

Computer science and discrete probability

- Probability space
- Independent and dependent events
- Conditional probability
- Bayes' Theorem
- Discrete random variables
- Expected values
- Random-number generation
- Simulation of chance
- Applications: games, average-case analysis, evolutionary computation



David Keil 1/03 1

Basic concepts

- *Probability of an event ($P(x)$):* The frequency of occurrence of event x , $0 \leq P(x) \leq 1$
- *Sample space:* All possible results of an experiment
- *Discrete probability* assumes finite sample space
- *Compound events:* sets of events

David Keil 1/03 2

Uniform probability space

- A probability space S is a set of possible outcomes of an experiment
- *Example:* S for a die throw is $\{1, 2, 3, 4, 5, 6\}$
- Let $|S| = n$ for probability space S
- Uniform probability function $P : S \rightarrow R$ is defined $P(x) = (1/n)$ for any x in S
- *Example:* Using fair die, $P(3) = 1/6$, because there are 6 possible events, all equally likely

David Keil 1/03 3

Theorems

Let S be a finite probability space,
 P a distribution, $A, B \in S$

1. **Monotonicity:** $A \subseteq B \rightarrow P(A) \leq P(B)$
Example: In cards, $P(\text{Heart}) \leq P(\text{Red})$

2. **Complementary event:**

$$P(\sim A) = 1 - P(A)$$

Example: $P(\heartsuit) = 1 - P(\spadesuit, \diamondsuit, \clubsuit)$ in cards
 (because $1/4 = 1 - 3/4$)

Proofs omitted – use intuition!

David Keil 1/03 4

More theorems

Let S be a finite probability space,
 P a distribution, $A, B \in S$ as before

3. *Additivity under disjunction:*

Given $A \cap B = \emptyset$, then

$$P(A \cup B) = P(A) + P(B)$$

Example: $P(\heartsuit \text{ or } \spadesuit) = P(\heartsuit) + P(\spadesuit) = 1/2$

4. *Partial additivity:*

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Example: $P(\heartsuit \cup \text{Queen}) = 1/4 + 1/13 - 1/52$

(the probability of this or that happening is the sum of the probabilities of each, discounting the chance of both)

David Keil 1/03 5

Conditional probability

- $P(A | B) = P(A \cap B) \div P(B)$
- *Interpretation:* The probability of event A , given event B , is the probability that both will occur, divided by the probability of B
- *Example:* You throw two coins. Given that the first is heads, what's the chance of two heads?

$$P(c_1 = c_2 = H | c_1 = H) =$$

$$P(\text{both are H and } c_1 = H) / P(c_1 = H)$$

$$= 1/4 / 1/2 = 1/2$$

David Keil 1/03 6

Independent events

- *Intuition:* Independent events can have no effect on each other or overlap with each other
- *Formally:* Events A and B are independent iff $P(A \cap B) = P(A)P(B)$
- *Example:* For draw of cards, $P(\heartsuit)$ is independent of $P(J \text{ or } Q \text{ or } K)$

David Keil 1/03 7

Bayes' Theorem (inverse probability)

- *Intuition:* given some knowledge of an object, and some statistics about the population containing the object, what else can we surmise about the object?
- *Example:* Suppose we know $\frac{2}{3}$ of the numbered cards in a pile are red, and $\frac{1}{4}$ of the JQK cards are red, and $\frac{3}{4}$ of all the cards are JQK. If a card randomly drawn is red, then by Bayes' Theorem we can calculate the probability that it is a J, Q, or K.

David Keil 1/03 8

Permutations

- *Definition:* Orderings of n objects taken k at a time, without repetition; there are $(n!)$ permutations for n objects
- *Example:* There are $5! = 120$ ways to order the letters A, B, C, D, E
- *k-permutations* ($P(n,k)$): Orderings of n objects taken k at a time; there are $(n! / (n - k)!)$ k -permutations of n objects
- *Example:* there are $P(6, 3) = 120$ different ways to throw a die such that only 1, 2, or 3 show

David Keil 1/03 9

Combinations

- *Definition:* Selections of n objects taken k at a time, without order or repetition;
 $C(n, k) = n! / (n - k)! k!$
- *Example:* There are $C(36, 6)$ ways to play the lottery where 6 numbers are chosen out of 36

David Keil 1/03 10

Probabilities of success

- Assume n trials in an experiment where the result may be success or failure, and the probability of success in each trial is p
- Each way of having k successes has probability $p^k(1 - p)^{n - k}$
- There are $C(n, k)$ ways to have k successes

David Keil 1/03 11

The binomial distribution

- Probability of k successes (the binomial distribution function):
 $B(n, k; p) = C(n, k) p^k (1 - p)^{n - k}$
- *Example:* The probability of having 4 or more heads in 6 coin throws is

$$\sum_{k=4}^6 B(6, k; 0.5)$$

David Keil 1/03 12

Discrete random variables

- *Definition:* A function from a finite sample space to a finite set of outcomes
- *Example:* Let random variable χ (“Chi”) be the sum of scores for two dice. Then χ takes the value 1 in no case, 2 in 1 case, 3 in 2 cases $\{(1,2), (2,1)\}$, etc.

David Keil 1/03 13

Random distributions

- Probability that a random variable takes a given value is the probability of the set of outcomes where that holds,
$$P(\chi = k) = P(\{ s \in S \mid \chi(s) = k \})$$
- Probability distribution function, $f_{\chi}(x)$, maps from outcomes to their probabilities
- Examples:
 - Uniform distribution (flat graph)
 - Gaussian distribution (“normal curve”)

David Keil 1/03 14

Expected values

- For n equally likely outcomes, where $f(k)$ is the value of the k th outcome, the expected outcome is
$$\sum_{k=1}^n f(k)$$
- *Examples:*
 - In coin toss, expected value is (0.5)
 - In die throw, expected outcome is
$$(1 + 2 + 3 + 4 + 5 + 6) / 6 = 3.5$$
 - Expected time for linear search is
$$(n / 2)$$
 (see topic 7)

David Keil 1/03 15

Random-number generation

- Computers are deterministic
- We *simulate* randomness by using a function with a humanly hard-to-predict return value
- Function *rand()*, in *stdlib.h*, generates a pseudorandom value in the range 0 to the argument
- On each call an internal *seed* value is updated
- To initialize seed, call *srand()*

David Keil 1/03 16

Using the C library's *rand* function

```
#include <stdio.h>           [random.c]
#include <stdlib.h>
#include <time.h>
void main(void)
{
    /* Initialize seed value, using current
time: */
    time_t ltime;
    time(&ltime);
    srand(32767*ltime);

    /* Simulate rolling of dice: */
    printf("Result of rolling dice: %d, %d.\n",
        rand() % 6 + 1, rand() % 6 + 1);
}
```

Sample output: Result of rolling dice: 2, 5

David Keil 1/03 17

Simulations of chance

Exercise:

- a) Compute the probability that the outcome of 2 throws of a die is at least 4
- b) Simulate this with 1000 tosses using *rand()*
- c) Compare empirical results (b) with theoretical prediction (a)

David Keil 1/03 18

Applications of probability in CS

- *Games*: Making play more realistic by generating unpredictable events
- *Average-case analysis of algorithms*: Expected (as opposed to worst-case) times may be obtained for Bubble sort, linear search, hashing
- *Evolutionary computation*: A new field modeled on natural evolution and using randomness heavily

David Keil 1/03 19

Evolutionary computation

- Populations of solutions to a problem are generated starting with a random guess
- Fitness tests are applied
- A new generation is chosen and altered using random mutation and crossover
- The process iterates, yielding better and better solutions

David Keil 1/03 20