Topics in
Probability Theory and Stochastic Processes
Steven R. Dunbar

The Hitting Time Theorem

Rating
Mathematicians Only: prolonged scenes of intense rigor.
Section Starter Question

Suppose a simple random walk of 6 steps with $Y_i = \pm 1$ has 4 steps +1 and 2 steps −1. Explicitly enumerate all the possible walks. In how many ways does the walk stay less than 2 throughout the entire walk until step 6?

Key Concepts

1. The Hitting Time Theorem expresses a conditional probability. The Hitting Time Theorem says that under reasonable, or even relatively weak conditions on the steps, the conditional probability that an $n$-step random walk starting at 0, given that it ends at $k > 0$, first hits $k$ at time $n$ is $k/n$.

2. If the $Y_i$ are i. i. d. r. v. or the joint probability distribution of $(Y_1, \ldots, Y_n)$ is invariant under rotations or exchangeable, then

\[ \mathbb{P}[T_n = k, T_t < k, \text{ for all } t < n] = \frac{k}{n} \cdot \mathbb{P}[T_n = k] \]

3. The Hitting Time Theorem is equivalent to the Ballot Theorem.

4. The conditions of invariance under rotations and exchangeability are weaker than the traditional requirement that $Y_i$ be independent and identically distributed.
Vocabulary

1. The **Hitting Time Theorem** shows the conditional probability that an \( n \)-step random walk with steps \( (Y_1, \ldots, Y_n) \) having a joint probability distribution with reasonable conditions starting at 0, given that it ends at height \( k > 0 \), first hits \( k \) at time \( n \) is \( k/n \).

2. If the joint probability distribution of \( (Y_1, \ldots, Y_n) \) is the same as the joint probability distribution of the rotated sequence

\[
(Y_{r+1}, \ldots, Y_n, Y_1, \ldots, Y_r)
\]

then we say the distribution is **invariant under rotations**.

3. A random vector \( (Y_1, \ldots, Y_n) \) is **exchangeable** if the joint probability distribution is invariant for any permutation of \( (Y_1, \ldots, Y_n) \).

Mathematical Ideas

Background and History

The Ballot Theorem can be interpreted as the probability that, during the vote counting in an election between two candidates, the winning candidate is always ahead of the other. Suppose that in an election with \( n \) votes, candidate A receives \( a \) votes and candidate B receives \( b \) votes, where \( a - b = k \) for some positive integer \( k \). Assume that all ballot permutations are equally likely during the counting. By dividing by the total number of ballot permutations in the counting of the specialized ballot theorem, the probability is \( k/n \), see [The Ballot Theorem and the Reflection Principle](#).

The problem can be restated as the probability that an \( n \)-step random walk taking independent and identically distributed \( \pm 1 \) steps stays positive after time 0, given that it ends at height \( k > 0 \), is \( k/n \). If each step has probability \( 1/2 \), this probability can be derived easily using the reflection
principle, an argument appearing in many textbooks. In fact, it is possible to prove the same result under weaker hypotheses, [2], [5].

The Hitting Time Theorem says the conditional probability that an \( n \)-step random walk starting at 0, \( \text{given that it ends at } k > 0 \), first hits \( k \) at time \( n \) is \( k/n \).

**Remark.** Note that:

- The Hitting Time Theorem expresses a conditional probability.
- By time reversal and symmetry around \( k \), the Hitting Time Theorem is equivalent to the statement that an \( n \)-step random walk taking independent and identically distributed \( \pm 1 \) steps stays positive after time 0, given that it ends at height \( k > 0 \).
- Note that the Hitting Time Theorem is different from the Gambler’s Ruin. In its usual form the Gambler’s Ruin gives the probability that a random walk, given that it starts from a positive value ever hits the value \( k \) before it hits the value 0.
- The Hitting Time Theorem is also different from the Positive Walks Theorem. The Positive Walks Theorem gives the probability that an \( n \)-step random walk starting at 0 is always positive throughout the remainder of the walk, \( \text{without regard to the value of the endpoint} \).
- The Hitting Time Theorem has other applications to branching processes and random graphs, see [5] for references.

In 1949 Otter gave the first proof of the Hitting Time Theorem for \( k = 1 \). Kemperman in 1961 and Dwass in 1969 gave proofs for \( k \geq 2 \) using generating functions and the Lagrange inversion formula. For simple Bernoulli random walks making steps of size \( \pm 1 \), a proof using the Reflection Principle is in [1]. See [5] for references and discussion.

**The Hitting Time Theorem for Independent Identically Distributed Random Variables**

Fix \( n \geq 1 \). Let \( (Y_1, \ldots, Y_n) \) be a sequence of random variables \( Y_i \) taking values in \( \{\ldots, -2, -1, 0, 1\} \). Note that this is more general than, but includes, the standard random walk where the random variables take values in \( \{-1, 1\} \).
Define the walk $T = (T_0, \ldots, T_n)$ as the usual piecewise linear function defined by the values $T_i = \sum_{j=0}^i Y_j$ at the integers $i = 0, 1, 2, \ldots, n$. Note that because the only positive value assumed by the steps $Y_i$ is 1, for the random walk $T$ to gain a height $k$, it has to pass through all $k-1$ intermediate integer values.

