5. independence

Defn: Two events E and F are *independent* if

$P(EF) = P(E)\,P(F)$

If $P(F)>0$, this is equivalent to: $P(E|F) = P(E)$ (proof below)

Otherwise, they are called *dependent*

Roll two dice, yielding values D_1 and D_2

1) $E = \{ D_1 = 1 \}$

   $F = \{ D_2 = 1 \}$

   $P(E) = 1/6,\ P(F) = 1/6,\ P(EF) = 1/36$

   $P(EF) = P(E) \cdot P(F) \Rightarrow$ E and F *independent*

   Intuitive; the two dice are not physically coupled

2) $G = \{D_1 + D_2 = 5\} = \{(1,4),(2,3),(3,2),(4,1)\}$

   $P(E) = 1/6, P(G) = 4/36 = 1/9, P(EG) = 1/36$

   not independent!

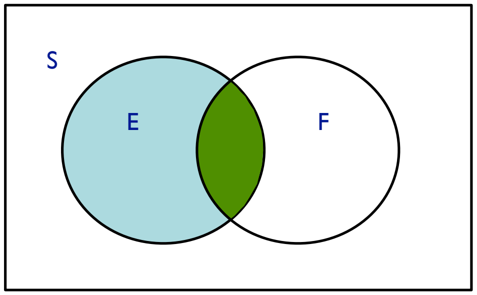
   E, G are *dependent* events

   The dice are still not physically coupled, but "$D_1 + D_2 = 5$" couples them mathematically: info about D_1 constrains D_2. (I.e., dependence/independence not always intuitively obvious; "use the definition, Luke.")

Theorem: E, F independent $\Rightarrow$ E, F^c independent

Proof: $P(EF^c) = P(E) - P(EF)$
$= P(E) - P(E)\,P(F)$
$= P(E)\,(1-P(F))$
$= P(E)\,P(F^c)$

$E = EF \cup EF^c$



$P(E) = P(EF) + P(EF^c)$

Theorem: if $P(E)>0$, $P(F)>0$, then
E, F independent $\Leftrightarrow P(E|F)=P(E) \Leftrightarrow P(F|E) = P(F)$

Proof: Note $P(EF) = P(E|F)\,P(F)$, *regardless of in/dep.*

Assume independent. Then

$P(E)P(F) = P(EF) = P(E|F)\,P(F) \Rightarrow P(E|F)=P(E)$ ($\div$ by $P(F)$)

Conversely, $P(E|F)=P(E) \Rightarrow P(E)P(F) = P(EF)$ ($\times$ by $P(F)$)

Two events E and F are *independent* if

$P(EF) = P(E)\,P(F)$

If $P(F)>0$, this is equivalent to: $P(E|F) = P(E)$

Otherwise, they are called *dependent*

Three events E, F, G are independent if

$P(EF) = P(E)\,P(F)$

$P(EG) = P(E)\,P(G)$ *and* $P(EFG) = P(E)\,P(F)\,P(G)$

$P(FG) = P(F)\,P(G)$

Example: Let X, Y be each {-1, 1} with equal prob

$E = \{X = 1\}, F = \{Y = 1\}, G = \{XY = 1\}$

$P(EF) = P(E)P(F), P(EG) = P(E)P(G), P(FG) = P(F)P(G),$

all 1/4 but *P(EFG) = 1/4 too*!!! (because $P(G|EF) = 1$)

In general, events $E_1, E_2, \ldots, E_n$ are independent if for *every subset* S of {1,2,…, n}, we have

$$P\left(\bigcap_{i\in S} E_i\right) = \prod_{i\in S} P(E_i)$$

(Sometimes this property holds only for small subsets S. E.g., E, F, G on the previous slide are *pairwise* independent, but not fully independent.)

Suppose a biased coin comes up heads with probability p, *independent* of other flips

P(n heads in n flips) = p^n

P(n tails in n flips) = $(1-p)^n$

P(exactly k heads in n flips) $= \binom{n}{k} p^k (1-p)^{n-k}$

Aside: note that the probability of *some* number of heads = $\sum_k \binom{n}{k} p^k (1-p)^{n-k} = (p + (1-p))^n = 1$ as it should, by the binomial theorem.

