EECS 495: Randomized Algorithms
Assignment 1

Michele Budinich, Bach Q. Ha, Benjamin Prosnitz

February 1, 2010

1 Introduction

Randomized algorithm analyses normally assume access to a sequence of unbiased truly random bits. Unfortunately, it is not always easy to generate random bits with this property. Sources of randomness often are biased in one way or another. Is it possible to use an biased source of randomness to produce an unbiased sequence of random bits?

To word it more formally, suppose we are given a biased coin $\mathbf{C}$, which outputs $\mathbf{H}$ with probability p and $\mathbf{T}$ with probability $(1-p)$. We might wish to generate a biased coin $\mathbf{C}'$ which outputs $\mathbf{H}'$ or $\mathbf{T}'$ with equal probability $\frac{1}{2}$.

In 1951, von Neumann suggested the following procedure: flip the biased coin twice, then, based on the outcomes, do one of the following:

- $\mathbf{HT}$: output $\mathbf{H}'$,
- $\mathbf{TH}$: output $\mathbf{T}'$,
- $\mathbf{TT}, \mathbf{HH}$: try again,

The intuition behind von Neumann's method is that one can combine sequences of $\mathbf{H}$ and $\mathbf{T}$ into events that have equal probability. Since the sequence $\mathbf{HT}$ has the same probability of occurring as $\mathbf{TH}$, if we output $\mathbf{H}'$ for one and $\mathbf{T}'$ for the other there will be an equal probability of producing $\mathbf{H}'$ and $\mathbf{T}'$. One can extend this idea to develop more efficient schemes.

1.1 Performance of von Neumann's Method

We now want to analyze the expected number of coin flips needed by the von Neumann procedure to output a single unbiased bit.

Let X be a random variable representing the number of coin flips before the algorithm terminates. Since the biased coin flips used by the algorithm are bernoulli trials, X follows a geometric distribution. Specifically, the success probability of each underlying bernoulli trial is the probability associated with either $\mathbf{HT}$ or $\mathbf{TH}$. Since the events are mutually exclusive, this is just the sum of the two (equal) probabilities, i.e. $2p(1-p)$.

One way to determine the expectation of a geometric random variable is using its memoryless property. To lighten up notation in the following derivation we assume X has success probability p' and failure probability $q' = (1-p')$, so that $\Pr(X=k) = pq^{k-1}$. The memoryless property of the geometric distribution can be shown by considering

$$\Pr(X = k+h | X > h) = \frac{p'q'^{k+h-1}\Pr(X > h | X = k+h)}{q'^{h}} = pq'^{k-1} = \Pr(X=k). \tag{1.1}$$

Consider the expectation of X conditioned on the result of the first bernoulli trial:

$$\mathbf{E}[X|X=1] = 1$$

If, on the other hand, the first trial fails (i.e. $X > 1$) we have

$$\begin{aligned}\mathbf{E}[X|X>1] &= \sum_{k=1}^{\infty} k\Pr(X=k|X>1) \\ &= \sum_{k=2}^{\infty} k\Pr(X=k|X>1)\end{aligned}$$

since $\Pr(X = 1|X > 1) = 0$. Changing variables, with $k' = k - 1$, we get

$$= \sum_{k'=1}^{\infty} (k' + 1)\Pr(X = k' + 1|X > 1)$$

and by (1.1)

$$\begin{aligned} &= \sum_{k'=1}^{\infty} (k' + 1)\Pr(X = k') \\ &= \sum_{k'=1}^{\infty} \Pr(X = k') + \sum_{k'=1}^{\infty} k'\Pr(X = k') \\ &= 1 + \mathbf{E}[X]. \end{aligned}$$

By the total probability theorem, $\mathbf{E}[X] = \mathbf{E}[\mathbf{E}[X|Y]]$, so

$$\begin{aligned} \mathbf{E}[X] &= p'\mathbf{E}[X|X = 1] + q'\mathbf{E}[X|X > 1] \\ &= p' + q'(1 + \mathbf{E}[X]) \\ &= 1 + q'\mathbf{E}[X], \end{aligned}$$

since $q' = 1 - p'$. So the expectation of a random variable with geometric distribution is simply

$$\mathbf{E}[X] = \frac{1}{p'}.$$

In von Neumann's procedure, the success probability at each iteration is $p' = 2p(1 - p)$, and since we need two biased coin flips per iteration, the expected number of coin flips needed to output a single unbiased bit is

$$\text{expected nr. of flips} = 2\mathbf{E}[X] = 2\frac{1}{p'} = \frac{1}{p(1 - p)}. \tag{1.2}$$

This formula follows intuitive sense. If we were to take $p = 1$ or $p = 0$, only **H** or **T** would result and it would take infinitely long to terminate. The best coin for this method is clearly the coin that is already unbiased (though one should note that the method is clearly inefficient for an unbiased coin).