**Theorem 1** (Hitting Time Theorem). Let $k \geq 0$. For a random walk starting at 0 with i. i. d. integer-valued steps $Y_i$ with $Y_i \leq 1$ for $i = 1, 2, 3 \ldots$ then

$$
P[T_n = k, T_t < k, \text{ for all } t < n] = \frac{k}{n} \cdot P[T_n = k].$$

**Proof.** The proof proceeds by induction on $n \geq 1$. When $n = 1$, both sides are 0 when $k > 1$ and $k = 0$, and are equal to $P[Y_1 = 1]$ when $k = 1$. This is the base case of the induction.

For the induction step, take $n \geq 2$ and consider the case $k = 0$. In this case, the left side requires $T_t < 0$ for all $t < n$, yet $T_0 = 0$ so the event is empty and the probability is 0. The right side is also equal to 0 so the case $k = 0$ is satisfied. Thus, we may assume that $k \geq 1$.

So take $n \geq 2$ and consider the case $k > 0$. Condition on the first step to obtain

$$
P[T_n = k, T_t < k \text{ for all } t < n] = \sum_{s = -\infty}^1 P[T_n = k, T_t < k \text{ for all } t < n \mid Y_1 = s] P[Y_1 = s].$$

For a given $s$ and $m = 0, \ldots, n-1$, let $T_m^{(s)} = T_{m+1} - s$. Because the steps are independent, identically distributed random variables, by the random walk Markov property, with $T_0 = 0$ and conditionally on $Y_1 = s$ the random walk $\{T_m^{(s)}\}_{m=0}^{n-1}$ has the same distribution as the random walk path $\{T_m\}_{m=0}^{n-1}$.

That is,

$$
P[T_n = k, T_t < k \text{ for all } t < n \mid Y_1 = s] = P[T_{n-1}^{(s)} = k-s, T_t^{(s)} < k-s \text{ for all } t < n-1] = \frac{k-s}{n-1} P[T_{n-1}^{(s)} = k-s]$$
where in the last equality we have used the induction hypothesis which is allowed since $n - 1 \geq 1$ and $k \geq 1$ with $s \leq 1$ so $k - s \geq 0$. Because the steps are independent, identically distributed random variables $P \left[ T_{n-1}^{(s)} = k - s \right] = P \left[ T_n = k \mid Y_1 = s \right]$, (see Figure 1) substitute into the summation to obtain

$$P \left[ T_n = k, T_t < k \text{ for all } t < n \right] = \sum_{s=\infty}^{1} \frac{k-s}{n-1} P \left[ T_n = k \mid Y_1 = s \right] P \left[ Y_1 = s \right].$$

Temporarily ignore the $n - 1$ in the denominator and consider the summation

$$\sum_{s=\infty}^{1} (k-s) P \left[ T_n = k \mid Y_1 = s \right] P \left[ Y_1 = s \right].$$
Figure 2: Example of the Hitting Time Theorem with \( n = 6, k = 2, a = 4, b = 2 \).

By the properties of conditional expectation

\[
P[T_n = k | Y_1 = s] P[Y_1 = s] = P[T_n = k, Y_1 = s] = P[Y_1 = s | T_n = k] P[T_n = k]. \tag{1}
\]

Thus,

\[
\sum_{s=\infty}^{1} (k - s) P[T_n = k | Y_1 = s] P[Y_1 = s] = \sum_{s=\infty}^{1} (k - s) P[Y_1 = s | T_n = k] P[T_n = k] = P[T_n = k] (k - E[Y_1 | T_n = k]).
\]

By the assumption of independent, identically distributed steps, the conditional expectation \( E[Y_i | T_n = k] \) is independent of \( i \) so that

\[
E[Y_i | T_n = k] = \frac{1}{n} \sum_{i=1}^{n} E[Y_i | T_n = k] = \frac{1}{n} E \left[ \sum_{i=1}^{n} Y_i | T_n = k \right] = \frac{k}{n},
\]

since \( \sum_{i=1}^{n} Y_i = T_n = k \) when \( T_n = k \). Recalling the temporarily ignored factor of \( \frac{1}{n-1} \) we arrive at

\[
P[T_n = k, T_i < k, \text{ for all } t < n] = \frac{1}{n-1} \left[ k - \frac{k}{n} \right] \cdot P[T_n = k] = \frac{k}{n} P[T_n = k].
\]

This completes the proof by induction. \( \square \)

Example. See Figure 2 for an example.
Example Application of the Hitting Time Theorem

The following example is adapted from [3] and [4]. The original problem is:

The Kentucky Derby is on Saturday, and a field of 20 horses is slated to run “the fastest two minutes in sports” in pursuit of the right to be draped with a blanket of roses. But let’s consider, instead, the Lucky Derby, where things are a little more bizarre:

The bugle sounds, and 20 horses make their way to the starting gate for the first annual Lucky Derby. These horses, all trained at the mysterious Riddler Stables, are special. Each second, every Riddler-trained horse takes one step. Each step is exactly one meter long. But what these horses exhibit in precision, they lack in sense of direction. Most of the time, their steps are forward (toward the finish line) but the rest of the time they are backward (away from the finish line). As an avid fan of the Lucky Derby, you’ve done exhaustive research on these 20 competitors. You know that Horse One goes forward 52 percent of the time, Horse Two 54 percent of the time, Horse Three 56 percent, and so on, up to the favorite filly, Horse Twenty, who steps forward 90 percent of the time. The horses’ steps are taken independently of one another, and the finish line is 200 meters from the starting gate.