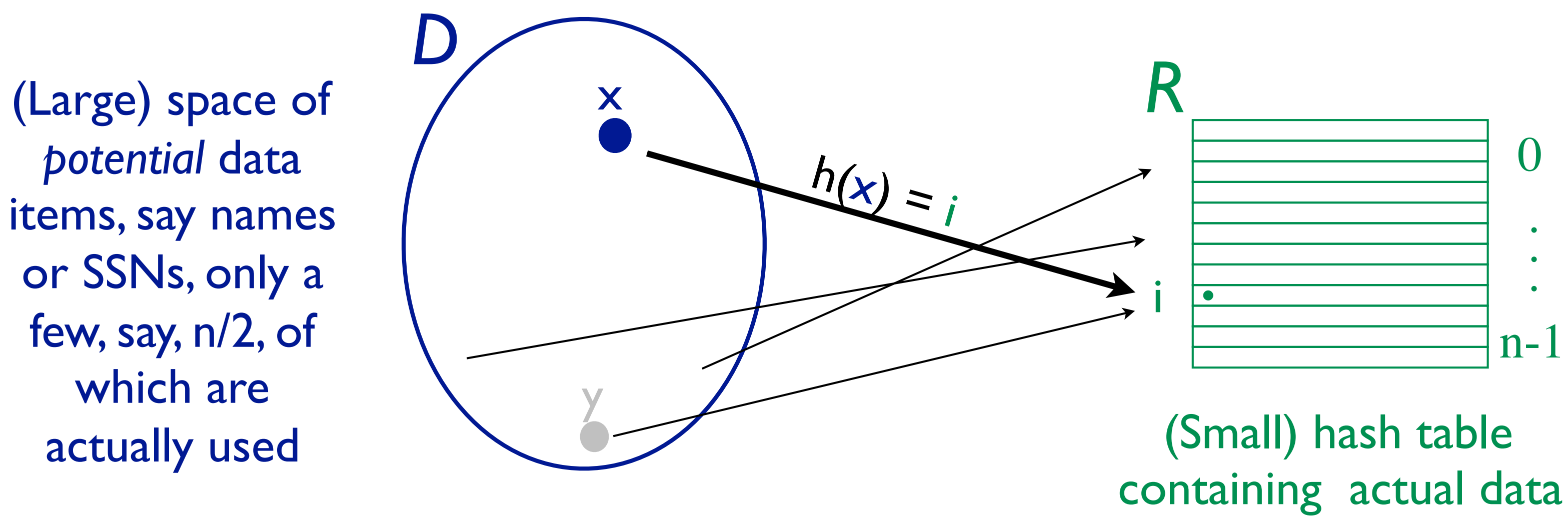
Suppose a biased coin comes up heads with probability p, *independent* of other flips

P(exactly k heads in n flips) $= \binom{n}{k} p^k (1-p)^{n-k}$

Note when $p=1/2$, this is the same result we would have gotten by considering n flips in the "equally likely outcomes" scenario. But $p \neq 1/2$ makes that inapplicable. Instead, the *independence* assumption allows us to conveniently assign a probability to each of the 2^n outcomes, e.g.:

$\Pr(\text{HHTHTTT}) = p^2(1-p)p(1-p)^3 = p^{\#H}(1-p)^{\#T}$

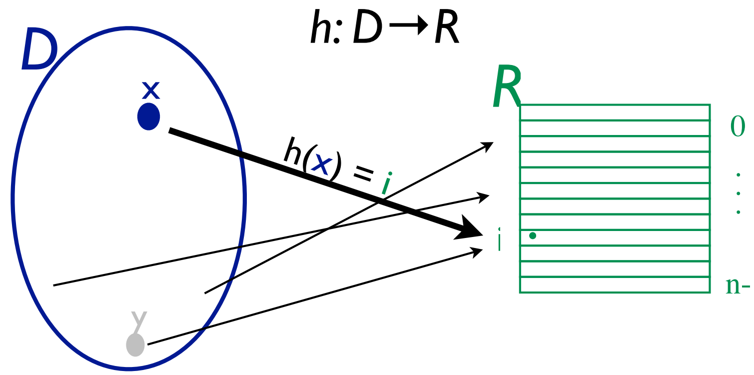
A data structure problem: fast access to small subset of data drawn from a large space.