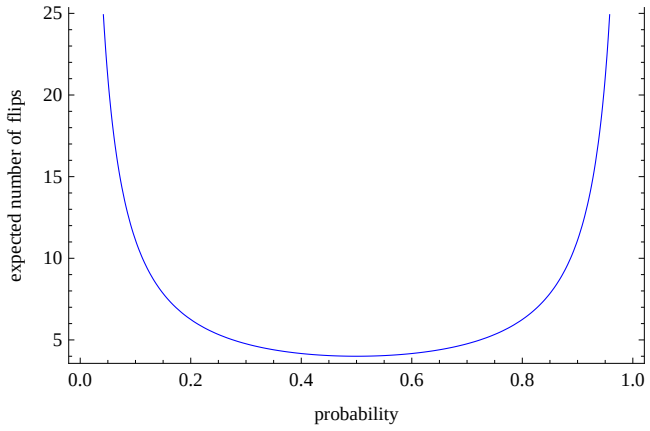

Figure 1: Expected number of biased flips needed to obtain one unbiased bit using the procedure outlined by von Neumann.

2 Improvements

2.1 Optimizing for a Specific Input Coin

Suppose that we are given a coin of known bias, with the probability of getting **H** being $p = \frac{2}{3}$. In expectation, von Neumann's procedure would require $\frac{9}{2}$ coin flips. Can we improve von Neumann's method using the fact that we know the bias of this input coin?

Consider the following procedure:

1. Flip two input coins
2. depending on the values of these coins:
   - **HT**,**TH**: output $\mathbf{T}'$ and terminate
   - **HH**: output $\mathbf{H}'$ and terminate
   - **TT**: go to step 1

Note that the probability of getting **HH** is $\frac{4}{9}$ and the probability of getting **HT** or **TH** are each $\frac{2}{9}$. Therefore, the probabilities of getting $\mathbf{H}'$ or $\mathbf{T}'$ are both $\frac{4}{9}$. It is clear that this method must be better than the original once, since in case the double flip results in **HH**, this specialized procedure will terminate while the general one will not.

The probability that an iteration fails (i.e. doesn't output) is now $q' = \Pr(\mathbf{TT}) = \frac{1}{9}$. Substituting the success probability $p' = 8/9$ in (1.2) we get

$$\text{expected nr. of flips} = 2\mathbf{E}[X] = 2\frac{1}{p'} = \frac{9}{4}.$$

Thus the specialized procedure *halves* the expected number of tosses.

2.2 Generic Input Coins

1. As in von Neumann, flip two pairs of coins repeatedly until the algorithm terminates
2. Evaluate the flips $\mathbf{C}_1, \ldots, \mathbf{C}_n$ as follows:
   (a) Let $k = \lfloor \log_2 n \rfloor$, where n is the number of flips
   (b) For $j \in \{1, \ldots, k\}$:
      i. If $n \mod 2^k \neq 0$, skip this value of j
      ii. Consider $g = (\mathbf{C}_{n-2^k+1}, \ldots, \mathbf{C}_{n-2^{k-1}})$ and $h = (\mathbf{C}_{n-2^{k-1}+1}, \ldots, \mathbf{C}_n)$
      iii. If $g = (\mathbf{H}, \ldots, \mathbf{H})$ and $h = (\mathbf{T}, \ldots, \mathbf{T})$, output $\mathbf{H}'$ and terminate
      iv. If $g = (\mathbf{T}, \ldots, \mathbf{T})$ and $h = (\mathbf{H}, \ldots, \mathbf{H})$, output $\mathbf{T}'$ and terminate
   (c) Go to step 1 (if it did not terminate)

This algorithm is effectively von Neumann operating at all possible scales. It terminates if a sequence of mixed heads or tails is found, such as **HT** or **TTHH**. When the number of flips is a power of two, the algorithm only does not terminate if the flips are all heads or all tails. As a result, depending on p, the probability of not terminating after a given number of flips can become extremely small quickly.

A detailed analysis of this method is challenging. However, it is clear that its expected number of flips is smaller than that of von Neumann because the times when it terminates are a superset of those of the original von Neumann method.

2.3 A Simpler Algorithm

If we simplify the above algorithm, we can obtain a procedure that still outperforms von Neumann (although it's worse than the one just presented) and for which an analysis is possible. The algorithm is the following:

1. flip two coins
   (a) if $\mathbf{C}_{n-1}\mathbf{C}_n$ (i.e. the two flips just generated) are of type **HT** or **TH** output the value of the last flip and terminate,
   (b) otherwise, if n is a multiple of 4
      - if the suffix of length 4 is of the following kind

        HHTT
        TTHH

      output the value of the last coin and terminate,
   (c) otherwise continue (i.e. go to step 1).