Handicap this race and place your bets! In other words, what are the odds (a percentage is fine) that each horse wins?

It is easy to create a simulation for this scenario. A sample script is in the scripts section below. The simulation of 2000 races suggests that the probability of horse 20 winning is about 70.48% and the probability is about 22.86%.

Consider a particular horse. For each step, suppose it moves forward with probability $p$ (in this example $p > 1/2$) or backward with probability $1 - p$. Recall that reaching an even-numbered position $2k$ (such as 200 in the problem statement) can only occur in an even number of steps, say $2n$. Then the horse must take $n + k$ forward steps and $n - k$ backward steps. The horse starts at position 0 and we would like to know the probability that it will reach position $2k$ (where specifically in the example $k = 100$ with $2k = 200$) in exactly $2n$ steps. For convenience, adopt the following notation:

- $2k$ is a fixed parameter, the value to be reached;
• $2n$ is a possible number of steps required to reach the value $2k$ where $2n \geq 2k$;

• let $H$ be the random variable for the hitting time, that is, $H = 2n$ if $T_{2n} = 2k, T_t < 2k$, for all $t < 2n$; and

• for $2n \geq 2k$

\[
P[H = 2n] = P[T_{2n} = 2k, T_t < 2k, \text{ for all } t < 2n]
\]

\[
= \frac{k}{n} \cdot P[T_{2n} = 2k]
\]

\[
= \frac{k}{n} \binom{2n}{n+k} p^{n+k}(1-p)^{n-k}
\]

\[
= P(2n; 2k, p).
\]

This probability mass distribution does not seem to have a common name, but it is still possible to compute the statistics of this distribution.

**Theorem 2.**  
1. $P(2n; 2k, p)$ is a probability mass distribution, that is

\[
\sum_{n=k}^{\infty} P(2n; 2k, p) = 1.
\]

2. $E[H] = \frac{2k}{2p-1}$, that is

\[
\sum_{n=k}^{\infty} (2n) P(2n; 2k, p) = \frac{k}{2p-1}.
\]

3. Var $[H] = \frac{8kp(1-p)}{(2p-1)^3}$, that is

\[
\sum_{n=k}^{\infty} (2n - \frac{k}{2p-1})^2 P(2n; 2k, p) = \frac{8kp(1-p)}{(2p-1)^3}
\]

**Proof.** Starting with

\[
\sum_{n=k}^{\infty} \frac{k}{n} \binom{2n}{n+k} p^{n+k}(1-p)^{n-k}
\]
change the variables $n = j + k$ to rewrite the summation as

$$p^{2k} \sum_{j=0}^{\infty} \frac{k}{j+k} \binom{2(j+k)}{j} p^j (1-p)^j.$$  

This series resembling a binomial series expansion suggests that the series might have a closed form. In fact, using Maple to find a closed form solution, obtain

$$\sum_{j=0}^{\infty} \frac{k}{j+k} \binom{2(j+k)}{j} x^j = \frac{2^k}{(1 + \sqrt{1 - 4x})^{2k}}.$$  

In turn the closed form for the power series suggests that it might be possible to express the moment-generating function in closed form. Again using Maple to find a closed form expression

$$F_H(t) = \mathbb{E} [e^{tH}] = \sum_{n=k}^{\infty} \frac{k}{n} \binom{2n}{n+k} p^{n+k} (1-p)^{n-k} e^{2nt} = \frac{(2p)^{2k} e^{2kt}}{(1 + \sqrt{1 - 4e^{2t}p + 4e^{2t}p^2})^{2k}}.$$  

Then, once more using Maple, it is easy to check that

$$F_H(0) = \sum_{n=k}^{\infty} P(2n; 2k, p) = 1$$

and

$$\mathbb{E} [H] = F_H'(0) = \frac{2k}{2p - 1}.$$  

The calculation of the variance is left as an exercise. \[\square\]

For the horse race example, the mean number and standard deviation of steps required for each horse to reach position 200 are in Table 1.  