A solution: hash function h:D→R crunches/scrambles names from large space D into small one R.

Example: if x is (or can be viewed as) an integer:

$$h(x) = x \bmod n$$



Scenario: Hash m≤n keys from D into size n hash table.

How well does it work?

Worst case: *All* collide in one bucket. (Perhaps too pessimistic?)

Best case: *No* collisions. (Perhaps too optimistic?)

Exact analysis: …? (Perhaps too complex?)

A middle ground: Probabilistic analysis.

Below, for simplicity, assume

- Keys drawn from D randomly, independently (with replacement)

- *h* maps equal numbers of domain points into each range bin, i.e., $|D| = k|R|$ for some integer k, and $|h^{-1}(i)| = k$ for all $0 \leq i \leq n-1$

Many possible questions; a few analyzed below m keys hashed into a table with n buckets

Each string hashed is an *independent* sample from D

E = at least one string hashed to first bucket

What is P(E) ?

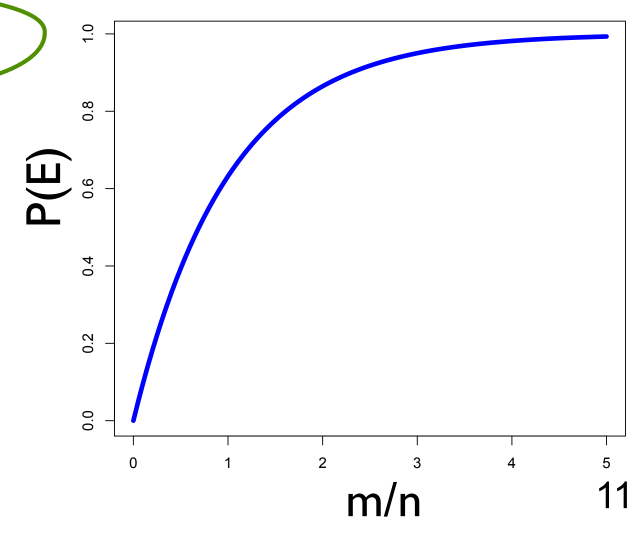
Solution:

F_i = string i *not* hashed into first bucket (i=1,2,…,m)

$P(F_i) = (n-1)/n$ for all i=1,2,…,m

Event $(F_1\ F_2\ \ldots\ F_m)$ = no strings hashed to first bucket

$$
\begin{aligned}
P(E) &= 1 - P(F_1\ F_2 \cdots F_m) \\
&\overset{\text{indp}}{=} 1 - P(F_1)\ P(F_2) \cdots P(F_m) \\
&= 1 - ((n-1)/n)^m \\
&= 1 - [((n-1)/n)^n]^{m/n} \\
&\approx 1 - \exp(-m/n)
\end{aligned}
$$



Let $|R| = n$, $D_0 \subseteq D$, $|D_0| = m$. A hash function $h{:}D \to R$ is *perfect* for D_0 if $h{:}D_0 \to R$ is injective (no collisions). How likely is that?