Again it is easy to see that this algorithm does better than von Neumann's, since it always terminates when von Neumann's does and it also terminates in cases in which von Neumann's doesn't (e.g. **HHTT**). The correctness can be easily verified, since **HHTT** and **TTHH** have the same probability.

Analysis It is possible to analyze the expected number of flips for this method. The expected number of flips when only considering cases in which we terminate on a set of 4 flips is:

$$2 + 2(p^2 + (1-p)^2)$$

If we consider the process on the granularity of 4 flips, the procedure is memoryless so we can solve for its expected value:

$$\begin{aligned} E[X] &= [2 + 2(p^2 + (1-p)^2)] + (p^4 + (1-p)^4)E[X] \\ &= \frac{2 + 2(p^2 + (1-p)^2)}{1 - p^4 - (1-p)^4} \end{aligned}$$

Figure 2 shows the expectation of the improved method (in red) compared to von Neumann's (blue).

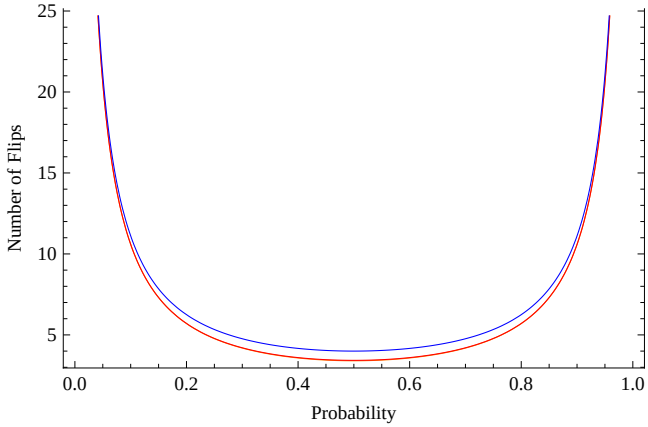

Figure 2: Comparison of the expected number of flips from von Neumann's method (blue line) and the improved algorithm proposed (red line)

3 Producing a Biased Output

It is possible to use an unbiased coin generator to produce a desired biased output. Suppose that we want to produce an output coin $\hat{\mathbf{C}}$ and generate $\hat{\mathbf{H}}$ with a probability of q or $\hat{\mathbf{T}}$ with a probability of $(1-q)$.

As before, let $\mathbf{C}'$ be the unbiased coin that outputs values $\mathbf{H}'$ or $\mathbf{T}'$ with probability $\frac{1}{2}$.

Consider the bits representing 1/2 and below in the binary representation of q, S_q. Also consider the binary representation of $1-q$, S_{1-q}. Since S_{1-q} is the two's complement of S_q, it is equal to $\overline{S_q}$ up until the lowest bit that is 1, at which point they are the same. Because of this, we can perform the following procedure to generate a biased output:

1. Let $i = 1$
2. Flip $\mathbf{C}'$
3. Do one of the following:
   - If $S_q(i) = S_{1-q}(i)$, let $\hat{\mathbf{C}} = \mathbf{C}'$ and stop
   - If $\mathbf{C}' = \mathbf{H}'$ and $S_q(i) = 1$, output $\hat{\mathbf{H}}$ and stop
   - If $\mathbf{C}' = \mathbf{T}'$ and $S_{1-q}(i) = 0$, output $\hat{\mathbf{T}}$ and stop
   - Else let $i = i + 1$ and go to step 2

Note that when bit i is 1, the probability of outputting $\hat{\mathbf{H}}$ is (1/2) and likewise the probability of outputting $\hat{\mathbf{T}}$ when bit i is 0 is (1/2). However, we can see that the procedure is memoryless leading to an exponential probability of a given event being outputted. So the probability of terminating on step i is $(1/2)^i$. The probability of outputting $\hat{\mathbf{H}}$ is then $(\frac{1}{2})^k + \sum_{i=1}^{k-1} S_q(i)(\frac{1}{2})^i = q$, where k is the lowest bit of q that is 1 (if one exists). Because of this, the appropriate biased output is generated.

Now, note that the probability of matching up with a given value on step i is $(1/2)^i$. The expected number of steps to output in the case where there is no lowest 1 bit is $\sum_i^\infty (\frac{1}{2})^i = \frac{1}{1-1/2} = 2$. So in expectation it takes no more than twice as long to output a biased coin as it does to output an unbiased one.