The probability distributions seem to be approximately normal in shape, see Figure. This is reasonable since the hitting time to level $2k$ is the sum of $2k$ random one-level hitting times, first the hitting time from the initial point 0 to first reach 1, then the independent hitting time to first reach 2 from 1, and so on. The one-level hitting times are independent since the
<table>
<thead>
<tr>
<th>Horse</th>
<th>p</th>
<th>Mean</th>
<th>Variance</th>
<th>StdDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.52</td>
<td>5000.</td>
<td>3120000.</td>
<td>1766.3522</td>
</tr>
<tr>
<td>2</td>
<td>0.54</td>
<td>2500.</td>
<td>388125.</td>
<td>622.99679</td>
</tr>
<tr>
<td>3</td>
<td>0.56</td>
<td>1666.6667</td>
<td>114074.07</td>
<td>337.74853</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>1250.</td>
<td>47578.125</td>
<td>218.12410</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>1000.</td>
<td>24000.</td>
<td>154.91933</td>
</tr>
<tr>
<td>6</td>
<td>0.62</td>
<td>833.33333</td>
<td>13634.259</td>
<td>116.76583</td>
</tr>
<tr>
<td>7</td>
<td>0.64</td>
<td>714.28571</td>
<td>8396.5015</td>
<td>91.632426</td>
</tr>
<tr>
<td>8</td>
<td>0.66</td>
<td>625.</td>
<td>5478.5156</td>
<td>74.016995</td>
</tr>
<tr>
<td>9</td>
<td>0.68</td>
<td>555.55556</td>
<td>3731.1385</td>
<td>61.083046</td>
</tr>
<tr>
<td>10</td>
<td>0.7</td>
<td>500.</td>
<td>2625.</td>
<td>51.234754</td>
</tr>
<tr>
<td>11</td>
<td>0.72</td>
<td>454.54545</td>
<td>1893.3133</td>
<td>43.512220</td>
</tr>
<tr>
<td>12</td>
<td>0.74</td>
<td>416.66667</td>
<td>1391.7824</td>
<td>37.306600</td>
</tr>
<tr>
<td>13</td>
<td>0.76</td>
<td>384.61538</td>
<td>1037.7788</td>
<td>32.214574</td>
</tr>
<tr>
<td>14</td>
<td>0.78</td>
<td>357.14286</td>
<td>781.70554</td>
<td>27.958997</td>
</tr>
<tr>
<td>15</td>
<td>0.8</td>
<td>333.33333</td>
<td>592.59259</td>
<td>24.343225</td>
</tr>
<tr>
<td>16</td>
<td>0.82</td>
<td>312.5</td>
<td>450.43945</td>
<td>21.223559</td>
</tr>
<tr>
<td>17</td>
<td>0.84</td>
<td>294.11765</td>
<td>341.94993</td>
<td>18.491888</td>
</tr>
<tr>
<td>18</td>
<td>0.86</td>
<td>277.77778</td>
<td>258.05898</td>
<td>16.064214</td>
</tr>
<tr>
<td>19</td>
<td>0.88</td>
<td>263.15789</td>
<td>192.44788</td>
<td>13.872559</td>
</tr>
<tr>
<td>20</td>
<td>0.9</td>
<td>250.</td>
<td>140.625</td>
<td>11.858541</td>
</tr>
</tbody>
</table>

Table 1: Statistics for the hitting time distribution for each horse in the race.
Figure 3: Representative probability mass distributions for horses 20, 16, 12, and 8.
individual steps of the random walk are independent. Thus, as the sum of many independent random variables, the distribution of the hitting time to $2k$ is approximately normal.

Suppose each horse $i$ finishes in $h_i$ steps. Computing the probability that specific horse $j$ wins the race amounts to finding the probability that $h_i > h_j$ for all $i \neq j$. Because the probability distribution is completely specified, it is possible to explicitly compute the probability that for example horse 20 is the winner. In R, fill vectors of length 10,001 with the probability distribution for each horse. This covers the probability of each horse winning in the minimum possibility of 200 steps up to 20,000 steps, about 3 standard deviations more than the mean number of steps required for the slowest horse 1 to win. Calculating a cumulative sum along this vector, gives the cumulative probability distribution for each horse. Subtracting the cumulative probability distribution from 1 is the complementary cumulative probability. At each index or step number $h_j$, this is the probability that horse reaches the goal in more than $h_j$ steps. Multiply all of these probabilities from horse 1 to horse 19, and then multiply by the probability of horse 20 winning in $h_j$ steps. Computing the product of so many small magnitude factors could lead to numerical underflow, so instead compute the exponential of the sum and of the respective logarithms. Summing over all possible indices gives the probability of horse 20 winning over all other horses, namely 0.7041477

Since the probability distributions are approximately normal, one can also approximate this probability with:

$$
\mathbb{P}[h_i > h_j, i \neq j] \approx \int_{-\infty}^{\infty} \mathbb{P}[h_j = x] \prod_{i \neq j} \mathbb{P}[h_i > x] \, dx \\
= \int_{-\infty}^{\infty} \frac{1}{\sigma_j} \phi \left( \frac{x - \mu_j}{\sigma_j} \right) \prod_{i \neq j} \left( 1 - \Phi \left( \left( \frac{x - \mu_i}{\sigma_i} \right) \right) \right) \, dx.
$$

The probability calculation uses the usual notation: $\phi$ is the standard normal probability density (pdf), $\Phi$ is the cumulative distribution (cdf) of the standard normal distribution, and $\mu_k$ and $\sigma_k$ are the means and standard deviations computed in Table 1.

Analytic integration of the probability is impractical, and even the nu-
Numerical integration of
\[ H_j(x) = \frac{1}{\sigma_j} \phi \left( \frac{x - \mu_j}{\sigma_j} \right) \prod_{i \neq j} \left( 1 - \Phi \left( \frac{x - \mu_i}{\sigma_i} \right) \right) \]
is challenging. Taking \( H_{20}(x) \) as an example, the product term
\[ \prod_{i \neq 20} \left( 1 - \Phi \left( \frac{x - \mu_i}{\sigma_i} \right) \right) \]
is approximately 1 for \( x < \mu_{19} - 3\sigma_{19} \approx 221.5402 \). Since each factor is
less than 1 and \( 1 - \Phi((x - \mu_{19})/\sigma_{19}) \approx 0 \) for \( x > \mu_{19} + 3\sigma_{19} \approx 304.7756 \),
the product decreases to approximately 0 for \( x > 304.7756 \). The other
factor \( \frac{1}{\sigma_{20}} \phi \left( \frac{x - \mu_{20}}{\sigma_{20}} \right) \approx 0 \) outside the interval \((\mu_{20} - 3\sigma_{20}, \mu_{20} + 3\sigma_{20}) \approx (214.4244, 285.5756)\) and has a maximum value of \( \phi(0) \approx 0.03364 \) at \( x = 250 \).
Therefore, \( H_{20}(x) \) is approximately 0 outside the interval \((214.4244, 285.5756)\) and reaches a maximum approximately \( H_{20}(250) \approx 0.02633 \). Again, computing
the product of so many small magnitude factors could lead to numerical underflow, so instead the exponential of the sum and difference of the respective logarithms is computed.

