Cases studies for CST IA Probability

R.J. Gibbens

Computer Laboratory
University of Cambridge

February 2012

Overview

Two short cases studies where probability has played a pivitol role:

1. Birthday problem ("birthday attack")
   - cryptographic attacks
2. Probabilistic classification ("naive Bayes classifier")
   - email spam filtering

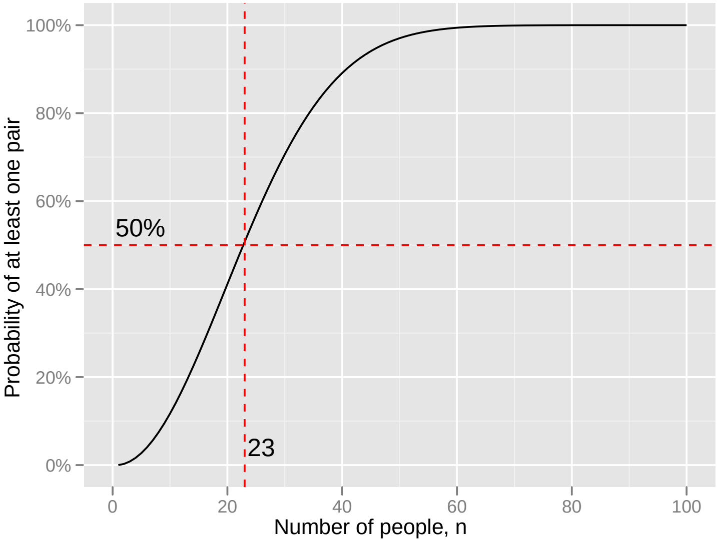
The birthday problem

Consider the problem of computing the probability, $p(n)$, that in a party of n people at least two people share a birthday (that is, the same day and month but not necessarily same year).
It is easiest to first work out $1 - p(n) = q(n)$, say,
where $q(n) = \mathbb{P}(\text{none of the } n \text{ people share a birthday})$ then

$$\begin{aligned} q(n) &= \left(\frac{364}{365}\right)\left(\frac{363}{365}\right)\cdots\left(\frac{365-n+1}{365}\right) \\ &= \left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\cdots\left(1-\frac{n-1}{365}\right) \\ &= \prod_{k=1}^{n-1}\left(1-\frac{k}{365}\right). \end{aligned}$$

Surprisingly, $n = 23$ people suffice to make $p(n)$ greater than 50%.

Graph of $p(n)$



100%
80%
60%
40%
20%
0%
50%
23
0
20
40
60
80
100
Probability of at least one pair
Number of people, n

Assumptions

We should record some of our assumptions behind the calculation of $p(n)$.

1. Ignore leap days (29 Feb)
2. Each birthday is equally likely
3. People are selected independently and without regard to their birthday to attend the party (ignore twins, etc)

Examples: coincidences on the football field

Ian Stewart writing in Scientific American illustrates the birthday problem with an interesting example. In a football match there are 23 people (two teams of 11 plus the referee) and on 19 April 1997 out of 10 UK Premier Division games there were 6 games with birthday coincidences and 4 games without.

Examples: cryptographic hash functions

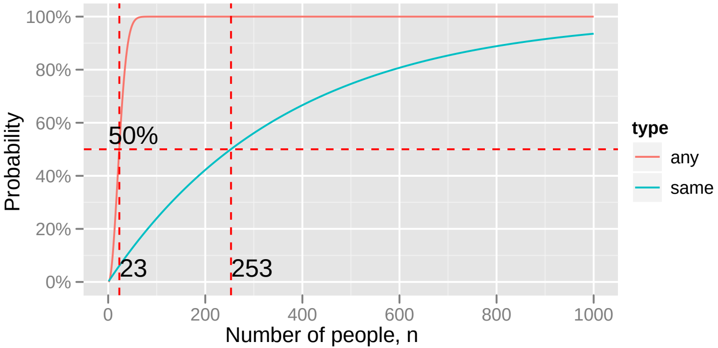
A hash function $y = f(x)$ used in cryptographic applications is usually required to have the following two properties (amongst others):

1. one-way function: computationally intractible to find an x given y.
2. collision-resistant: computationally intractible to find distinct x_1 and x_2 such that $f(x_1) = f(x_2)$.

Probability of same birthday as you

Note that in calculating $p(n)$ we are not specifying which birthday (for example, your own) matches. For the case of finding a match to your own birthday amongst a party of n other people we would calculate

$$1 - \left(\frac{364}{365}\right)^n .$$



General birthday problem

Suppose we have a random sample $X_1, X_2, \ldots, X_n$ of size n where X_i are IID with $X_i \sim U(1, d)$ and let $p(n, d)$ be the probability that there are at least two outcomes that coincide. Then

$$p(n,d) = \begin{cases} 1 - \prod_{k=1}^{n-1}\left(1 - \frac{k}{d}\right) & n \leq d \\ 1 & n > d\,. \end{cases}$$

The usual birthday problem is the special case when $d = 365$.

