Discrete Random Variables

ECE 313

Probability with Engineering Applications

Lectures 13 & 14 - October 11 & 13, 1999

Professor Ravi K. Iyer

University of Illinois

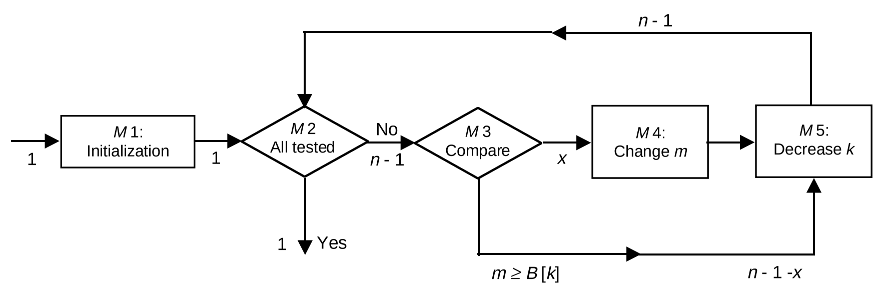
Analysis of Program Max

- Consider the **Pascal program MAX** that finds the largest element in the given array *B*.
- Given an array of *n* elements, *B*[1], *B*[2],…..*B*[n], we will find *m* and *j* such that $m = B[j] = \max\{B[k] \mid 1 \le k \le n\}$, and for which *j* is as large as possible.

```
program MAX (input, output);
     label 1, 2, 3, 4, 5, 6;
     const n = 100;
     var j, k, m: integer;
             B: array [1..n] of integer;
begin
1:  j : = n;  k : = n - 1;  m : = B[n]
2:   while (k > 0) do
         begin
3:            if B[k] > m
                 then
4.                  begin
                         j : = k;
                          m : = B[k]
                    end;
5.             k : = k - 1
         end;
6:  writeln (j, m)
end.
```

Analysis of Program Max (cont.)

- Analyze the time required for its execution.
- The time of execution depends on:
  - machine
  - compiler
  - input data
- Examine the effect of input data on execution time.
- We study frequency counts for each step.



Analysis of Program Max (cont.)

- X is the number of times we must change the value of the current maximum.
- The value of X depends on the pattern of numbers constituting the elements of the array B.
- Each such pattern may be considered a sample point with a fixed assigned probability.
- X can be thought of as a random variable over the sample space.

Step number	Frequency count	Number of statements
M1	1	3
M2	n	1
M3	$n - 1$	1
M4	X	2
M5	$n - 1$	1
M6	1	1

Analysis of Program Max (cont.)

- Determine the distribution of the random variable X for a given assignment of probabilities over the sample space.
- The image of the random variable X is $\{0, 1,\ldots, n - 1\}$:
  - minimum value of X occurs when $B[n] = \max \{B[k] \mid 1 \le k \le n\}$
  - maximum value of X occurs when $B[1] > B[2] > \ldots > B[n]$
- For simplicity, assume that $B[k]$ are distinct values. Without a loss of generality, also assume the vector of elements ($B[1], B[2],\ldots B[n]$ is any one of the permutations of the integers $\{1, 2,\ldots,n\}$, $n!$
- Sample space S_n: set of all permutations of n integers $\{1, 2,\ldots,n\}$.
- Assume that all $n!$ permutations are equally likely.
- For all s in S_n, $P(s) = 1/n!$:

  We define a random variable X_n as a function with domain S_n and the image $\{0, 1,\ldots, n - 1\}$.

Analysis of Program Max (cont.)

- As n changes, we have a sequence of random variables X_1, $X_2,\ldots$, where X_i is defined on the sample space S_i.

  The probability mass function of X_n, $p_{X_n}(k)$ is denoted by p_{nk}. Then:

$$P_{nk} = P(X_n = k)$$
$$= \frac{\text{numbers of permutations of } n \text{ objects for which } X_n = k}{n!}$$

Analysis of Program Max (cont.)

- Now establish a recurrence relation for p_{nk}:

  Consider a sample point $s = (b_1, b_2, \ldots, b_n)$, a permutation on {1, 2,…, n}.

  Consider two mutually exclusive and collectively exhaustive events:

  A = "$b_1 = n$"

  $\bar{A}$ = "$b_1 \neq n$"
- If event A occurs, then a comparison with b_1 (in program MAX) will force a change in the value of m.

Analysis of Program Max (cont.)

- The value obtained for X_n will be one higher than a similar value obtained while examining the previous n -1 elements $(b_2, \ldots, b_n)$.
- Note that $(b_2, \ldots, b_n)$ is a permutation on {1, 2, …, n -1}.
- The number of times the value of m gets changed while examining $(b_2, \ldots, b_n)$ is X_{n-1}.
- Thus:

$$P(X_n = k \mid A) = P(X_{n-1} = k-1) = p_{n-1,k-1}$$

- The occurrence of event $\bar{A}$ implies that the count of exchanges does not change when we examine b_1:

$$P(X_n = k \mid \bar{A}) = P(X_{n-1} = k) = p_{n-1,k}$$

Analysis of Program Max (cont.)

- By assumption of equiprobable sample space, we have:

  $P(A)=1/n$ and $p(\overline{A})=(n-1)/n$

  By the theorem of total probability:

$$
\begin{aligned}
P_{nk} &= P(X_n = k) \\
&= P(X_n = k \mid A)P(A) + P(X_n = k \mid \overline{A})P(\overline{A}) \\
&= \frac{1}{n} p_{n-1,k-1} + \frac{n-1}{n} p_{n-1,k}
\end{aligned}
$$

- Thus, we can recursively compute p_{nk} if we provide the initial conditions.

  Since the image of X_n is {0,1,…,n-1}, we know that: $P_{nk} = 0$ if $k < 0$.

  With $n = 1$, the **while** loop in program MAX will never be executed.

Analysis of Program Max (cont.)

- Therefore X_1 =0, i.e:

$$P(X_1=0) = p_{1,0} = 1 \text{ and } P(X_1=1) = p_{1,1} = 0$$

- The complete specification to evaluate p_{nk} is:

$$p_{1,0} = 1$$
$$p_{1,1} = 0$$

$$
p_{nk} = \begin{cases} \frac{1}{n} p_{n-1,k-1} + \frac{n-1}{n} p_{n-1,k}, & 0 \le k \le n-1, n \ge 2, \\ 0, & \text{otherwise.} \end{cases}
$$