The low variation with an aspect ratio of \( \frac{1}{\sigma_{20}} (\phi(0))/(6\sigma_{20}) \approx 1/2115 \)
makes even adaptive integration challenging. The default integration rou-
tine of \( R \) gives a nonsensical answer of 1.311008. With the additional \( R \) library \texttt{ pracma} of advanced numerical analysis routines, the \texttt{quad} function
using adaptive Simpson quadrature yields 0.7086704 and the \texttt{quadl} function
using adaptive Lobatto quadrature yields the very similar 0.7078003.
Numerical integration with Scientific Python using either general purpose
integration or Romberg integration gives similar results of 0.70867.

The relative error between the exact discrete calculation and the approx-
imate calculation with the normal distribution is 0.006422374.

Therefore we estimate horse 20 has about a 71\% chance of winning. Re-
peating the analysis with the necessary changes, horse 19 has about a 21\% chance of winning. One of the other 18 horses winning has about an 8\% chance.

The Hitting Time Theorem Under Rotation Invariance

Fix \( n \geq 1 \). Let \( (Y_1, \ldots, Y_n) \) be a sequence of random variables \( Y_i \) taking
values in \( \{ \ldots, -2, -1, 0, 1 \} \). Define the walk \( T = (T_0, \ldots, T_n) \) as the usual
Table 2: Example of a rotation of a Bernoulli random walk with \( Y_i = \pm 1 \), \( n = 10 \), and \( r = 5 \).  

<table>
<thead>
<tr>
<th>( j )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_j )</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( T_j )</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>( Y^{(5)} )</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>( T^{(5)} )</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

piecewise linear function defined by the values \( T_i = \sum_{j=0}^{i} Y_j \) at the integers \( i = 0, 1, 2, \ldots, n \). Note also that because the only positive value assumed by the steps \( Y_i \) is 1, for the random walk \( T \) to gain a height \( k \), it has to pass through all \( k - 1 \) intermediate integer values. Given \( r \) with \( 0 < r \leq n \), define the rotation of \( T \) by \( r \) as the walk \( T^{(r)} = (T_0^{(r)}, \ldots, T_n^{(r)}) \) corresponding to the rotated sequence \( (Y_{r+1}, \ldots, Y_n, Y_1, \ldots, Y_r) \). Another way to represent the rotation of the sequence is

\[
T^{(r)}_t = \begin{cases} 
T_{t+r} - T_r & 0 \leq t \leq n - r; \\
T_n - T_r + T_{t+r-n} & n - r < t \leq n.
\end{cases}
\]

Note that \( T^{(r)}_n = T_n + T_{n+r-n} - T_r = T_n \) and the entire walk rotated by \( n \) is the original walk, \( T^{(n)} = T \). We say that \( T^{(r)} \) peaks at \( n \) if \( T^{(r)}_t < T^{(r)}_n = T_n \) for all \( t < n \). See Table 3 for several examples.

**Lemma 3.** If \( T_n = k \geq 1 \), then exactly \( k \) rotations of \( T \) peak at \( n \).

*Proof.* Set \( M = \max\{T_0, \ldots, T_n\} \). If \( T_r \leq T_s \) for some \( r \) and \( s \) such that \( 0 \leq s < r \leq n \), then \( T^{(r)}_{n-r+s} = T_n + T_s - T_r \geq T_n \), so \( T^{(r)} \) can only peak at \( n \) if \( r = r_i = \min(t > 0 \mid T_t = i) \) for some \( i \) satisfying \( 0 = T_0 < i \leq M \). That is, \( T^{(r)} \) can only peak at \( n \) if there are no indices prior to \( r \) with values greater than \( T_r \). Since the values of \( T_j \) has to pass through all intermediate integer values, \( r \) must be \( r_i = \min(t > 0 \mid T_t = i) \). The property of passing through all intermediate values implies that \( 0 < r_1 < r_2 < \cdots < r_M \). Therefore, for \( 1 \leq i \leq M \), \( \max\{T^{(r_i)}_t \mid 0 \leq t \leq n-r_i\} = \max\{T_t : r_i \leq t \leq n\} - T_{r_i} = M - i \), and \( \max\{T^{(r_i)}_t \mid n-r \leq t < n\} = T_n + \max\{T_t \mid 0 \leq t < r_i\} - T_{r_i} = T_n - 1 \). It follows that \( T^{(r_i)} \) peaks at \( n \) if and only if \( M - T_n < i \leq M \). \( \square \)
<table>
<thead>
<tr>
<th>$j$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_j$</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(1)}$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j^{(1)}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(2)}$</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j^{(2)}$</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(3)}$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$T_j^{(3)}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(4)}$</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j^{(4)}$</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(5)}$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$T_j^{(5)}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(6)}$</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j^{(6)}$</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(7)}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$T_j^{(7)}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(8)}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$T_j^{(8)}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$Y_j^{(9)}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_j^{(9)}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3: Example of the lemma with $n = 10$, counting the rotations which peak at 10.
Example. This example is a specific illustration of the proof of the lemma with \( n = 10 \) and the random walk given in Table 3.