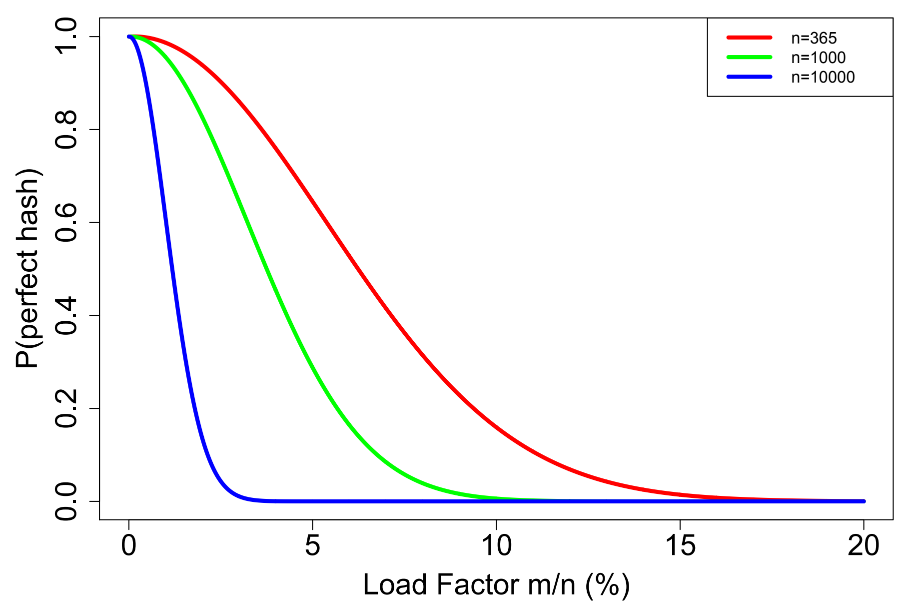
(i) Fix h; pick m elements of D_0 *independently* at random $\in D$

Again, suppose h maps $(1/n)^{th}$ of D to each element of R. This is like the birthday problem:

$$P(h \text{ is perfect for } D_0) = \frac{n}{n}\frac{n-1}{n}\cdots\frac{n-m+1}{n}$$

Except for very empty tables, a "perfect" hash is improbable

(Q: why less likely with larger n, fixed m/n?)



Let $|R| = n$, $D_0 \subseteq D$, $|D_0| = m$. A hash function $h:D \rightarrow R$ is *perfect* for D_0 if $h:D_0 \rightarrow R$ is injective (no collisions). How likely is that?

(ii) Fix D_0; pick $\underset{\sim}{h}$ at random (among all with constant $|h^{-1}(i)|$)

