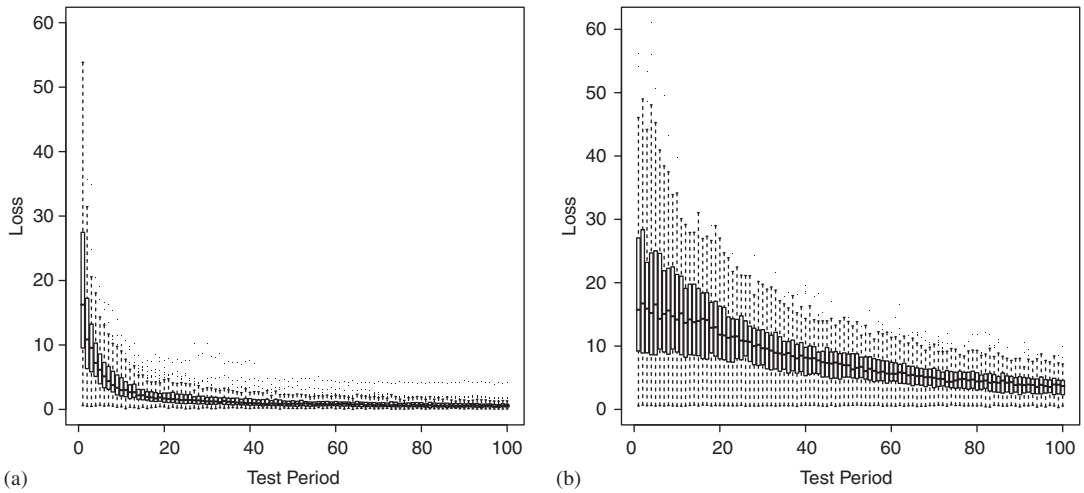


Figure 13. Regret distributions for (a) the fractional factorial bandit and (b) the binomial bandit, under randomized probability matching, when the underlying process has probit structure.

Let $\{\mathbf{x}_a : a = 1, \ldots, 120\}$ denote the possible configurations of $\mathbf{x}$. In each update period, 100 observations were randomly allocated to the different possible configurations according to w_{at}. At each time step 1000 draws were produced from $p(\theta|\mathbf{y}_t)$ using the algorithm in Section A.2, assuming independent $\mathcal{N}(0, 1)$ priors for all coefficients. These draws were used to compute w_{at} as described in Section 2.1, with $\mu_a(\theta) = \Phi(\theta^{\mathrm{T}}\mathbf{x}_a)$. Note that Φ is a monotonic function; hence, the arm with the largest $\mu_a(\theta)$ is the same as the arm with the largest $\theta^{\mathrm{T}}\mathbf{x}_a$. This allows us to speed up the computation by skipping the application of Φ.

For every experiment run with the fractional factorial bandit, a parallel experiment was run under the binomial bandit with the same 'true' success probabilities. Figure 12 compares the regret distributions across the 100 experiments for the fractional factorial and binomial bandits. The mean regret for the binomial bandit is 745 conversions out of 10 000 trials. The mean regret for the fractional factorial bandit is 166 conversions out of 10 000 trials, a factor of 4.5 improvement over the binomial bandit. Figure 13 compares the period-by-period regret distributions for the fractional factorial and binomial bandits. The regret falls to zero much faster under the fractional factorial bandit scheme. Figure 14 compares w_{at} for the fractional factorial and binomial bandits for one of the 100 experiments. In this particular experiment, the top four success probabilities are within 2% of one another. The fractional factorial bandit is able to identify the optimal arm within a few hundred observations. The binomial bandit remains confused after 10 000 observations.

6. CONCLUSION

This paper has shown how randomized probability matching can be used to manage the multi-armed bandit problem. The method is easy to apply, assuming one can generate posterior draws from $p(\theta|\mathbf{y}_t)$ by Markov chain Monte Carlo or other methods. Randomized probability matching performs reasonably well compared to optimal methods, when they are available, and it is simple