For the example, \( M = \max\{T_0, \ldots, T_{10}\} = 4 \). For the example \( 1 = T_5 \leq T_4 = 2 \) for \( r = 5 \) and \( s = 4 \) such that \( 0 \leq 4 = s < r = 5 \leq n = 10 \). Now consider \( T_{10-5+4}^{(5)} = T_9^{(5)} = 5 = T_{10} + T_4 - T_5 = 4 + 2 - 1 = 5 \geq 4 \). Therefore, \( T^{(5)} \) cannot peak at \( n = 10 \). Table 2 also illustrates that \( T^{(5)} \) cannot peak at \( n = 10 \).

Now consider \( i = 2 \) satisfying \( 0 = T_0 < i = 2 \leq M = 4 \). Then \( 2 = r_2 = \min(t > 0 \mid T_t = 2) \). Note that \( T^{(2)} \) peaks at \( n = 10 \).

The definition of \( r_i \) is \( r_i = \min(t > 0 \mid T_t = i) \). For the example, \( 0 < 1 = r_1 < 2 = r_2 < 9 = r_3 < 10 = r_4 \). The next point in the lemma is that for \( 1 \leq i \leq M \), \( \max\{T_t^{(r_i)} \mid 0 \leq t \leq n - r_i\} = \max\{T_t : r_i \leq t \leq n\} - T_{r_i} = M - i \). For this example, Table 2 gives all values for comparison, recalling that \( T^{(10)} = T \).

\[
\begin{array}{cccc}
    r_i & \max\{T_t^{(r_i)} \mid 0 \leq t \leq n - r_i\} & \max\{T_t : r_i \leq t \leq n\} - T_{r_i} & M - i \\
    1 & 3 & 4 - 1 = 3 & 4 - 1 = 3 \\
    2 & 2 & 4 - 2 = 2 & 4 - 2 = 2 \\
    9 & 1 & 4 - 3 = 1 & 4 - 3 = 1 \\
    10 & 0 & 4 - 4 = 0 & 4 - 4 = 0
\end{array}
\]

Finally, the last point in the lemma is that for \( 1 \leq i \leq M \), \( \max\{T_t^{(r_i)} : n - r_i \leq t < n\} = T_n + \max\{T_t : 0 \leq t \leq r_i\} - T_{r_i} = T_n - i \). For this example, Table 3 gives all values for comparison, recalling that \( T^{(10)} = T \).

Definition. Call two walks \textbf{equivalent} if one is a rotation of the other. This defines an equivalence relation on the set of \( n \)-step walks.

**Theorem 4** (Hitting Time Theorem). If the joint probability distribution of \( (Y_1, \ldots, Y_n) \) is invariant under rotations, then

\[
P[T_n = k, T_t < k \text{ for all } t < n] = \frac{k}{n} \cdot P[T_n = k]
\]

Proof. Consider a single equivalence class, and let \( x = (x_1, \ldots, x_n) \) be the sequence of increments of one member of the class. Every walk in the class ends at the same height \( k \), and the class contains either \( n \) walks, or \( n/p \) walks for some divisor \( p \) of \( n \) if \( (x_1, \ldots, x_n) \) happens to be periodic. In either case, the lemma implies that if \( k \geq 1 \) and the joint probability distribution
of $Y = (Y_1, \ldots, Y_n)$ is invariant under rotations, then

\[
P[T_n = k, T_t < k \text{ for all } t < n, \text{ with } Y \text{ equivalent to } x] = \frac{k}{n} \cdot P[T_n = k, Y \text{ equivalent to } x].
\]

Choose one sequence of increments $x$ from each equivalence class and sum over them to obtain the result. \hfill \square

Remark. The condition that the joint probability distribution of $(Y_1, \ldots, Y_n)$ is invariant under rotations here is weaker than the traditional requirement that $Y_i$ be independent and identically distributed.

Remark. This proof is another version of the Cycle Lemma from the proof of the Ballot Theorem.

The Hitting Time Theorem Under Exchangeability

Definition. A random vector $(Y_1, \ldots, Y_n)$ is exchangeable if the joint probability distribution of $(Y_1, \ldots, Y_n)$ is invariant for any permutation of $(Y_1, \ldots, Y_n)$.

Theorem 5 (Hitting Time Theorem). If the joint probability distribution of $(Y_1, \ldots, Y_n)$ is exchangeable, then

\[
P[T_n = k, T_t < k, \text{ for all } t < n] = \frac{k}{n} \cdot P[T_n = k].
\]

Remark. The proof is not given here, see \cite{5}. The idea of the proof is that the joint probability distribution of $(Y_1, \ldots, Y_n)$ conditioned on $\sum_{i=1}^{n} Y_i = k$ is still exchangeable. This implies the conditional expectation $E[Y_i | T_n = 0]$ is independent of $i$. Then the conditional expectation step still holds, allowing the completion of the induction argument.