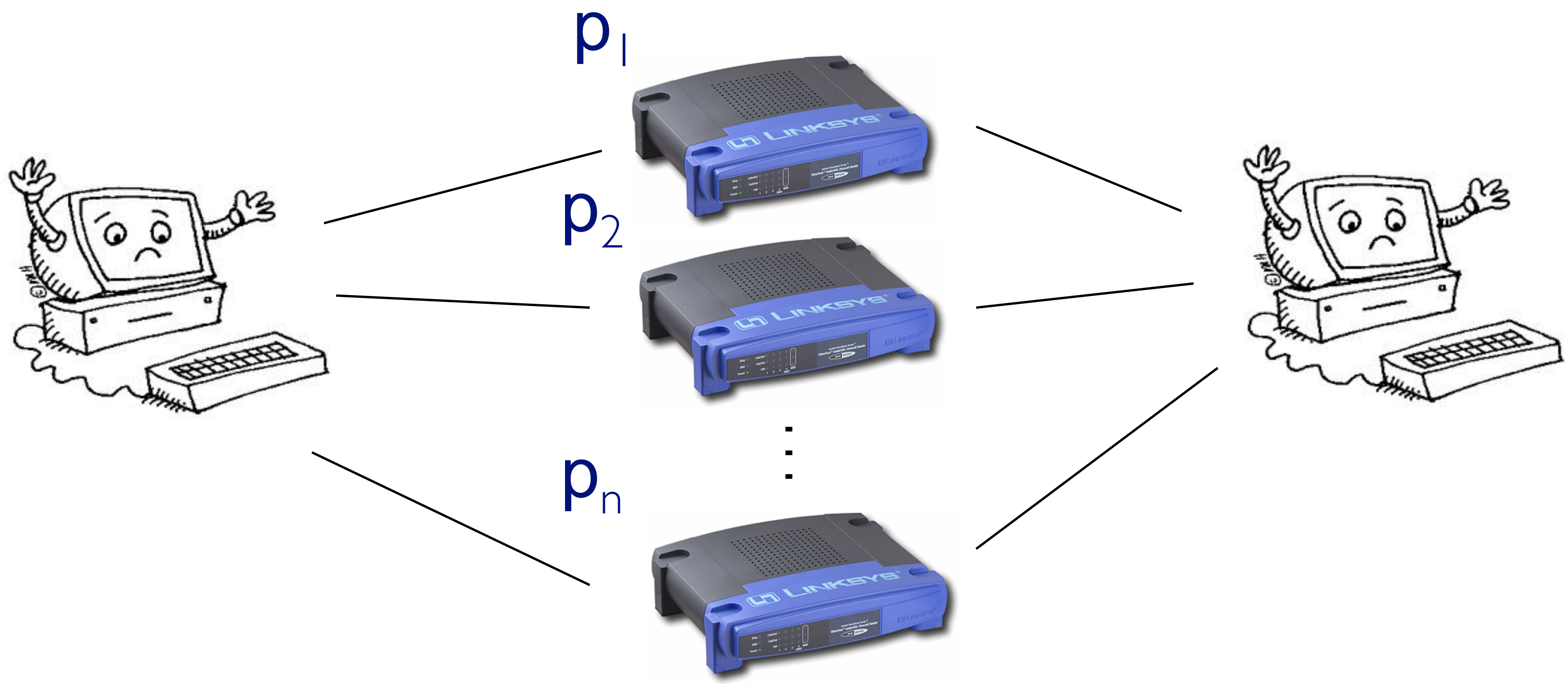
E.g., if $m = |D_0| = 23$ and $n = 365$, then there is ~50% chance that the first h you try is perfect for this *fixed* D_0. If it isn't, pick $h_{(2)}$, $h_{(3)}$, … With high probability, you'll quickly find a perfect one!

"Picking a random function h" is easier said than done, but, empirically, picking from a set of *parameterized* fns like

$$h_{a,b}(x) = (a \bullet x + b) \bmod n$$

where a, b are random 64-bit ints is a start.