Approximations

One useful approximation is to note that for $x \ll 1$ then $1 - x \approx e^{-x}$. Hence for $n \leq d$

$$
\begin{aligned}
p(n,d) &= 1 - \prod_{k=1}^{n-1}\left(1 - \frac{k}{d}\right) \\
&\approx 1 - \prod_{k=1}^{n-1} e^{-\frac{k}{d}} \\
&= 1 - e^{-(\sum_{k=1}^{n-1} k)/d} \\
&= 1 - e^{-n(n-1)/(2d)} .
\end{aligned}
$$

We can further approximate the last expression as

$$
p(n,d) \approx 1 - e^{-n^2/(2d)} .
$$

Inverse birthday problem

Using the last approximation

$$p(n,d) \approx 1 - e^{-n^2/(2d)}$$

we can invert the birthday problem to find $n = n(p,d)$, say, such that $p(n,d) \approx p$ so then

$$\begin{aligned} e^{-n(p,d)^2/(2d)} &\approx 1 - p \\ -\frac{n(p,d)^2}{2d} &\approx \log(1-p) \\ n(p,d)^2 &\approx 2d \log\left(\frac{1}{1-p}\right) \\ n(p,d) &\approx \sqrt{2d \log\left(\frac{1}{1-p}\right)}. \end{aligned}$$

In the special case of $d = 365$ and $p = 1/2$ this gives the approximation $n(0.5, 365) \approx \sqrt{2 \times 365 \times \log(2)} \approx 22.49$.

Expected waiting times for a collision/match

Let W_d be the random variable specifiying the number of iterations when you choose one of d values independently and uniformly at random (with replacement) and stop when any value is selected a second time (that is, a "collision" or "match" occurs).
It is possible to show that

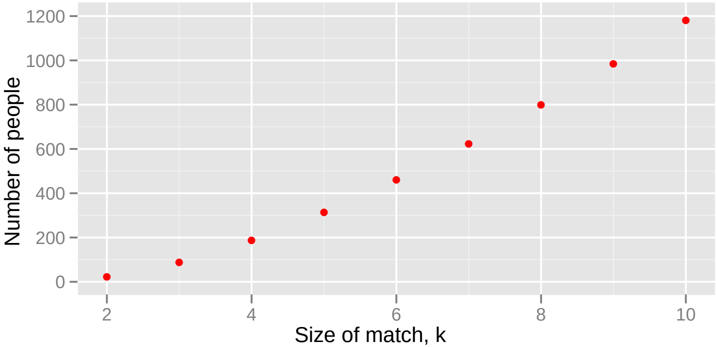
$$\mathbb{E}(W_d) \approx \sqrt{\frac{\pi d}{2}}\,.$$

Thus in the special case of the birthday problem where $d = 365$ we have that $\mathbb{E}(W_{365}) \approx \sqrt{\frac{\pi \times 365}{2}} \approx 23.94$.
In the case that we have a cryptographic hash function with 160-bit outputs ($d = 2^{160}$) then $\mathbb{E}(W_{2^{160}}) \approx 1.25 \times 2^{80}$. This level of reduction leads to so-called "birthday attacks". (See the IB course Security I for further details.)

Further results

Persi Diaconis and Frederick Mosteller give results on the minimum number n_k required to give a probability greater than $1/2$ of k or more matches with $d = 365$ possible choices.

k	2	3	4	5	6	7	8	9	10
n_k	23	88	187	313	460	623	798	985	1181



Email spam filtering

Suppose that an email falls into exactly one of two classes (spam or ham) and that various features $F_1, F_2, \ldots, F_n$ of an email message can be measured. Such features could be the presence or absence of particular words or groups of words, etc, etc.
We would like to determine $\mathbb{P}(C \mid F_1, F_2, \ldots, F_n)$ the probability that an email message falls into a class C given the measured features $F_1, F_2, \ldots, F_n$. We can use Bayes' theorem to help us.

Bayes' theorem for emails

We have that

$$\mathbb{P}(C \mid F_1, F_2, \ldots, F_n) = \frac{\mathbb{P}(C)\mathbb{P}(F_1, F_2, \ldots, F_n \mid C)}{\mathbb{P}(F_1, F_2, \ldots, F_n)}$$

which can be expressed in words as

$$\text{posterior probability} = \frac{\text{prior probability} \times \text{likelihood}}{\text{evidence}}.$$

Naive Bayes classifier

In the naive Bayes classifier we make the assumption of independence across features. So that

$$\mathbb{P}(F_1, F_2, \ldots, F_n \mid C) = \prod_{i=1}^{n} \mathbb{P}(F_i \mid C)$$

and then

$$\mathbb{P}(C \mid F_1, F_2, \ldots, F_n) \propto \mathbb{P}(C) \prod_{i=1}^{n} \mathbb{P}(F_i \mid C).$$

Decision rule for naive Bayes classifier

We then use the decision rule to classify an email with observed features $F_1, F_2, \ldots, F_n$ as spam if

$$\mathbb{P}(C = \text{spam}) \prod_{i=1}^{n} \mathbb{P}(F_i \mid C = \text{spam}) > \mathbb{P}(C = \text{ham}) \prod_{i=1}^{n} \mathbb{P}(F_i \mid C = \text{ham}) .$$

This decision rule is known as the maximum a posteriori (MAP) rule.

Surveys and a training set of manually classified emails are needed to estimate the values of $\mathbb{P}(C)$ and $\mathbb{P}(F_i \mid C)$.

References

Ian Stewart
What a coincidence!
Mathematical Recreations, Scientific American, Jun 1998, 95–96.

Persi Diaconis and Frederick Mosteller
Methods for studying coincidences.
Journal of American Statistical Association, Vol 84, No 408, Dec 1989, 853–861.