Sources

The history and background is adapted from \cite{5}. The proof for i. i. d. random variables is adapted from \cite{5}. The proof under rotation invariance is adapted from the article by Kager, \cite{2}. The comments about exchangeability are adapted from \cite{5}. 
Algorithms, Scripts, Simulations

Algorithm

Enumerating all walks

Set the number $a$ of upsteps and the number $b$ of downsteps. Then the total number of steps is the length $l = a + b$. There are then $\binom{a+b}{a-b}$ places to have the downsteps. In a sequence of nested loops systematically place the downsteps for $B$ in an $(a+b) \times \binom{a+b}{a-b}$ matrix of ones. Then cumulatively sum this matrix down the columns to get all possible walks and save the matrix for later plotting or analysis. Note that this algorithm does not scale well because of the inherent growth of the binomial coefficients.

Simulating the Lucky Derby Example

Set all parameters for the simulations, using a sufficient length of the race for most of the fastest horses to complete the race. Simulate all races in a vectorized manner, using the usual technique of creating uniformly distributed random variates, comparing row-wise to the Bernoulli probability, and cumulatively summing. Then using vectorized match functions, extract the index of the winning horse (breaking ties with minimum index), and then counting the number of times each horse wins with the histogram counter.

Computing the Win Probability in the Lucky Derby Example

Using a normal approximation to the first hitting time distribution for a Bernoulli random walk, construct the probability density for the 20th walk winning by multiplying the density and the survival functions, assuming independence. Then using different numerical quadrature routines, compute the approximate total probability of the 20th walker winning.
**Scripts**

**R script for exact probability calculation.**

```r
speeds <- seq(from=0.52, to=0.90, by=0.02)
nRaces <- 100000
nHorses <- 20
lengthRace <- 600
distance <- 200

randVals <- array( runif(nHorses * lengthRace * nRaces),
c(nHorses,
lengthRace, nRaces))
stepsSim <- 2*(randVals <= speeds) - 1
racesSim <- apply(stepsSim, c(1,3), cumsum)
whenFinish <- apply(racesSim, c(1,3), (function(x)
match(distance, x)))
winnerTime <- apply(whenFinish, 2, function(x) min(which(!is.na(x))))
winner <- whenFinish[cbind(winnerTime, 1:nRaces)]
hist(winner, breaks=seq(from=0.5, to=nHorses+0.5, by=1), plot=FALSE)
```

**R script for exact probability calculation.**

```r
P <- function(n, k, p) { (k/n) * dbinom((n+k), 2*n, p) }
speeds <- seq(from=0.52, to=0.90, by=0.02)
dP <- matrix(0, 20, 10001)
for (i in seq_along(speeds)) dP[i,] <- P(100:10100, 100, speeds[i])
pP <- t(apply(dP, 1, cumsum))
qP <- 1 - pP
lqP <- log(qP)
lqPprod <- exp( log(dP[20,]) + apply(lqP[1:19,], 2, sum))
cat("Probability horse 20 wins: ", sum(lqPprod), "\n")
```

**R script for normal approximation probability calculation.**

```r
distance <- 200
```
p <- seq(0.52, 0.90, length=20)
mu <- distance/(2*p-1)
sigma <- sqrt((4*distance*p*(1-p))/((2*p-1)^3))

h20 <- function(x) {
  (1/sigma[20]) * 
  exp( dnorm((x-mu[20])/sigma[20], 
    log=TRUE 
  ) + 
    sum( pnorm((x - mu)/sigma, 
      lower.tail=FALSE, log.p = TRUE 
    ) 
  ) - 
    pnorm( (x - mu[20])/sigma[20], 
      lower.tail=FALSE, log.p = TRUE 
    ) 
}

win20Integrate <- integrate(h20, 
  mu[20]-3*sigma[20], 
  mu[20]+3*sigma[20])
cat("integrate: ", win20Integrate$value, "\n")

library("pracma")
win20Quad <- quad(h20, 
  mu[20]-3*sigma[20], 
  mu[20]+3*sigma[20])
cat("quad: ", win20Quad, "\n")

win20QuadL <- quadl(h20, 
  mu[20]-3*sigma[20], 
  mu[20]+3*sigma[20])
cat("quadl: ", win20QuadL, "\n")

Octave  
Octave script for plotting walks.

k = 1;
a = 4;
b = 2;
steps = ones(a+b, nchoosek(a+b,a-b));
# Systematically place the downsteps for B
# using a-b nested loops, here a-b = 2 loops

c = 0;
for i = 1:(a+b)-1
  for j= (i+1):(a+b)
    c = c + 1;
    steps(i,c) = -k;
    steps(j,c) = -k;
  endfor;
endfor;

walk = cumsum(steps);

# Put the walk in a matrix whose
# first column is x-coordinates
# and first row is 0 to start at origin
plot_walk = zeros((a+b)+1, nchoosek(a+b,a-b) + 1);
plot_walk(2:(a+b)+1, 2:nchoosek(a+b,a-b) + 1) = walk;
plot_walk(:,1) = 0:(a+b);

save plot_walk.dat plot_walk

SciPy

```python
import scipy

distance = 200
p = scipy.linspace(0.52, 0.90, 20)
mu = distance / (2 * p -1)
sigma = scipy.sqrt((4 * distance * p * (1-p)) / ((2 * p - 1) ** 3))

from scipy.stats import norm

def h20(x):
    p = (1 / sigma[20 - 1]) *
        scipy.exp(scipy.stats.norm.logpdf((x-mu[20-1]) / 
       sigma[20-1]) +
        scipy.sum(scipy.stats.norm.logsf((x - mu) / 
       sigma ))) -
```
scipy.stats.norm.logsf((x - mu[20 - 1]) /
    sigma[20 - 1])
)
)
return p

import scipy.integrate as integrate

win20Quad = integrate.quad(h20,
    mu[20 - 1] - 3 * sigma[20 - 1],
    mu[20 - 1] + 3 * sigma[20 - 1])
print("quad", win20Quad[0])

win20Romberg = integrate.romberg(h20,
    mu[20 - 1] - 3 * sigma[20 - 1],
    mu[20 - 1] + 3 * sigma[20 - 1])
print("Romberg", win20Quad[0])