DOI: 10.1002/asmb

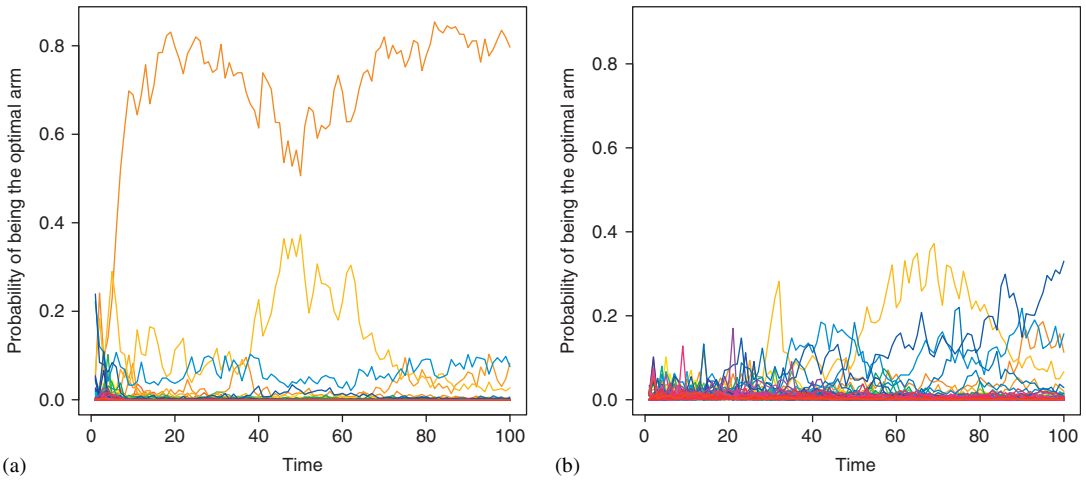


Figure 14. Evolution of w_{at} for one of the 100 fractional factorial bandit experiments in Section 5: (a) fractional factorial bandit and (b) binomial bandit.

enough to generalize to situations that optimal methods cannot handle. It is especially well suited to situations where learning occurs in batch updates, and it is robust in the sense that it had the lowest standard deviation of any method in all of the simulation studies that were tried. Finally, it combines features from several popular heuristics without the burden of specifying artificial tuning parameters.

I have illustrated the advantages of combining sequential and classical experiments by focusing on fractional factorial designs. Other important design ideas are similarly easy to incorporate. For example, randomized blocks for controlling non-experimental variation (i.e. variables which cannot be set by the experimenter) can be included in $\mathbf{x}_t$ either as first-order factors or as interactions. Thus, it is straightforward to control for temporal effects (e.g. day of week) or demographic characteristics of the customer using the product.

These techniques can be brought to several generalizations of multi-armed bandits by simply modifying the model used for the reward distribution. For instance, our examples focus on $\frac{0}{1}$ rewards, but continuous rewards can be handled by substituting regression models for logit or probit regressions. Restless bandits [25] that assume θ varies slowly over time can be handled by replacing $f_a(y|\theta)$ with a dynamic linear model [26]. Arm acquiring bandits [27] are handled gracefully by randomized probability matching by simply extending the design matrix used in the probit regression. Finally, one can imagine a network of experiments sharing information through a hierarchical model, practically begging for the name 'multi-armed mafia'.

APPENDIX A: DETAILS OF THE PROBIT MCMC ALGORITHMS

A.1. Posterior sampling for probit regression: standard case

This section describes algorithm introduced by Albert and Chib [24] to simulate draws from $p(\theta|\mathbf{y})$ in a probit regression model. Assume the prior distribution $\theta \sim \mathcal{N}(b, \Sigma)$, meaning the

DOI: 10.1002/asmb

normal distribution with mean vector μ and variance matrix Σ. Let $\mathcal{N}_a^+(\mu, \sigma^2)$ denote the normal distribution with mean μ and variance σ^2, truncated to only have support on the half line $z>a$. The complementary distribution $\mathcal{N}_a^-(\mu, \sigma^2)$ is truncated to have support on $z<a$. For the purpose of this section, let y_t be coded as $\frac{1}{-1}$ instead of $\frac{1}{0}$. The corresponding predictor variables are $\mathbf{x}_t$, and $\mathbf{X}$ is a matrix with $\mathbf{x}_t$ in row t.

1. For each t, simulate $z_t \sim \mathcal{N}_0^{y_t}(\theta^{\mathrm{T}}\mathbf{x}_t, 1)$.
2. Let $\mathbf{z}=(z_1, \ldots, z_n)$. Sample $\theta \sim \mathcal{N}(\tilde{\theta}, \Omega)$, where $\Omega^{-1}=\Sigma^{-1}+\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and $\tilde{\theta}=\Omega(\mathbf{X}^{\mathrm{T}}\mathbf{z}+\Sigma^{-1}b)$.

Repeatedly cycling through steps 1 and 2 produces a sequence of draws $(\theta, \mathbf{z})^{(1)}$, $(\theta, \mathbf{z})^{(2)}, \ldots$ from a Markov chain with $p(\theta, \mathbf{z}|\mathbf{y})$ as its stationary distribution. Simply ignoring $\mathbf{z}$ yields the desired marginal distribution $p(\theta|\mathbf{y})$.

A.2. Probit posterior sampling for denumerable experimental designs

When $\mathbf{x}_t$ contains only indicator variables, it is possible to list out all possible configurations of $\mathbf{x}$. Suppose there are k of them, stacked to form the $k \times p$ matrix $\hat{\mathbf{X}}$. Now let $\hat{\mathbf{y}}$ and $\hat{\mathbf{n}}$ be k-vectors with elements y_a and n_a denoting the number of conversions and trials from configuration a. Then $\mathbf{X}^{\mathrm{T}}\mathbf{X}=\hat{\mathbf{X}}^{\mathrm{T}}\mathrm{diag}(\hat{\mathbf{n}})\hat{\mathbf{X}}$, and one can approximate $\mathbf{X}^{\mathrm{T}}\mathbf{z}=\sum_t z_t\mathbf{x}_t=\sum_a \mathbf{x}_a \sum_{t:\mathbf{x}_t=\mathbf{x}_a} z_t=\sum_a \mathbf{x}_a z_a$ using the central limit theorem.

