Mathematics of Computer Science. - M.I.T. opencourseware


**Abstract:**
Random variable is the lecture about the way to calculate the random events probability with various method. If the possiability of events is stable, we can use indicator or distribution function or even Guessing strategy. Binomial distribution is when the possibility is 1/2 and all events are mutually independent, we can use it to count probability.


**For the English:**
indicator, envelop, swap, unbiased, contrarian

**Lec 21. RANDOM VARIABLE:**

DEF: a random variable R is a function R : S (sample space) => |R
ex) toss 3 coin // R = # heads, M = 1 if all 3 coins match | 0 otherwise

DEF: an Indicator (aka Bernoulli or characteristic)
r1, v1 is a r, v with range = [0, 1]
{HHH, TTT ~~ check origin graph}
{w | R(w) = x} is the event that R = X}
DEF: Pr(R = x) = ∑Pr(w)

Pr(R=2) = Pr(HHT, HTH, THH) = 3/8, Pr(m=1) = Pr(HHH, TTT) = 1/4
Pr(R>=2) = ∑Pr(R=i) = 1/2

for A ⊆ |R, Pr(R∈A) = ∑Pr(R=a)
A = {1, 3}, Pr(R∈A) = 1/2
Pr(R=2|m=1) = 0

DEF: two r1, v1's R1, R2 are indep- if ∀x1, x2 ∈ |R
Pr(R1=x1|R2=x2) = Pr(R1=x1) or Pr(R2=x2) = 0 => "no influence"
EQUIV: product form
∀x1, x2 ∈ |R Pr(R1=x1∩P2=x2) = Pr(R1=x1)*Pr(R2=x2)
Pr(R=2∩m=1) = 0 != Pr(R=2)*Pr(m=1) => R&M are not indep-
2 fair indep- 6side dice : D1, D2
let S = D1 + D2 let T = {1 if S = 7 | 0 = otherwise}
Pr(S=12∩D1=1) = 0 != Pr(S=12)&Pr(D1=1) => S&D => dependent
Pr(T=1|D1=1) = 1/6 = Pr(T=1) // Pr(T=1|D1=2) = 1/6 = Pr(T=1)
Pr(T=1|D1=6) = 1/6 = Pr(T=1) // Pr(T=0|D1=2) = 5/6 = Pr(T=0) - 보충

DEF: given a r,v R1 the prob- (or point) distribution func (pdf) for R is f(x) = Pr(R=x)
DEF: the culmulative dist func F (cdf) for R is f(x) = Pr(R=<x) = ∑Pr(R=y) (f(0)=p, f(1)=1)
ex) for an indicator rv f(0) = P f(1) = 1-p

For uniform r1,v1 on [1, n] fn(k) = 1/n for 1=< k =<n
Fn(k) = k/n

- Guessing strategy
0. envelopes contains y&z ∈ [0,n] where y<z
1. player chooses x uniformly in {1/2, 1|1/2, 2|1/2 ... n|1/2}
2. player hopes y<x<z
3. player opens random envelop to reveal r ∈ {y, z}
4. player swaps if r < x

check the origin tree node

y/2n, z-y/2n, n-z/2n => n, n, s, n, s, s => l, w, w, w, w, l
Pr(w) = y/2n + z-y/2n + z-y/2n + n-z/2n = n+z-y/2n = 1/2 + z-y/2n
>= 1/2 + 1/2n == PROFIT!

- Unbiased binomial dist
fn(k) = (n|k)2^-n   n>= 1   0 =< k =< n   unbiased = p == 1/2

-General binomial dist

fn,p(k) = (n|k)p^k*(1-p)^n-k   0 < p <1
ex) n components each fails mut- indep- with prob p (0<p<1)
R = # failures

THM: Pr(R=k) - fn,p(k) for any k (0=<k=<n)
check the origin tree node
---- sample space == 2^n components
(n|k) sample points have k failed components // each has prob- p^k*(1-p)^n-k
Pr(R=k) = (n|k)p^k*(1-p)^n-k = fn,p(k) - swap-possible
prob 1   1/2   1/10   1/100   1/1k   1/1m
Pr(50H)   1/2^50

fn,p(@n) =< 2^[@logp/@ + (1-@)log1-p/1-@]n / √2π@(1-@)n   for 0<@<1

max value at @ = p   for n = 100, p = 1/2
fn,p(pn) ~=< 1/ √2πp(1-p)n
Pr(50H) = 1/√50π = 0.08...

정규분포곡선그래프 생성