Problems to Work for Understanding

1. Use the Hitting Time Theorem to prove the Ballot Theorem.

2. Using the Reflection Principle, prove the probability that an $n$-step random walk taking independent and identically distributed steps $\pm 1$ with probability $1/2$ stays positive after time 0, given that it ends at height $k > 0$, is $k/n$.

3. Write a paragraph explaining in detail the differences and connections among the Hitting Time Theorem, the Positive Walks Theorem, and
the Gambler’s Ruin. Provide specific example using numerical values for the parameters.

4. Consider the case of a 7 step walk with \( Y_i = \pm 1 \) with 4 steps +1 and 3 steps −1. Make a chart of all possible walks, counting those that satisfy the Hitting Time Theorem.

5. Show that

\[
\mathbb{P}[h_i > h_j, i \neq j] \approx \int_{-\infty}^{\infty} \mathbb{P}[h_j = x] \prod_{i \neq j} \mathbb{P}[h_i > x] \, dx
\]

\[
= \int_{-\infty}^{\infty} \frac{1}{\sigma_j} \phi \left( \frac{x - \mu_j}{\sigma_j} \right) \prod_{i \neq j} \left( 1 - \Phi \left( \left( \frac{x - \mu_i}{\sigma_i} \right) \right) \right) \, dx.
\]

The calculation uses the usual notation: \( \phi \) is the standard normal probability density (pdf), \( \Phi \) is the cumulative distribution (cdf) of the standard normal distribution, and \( \mu_k \) and \( \sigma_k \) are the means and standard deviations.

6. Show that

\[
\sum_{n=k}^{\infty} P(2n; 2k, p) = 1 \quad \text{(sums to 1)}
\]

\[
\sum_{n=k}^{\infty} 2n P(2n; 2k, p) = \frac{2k}{2^q - 1} \quad \text{(computing the mean)}
\]

\[
\sum_{n=k}^{\infty} \left( 2n - \frac{2k}{2^q - 1} \right)^2 P(2n; 2k, p) = \frac{8np(1-p)}{(2p-1)^3} \quad \text{(computing the variance)}
\]

7. Modify the script using the exact probability distribution to calculate the probability of horse 19, 18, …, winning. At what stage do the calculations become unfeasible?

8. Modify the approximate scripts for the probability of winning the horse race to calculate the probability of horses 18, 17, … winning. At what stage do the calculations become unfeasible? How do the calculations compare to the exact probabilities?
9. If available, modify the scripts for the probability of winning the horse race to calculate the probability using either different numerical integration algorithms or infinite intervals or both. How do the results compare to the results in the horse race example?

10. Modify the scripts to display all walks with 4 steps +1 and 3 steps −1 to illustrate the Hitting Time Theorem.

11. Consider

\[ P[T_n = k, T_t < k, \text{ for all } t < n] = \frac{k}{n} \cdot P[T_n = k]. \]

Provide a detailed argument that when \( n = 1 \), both sides are 0 when \( k > 1 \) and \( k = 0 \), and are equal to \( P[Y_1 = 1] \) when \( k = 1 \). This is the base case of the induction.

12. (a) Give an example of a set of random variables with joint probability distribution that is exchangeable, but the set of random variables violates the condition “independent and identically distributed”. Keep the example as small and simple as possible.

(b) Give an example of a set of random variables with joint probability distribution that is invariant under rotations, but the set of random variables violates the condition “independent and identically distributed”. Keep the example as small and simple as possible.

(c) Give an example of a set of random variables with joint probability distribution that is exchangeable, but the set of random variables violates the condition of invariant under rotations. Keep the example as small and simple as possible.
Reading Suggestion:

References


Outside Readings and Links:

1.

2.

3.

4.

I check all the information on each page for correctness and typographical errors. Nevertheless, some errors may occur and I would be grateful if you would
alert me to such errors. I make every reasonable effort to present current and accurate information for public use, however I do not guarantee the accuracy or timeliness of information on this website. Your use of the information from this website is strictly voluntary and at your risk.

I have checked the links to external sites for usefulness. Links to external websites are provided as a convenience. I do not endorse, control, monitor, or guarantee the information contained in any external website. I don’t guarantee that the links are active at all times. Use the links here with the same caution as you would all information on the Internet. This website reflects the thoughts, interests and opinions of its author. They do not explicitly represent official positions or policies of my employer.

Information on this website is subject to change without notice.

Steve Dunbar’s Home Page, http://www.math.unl.edu/~sdunbar1
Email to Steve Dunbar, sdunbar1 at unl dot edu
Last modified: Processed from LATEX source on February 5, 2019