Consider the following parallel network

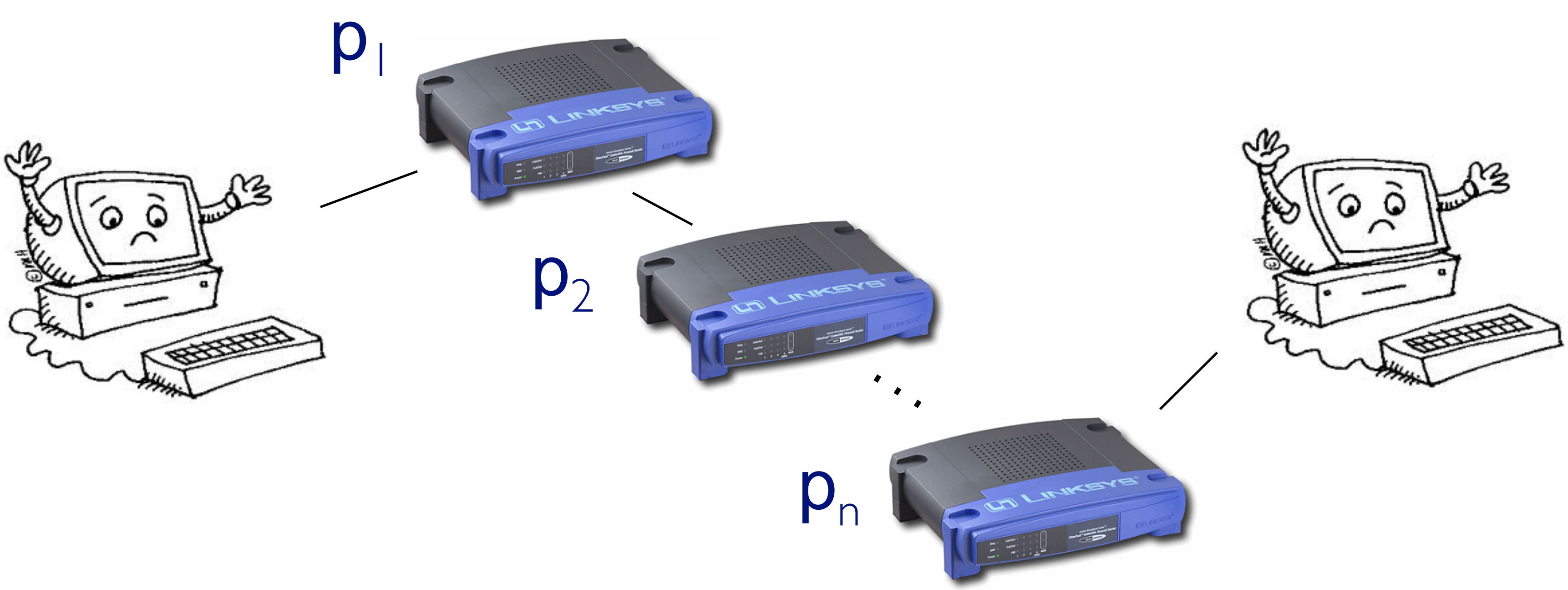


n routers, i^{th} has probability p_i of failing, independently

P(there is functional path) = 1 – P(all routers fail)

$$= 1 - p_1 p_2 \cdots p_n$$

Contrast: a *series* network



n routers, i^{th} has probability p_i of failing, independently

P(there is functional path) =

$$P(\textit{no} \text{ routers fail}) = (1 - p_1)(1 - p_2) \cdots (1 - p_n)$$

Recall: Two events E and F are independent if

$$P(EF) = P(E)\, P(F)$$

If E & F are independent, does that tell us anything about

$$P(EF|G),\ P(E|G),\ P(F|G),$$

when G is an arbitrary event? In particular, is

$$P(EF|G) = P(E|G)\, P(F|G)\ ?$$

In general, *no*.

Roll two 6-sided dice, yielding values D_1 and D_2

$E = \{ D_1 = 1 \}$

$F = \{ D_2 = 6 \}$

$G = \{ D_1 + D_2 = 7 \}$

E and F are independent

$P(E|G) = 1/6$

$P(F|G) = 1/6$, but

$P(EF|G) = 1/6$, *not* $1/36$

so E|G and F|G are not independent!

Definition:

Two events E and F are called *conditionally independent given G*, if

$P(EF|G) = P(E|G)\ P(F|G)$

Or, equivalently (assuming P(F)>0, P(G)>0),

$P(E|FG) = P(E|G)$

Example:

$\left.\begin{array}{l} E = \text{has lung cancer} \\ F = \text{carries matches} \end{array}\right\}$ non-independent (I think)

G = smokes cigarettes

conditioning can also remove **DEPENDENCE**

Randomly choose a day of the week

A = { It is not a Monday }

B = { It is a Saturday }

C = { It is the weekend }

A and B are dependent events

P(A) = 6/7, P(B) = 1/7, P(AB) = 1/7.

Now condition both A and B on C:

P(A|C) = 1, P(B|C) = ½, P(AB|C) = ½

P(AB|C) = P(A|C) P(B|C) ⇒ A|C and B|C independent

Dependent events can become independent by conditioning on additional information!

Another reason why conditioning is so useful

Events E & F are *independent* if

$P(EF) = P(E)\,P(F)$, or, equivalently $P(E|F) = P(E)$ (if $p(E)>0$)

More than 2 events are indp if, for *all subsets*, joint probability = product of separate event probabilities

Dependent means correlated, associated, (partially) predictive

Independence can greatly simplify calculations

For fixed G, conditioning on G gives a probability measure, $P(E|G)$

But "conditioning" and "independence" are orthogonal:

Events E & F that are (unconditionally) independent may become dependent when conditioned on G

Events that are (unconditionally) dependent may become independent when conditioned on G