Let $z_a=z_a^+ + z_a^-$, where z_a^+ is the sum of y_a draws from $\mathcal{N}_0^+(\theta^{\mathrm{T}}\mathbf{x}_a, 1)$, and z_a^- is the sum of $n_a - y_a$ draws from $\mathcal{N}_0^-(\theta^{\mathrm{T}}\mathbf{x}_a, 1)$. Let $\lambda(\alpha)=\phi(\alpha)/\phi(-\mu_a)/(1-\Phi(\alpha))$. If y_a is large, then z_a^+ is approximately normal with mean $y_a\{\mu_a+\lambda(-\mu_a)\}$ and variance $y_a\{1-\lambda(-\mu_a)(\lambda(-\mu_a)+\mu_a)\}$. Likewise, let $\delta(\alpha)=\phi(\alpha)/\Phi(\alpha)$. If $n_a - y_a$ is large, then z_a^- is normal with mean $(n_a-y_a)(\mu_a - \delta(-\mu_a))$ and variance $(n_a-y_a)\{1+\mu_a\delta(-\mu_a)-(\delta(-\mu_a))^2\}$.

For configurations with large (>50) values of y_a or $n_a - y_a$, compute z_a^+ or z_a^- using its asymptotic distribution. If $y_a<50$, then compute z_a^+ by directly summing y_a draws from $\mathcal{N}_0^+(\mu_a, 1)$. Likewise if $n_a - y_a<50$, then compute z_a^- by directly summing $n_a - y_a$ draws from $\mathcal{N}_0^-(\mu_a, 1)$.

REFERENCES

1. Google. Available from: www.google.com/websiteoptimizer, 2010.
2. Whittle P. Discussion of 'bandit processes and dynamic allocation indices'. *Journal of the Royal Statistical Society*, *Series B*: *Methodological* 1979; **41**:165.
3. Gittins JC. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society*, *Series B*: *Methodological* 1979 **41**:148–177.
4. Bellman RE. A problem in the sequential design of experiments. *Sankhya Series A* 1956; **30**:221–252.
5. Brezzi M, Lai TL. Incomplete leraning from endogenous data in dynamic allocation. *Econometrica* 2000; **68**(6):1511–1516.
6. Thompson WR. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 1933; **25**:285–294.
7. Thompson WR. On the theory of apportionment. *American Journal of Mathematics* 1935; **57**(2):450–456.
8. Cox DR, Reid N. *The Theory of the Design of Experiments*. Chapman & Hall, CRC: London, Boca Raton, 2000.
9. Chaloner K, Verdinelli I. Bayesian experimental design: a review. *Statistical Science* 1995; **10**:273–304.
10. Tierney L. Markov chains for exploring posterior distributions (disc: P1728–1762). *The Annals of Statistics* 1994; **22**:1701–1728.
11. Doucet A, De Frietas N, Gordon N. *Sequential Monte Carlo in Practice*. Springer: Berlin, 2001.
12. Powell WB. *Approximate Dynamic Programming*: *Solving the Curses of Dimensionality*. Wiley: New York, 2007.
13. Sutton RS, Barto AG. *Reinforcement Learning*: *An Introduction*. MIT Press: Cambridge, MA, 1998.

DOI: 10.1002/asmb

14. Brezzi M, Lai TL. Optimal learning and experimentation in bandit problems. *Journal of Economic Dynamics and Control* 2002; **27**:87–108.
15. Gittins J, Wang Y-G. The learning component of dynamic allocation indices. *The Annals of Statistics* 1992; **20**:1625–1636.
16. Berry DA, Fristedt B. *Bandit Problems*: *Sequential Allocation of Experiments*. Chapman & Hall: London, 1985.
17. Lai T-L, Robbins H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 1985; **6**:4–22.
18. Lai T-L. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics* 1987; **15**(3):1091–1114.
19. Agrawal R. Sample mean based index policies with $o(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability* 1995; **27**:1054–1078.
20. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 2002; **47**:235–256.
21. Robbins H. A sequential decision procedure with a finite memory. *Proceedings of the National Academy of Science* 1956; **42**:920–923.
22. Luce D. *Individual Choice Behavior*. Wiley: New York, 1959.
23. Yang Y, Zhu D. Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *The Annals of Statistics* 2002; **30**:100–121.
24. Albert JH, Chib S. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* 1993; **88**:669–679.
25. Whittle P. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability* 1988; **25A**:287–298.
26. West M, Harrison J. *Bayesian Forecasting and Dynamic Models*. Springer: Berlin, 1997.
27. Whittle P. Arm-acquiring bandits. *The Annals of Probability* 1981; **9**(2):284–292.

DOI: 10.1002/asmb