Still More Fun Indicator Random Variables

CS255

Chris Pollett

Feb. 6, 2006.

Outline

- Finishing up the Birthday Paradox
- Balls and Bins
- Streaks
- The On-Line Hiring Problem

Finishing up the Birthday Paradox

- Last day we used probabilities to answer the question how many people need to be in the same room for the odds of two people to share a birthday to exceed 50%?
- Today, we analyze the problem in terms of indicator variables.
- As indicator variables provide a convenient way to convert from probabilities to expectations, we will ask how many people need to be in the room before the expected number of shared birthdays exceeds 1?

Birthday Paradox continued

X_{ij} = I{person i and person j have the same birthday}

= 1 if person i and person j have the same birthday;

= 0 otherwise.

$E[X_{ij}]$ = Pr{person i and person j have the same birthday} = 1/n

Let X= $\sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij}$

So E[X] = $E\left[\sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij} \right]$ Using linearity of expectations.

$$= \sum_{i=1}^{k} \sum_{j=i+1}^{k} E[X_{i,j}]$$

$$= \binom{k}{2} \frac{1}{n} = \frac{k(k-1)}{2n}$$

When n=365, this is >1 if k=28

Balls and Bins

- Consider the process of tossing identical balls into *b* bins.
- Tosses will be assumed to be independent and any ball is equally likely to end up in any bin.
- The odds of ending up in a particular bin are thus $1/b$.
- Successfully, landing in a given bin can be viewed as a so-called Bernoulli trial. A **Bernoulli trial** is an experiment with two possible outcomes.
- Balls and bins arguments are useful for modeling a variety of processes in computer science such as hashing.

More Balls and Bins

- We can ask a variety of question about the ball tossing process:
  - *How many balls fall in a given bin*?
    - Since ball tossing into a given bin is a Bernoulli trial, the odds of *k* successes given *n* tosses follows th**e binomial distribution** $b(k;n, x_{in\_bin})$, where x_{in_bin}=Pr{ball in bin} = 1/*b*. $x_{not\_in\_bin}$= Pr{ball not in bin}= 1-1/*b*.
    - The sample space for the binomial distribution is {1,2,…} the possible values for k.
    - The probability of a given event can be determined by considering $(x_{in\_bin} + x_{not\_in\_bin})^n = (1/b + (1-1/b))^n = 1$. Expanding this, the term

$$\binom{n}{k} x_{in\_bin}^{k} \cdot x_{not\_in\_bin}^{n-k} = \binom{n}{k} \frac{1}{b}^{k} \cdot \left(1 - \frac{1}{b}\right)^{n-k}$$

represents the probability of getting k balls after n tosses.

$$(x_{in\_bin} + x_{not\_in\_bin})^{n-1}$$

    - Let *X* be a random variable whose value is the number of tosses to fall in the bin. Then E[*X*]=

$$\sum_{k=1}^{n} k \cdot \binom{n}{k} \left(\frac{1}{b}\right)^{k} \cdot \left(1-\frac{1}{b}\right)^{n-k} = n\left(\frac{1}{b}\right) \sum_{k=1}^{n} \binom{n-1}{k-1} \left(\frac{1}{b}\right)^{k-1} \cdot \left(1-\frac{1}{b}\right)^{n-k} = \frac{n}{b} \sum_{k=0}^{n-1} \binom{n-1}{k} \left(\frac{1}{b}\right)^{k} \cdot \left(1-\frac{1}{b}\right)^{(n-1)-k} = \frac{n}{b}$$

Yet More Balls and Bins

– *How many balls must one toss on average, until a given bin contains a ball?*

- We can have as our sample space {1, 2, ..} where an event *k* is supposed to indicate that the first time one got into bin was the *k*th toss. If *X* is the number of trials to succeed Pr{*X*=*k*} = (1-1/*b*)$^{k-1}$*1/*b*. This is a **geometric distribution**.

$$
\begin{aligned}
E[X] &= \sum_{k=1}^{\infty} k(1-1/b)^{k-1}\cdot(1/b) \\
&= (1/b)\sum_{k=1}^{\infty} k(1-1/b)^{k-1}, \text{ let } q = 1-1/b \\
&= (1/b)\frac{d}{dq}\sum_{k=0}^{\infty} q^k = (1/b)\frac{d}{dq}\frac{1}{1-q} = (1/b)\frac{1}{(1-q)^2} \\
&= (1/b)\frac{1}{(1-(1-1/b))^2} = b^2/b = b
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k=0}^{\infty} q^k &= \lim_{n\to\infty}\sum_{k=0}^{n} q^k \\
&= \lim_{n\to\infty}\frac{1-q^n}{1-q} \\
&= \frac{1}{1-q}
\end{aligned}
$$

As $(1-q)\sum_{k=0}^{n} q^k = 1-q^n$

Even More Balls and Bins

– *How many balls must one toss on average, until every bin contains at least one ball?*
  - Call a toss into an empty bin a hit.
  - We want to know the expected number *n* of tosses to get *b* hits.
  - Can partition the *n* tosses into stages where the ith stage is the number of tosses after the (*i*-1)st hit until the *i*th hit. The first stage is thus just the first toss.
  - The probability of there being a hit for a given toss in stage *i* is (*b*-*i*+1)/*b*
  - Let n_i denote the number of tosses in stage *i*. So the number of tosses to get *b* hits is $n = \sum_{i=1}^{b} n_i$
  - Each random variable n_i follows a geometric distribution with probability of success (*b*-*i*+1)/*b*. Using the same kind of calculation as the last slide $E[n_i]=b/(b-i+1)$.
  - Using linearity of expectation: E[n] = $$E[\sum_{i=1}^{b} n_i] = \sum_{i=1}^{b} E[n_i] = \sum_{i=1}^{b} \frac{b}{b-i+1} = b\sum_{i=1}^{b} \frac{1}{i} = b(\ln b + O(1))$$
  - This problem is also called the **coupon collector's problem**.

This sum can be bounded by the integral of 1/i.

Streaks

- Suppose you flip a fair coin n times. What is the longest streak of consecutive heads you expect to see?
  - The book gives a nice argument which we will skip that the answer is $\Theta(\log n)$ which we'll skip.

The On-Line Hiring Problem

- Suppose we don't want to interview everyone in the hiring problem to find the best candidate.
- Suppose further we only want to hire once.
- So we follow the following algorithm:

```
On-Line-Maximum(k,n)
bestscore = -infinity
for i <-- 1 to k
    do if score(i) > bestscore
            then bestscore <-- score(i)
for i <-- k+1 to n
    do if score(i) > bestscore then return i
return n
```

- We want to determine the odds of getting the best candidate as a function of *k*.

More on the On-line Hiring Problem

- Let $M(j) = \max_{1<=i<=j}\{score(i)\}$
- Let S be the event of choosing the best qualified applicant.
- Let S_i be the event the the ith applicant is the best qualified. So the S_i's are disjoint events.
- Note we never succeed in choosing the best candidate if i=1,..,k. So $\Pr\{S_i\}=0$ for these i.

$$Pr\{S\} = \sum_{i=k+1}^{n} Pr\{S_i\}$$

- To succeed at the best qualified applicant must be in location i. Call this event $B_{i.}$ The algorithm also can't select any candidate from among k+1 through i-1, which happens only if, for j in this range $score(j)$ < *bestscore*. So $score(k+1),..,score(i\text{-}1) < M(k)$. Let O_i denote this second event.

Still More on the On-Line hiring Problem

- O_i only depends on the relative order of the values in the positions 1 through *i*-1.
- B_i depends only on whether the value at position i is greater than all other positions.
- This turn out to imply B_i and O_i are independent.
- So $\Pr\{S_i\} = \Pr\{B_i \cap O_i\} = \Pr\{B_i\}\Pr\{O_i\} = 1/n * k/(i-1)$.
  - $\Pr\{B_i\} = 1/n$ since the best value is equally likely to be in any position
  - $\Pr\{O_i\} = k/i-1$ as the maximum value in position 1.. *i*-1 is equally likely to be in any position. So the odds its in the among *k*, is *k*/*i*-1.
- Thus we Pr{S} = $\sum_{i=k+1}^{n} Pr\{S_i\} = \sum_{i=k+1}^{n} \frac{k}{n(i-1)} = \frac{k}{n}\sum_{i=k+1}^{n}\frac{1}{i-1} = \frac{k}{n}\sum_{i=k}^{n-1}\frac{1}{i}$
- Bounding the sum in terms of the integral for 1/i. We get

$$\frac{k}{n}(\ln n - \ln k) \leq Pr\{S\} \leq \frac{k}{n}(\ln(n-1) - \ln(k-1))$$

- Differentiating with respect to *k* and setting to 0 one can show this is maximized when *k*= *n*/*e*.