

# TDAB01 Sannolikhetslära och Statistik

Jose M. Peña  
IDA, Linköping University, Sweden

## Föreläsning 5

# Översikt

- ▶ **Stora talen lag**
- ▶ **Centrala gränsvärdessatsen**
- ▶ **Simulering**
- ▶ **Monte Carlo metoder**

# Stora talens lag

- ▶ Medelvärde:  $\bar{X}_n = \frac{X_1 + X_2 + \dots + X_n}{n}$
- ▶ Medelvärdet av många oberoende slumpvariabler med samma väntevärde  $\mu$  och varians kommer att ligga allt närmare  $\mu$ .
- ▶ **Stora talens lag** (law of the large numbers, på engelska)

$$\lim_{n \rightarrow \infty} P(|\bar{X}_n - \mu| > \epsilon) = 0$$

- ▶ **Bevis** via Chebyshevs olikhet. Låt  $X = \bar{X}_n$ . Då  $\mathbb{E}(X) = \mu$ . Då

$$P(|X - \mu| > \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$

eftersom  $\sigma^2$  är  $\text{Var}(X) = \text{Var}(\bar{X}_n) = \text{Var}(X_i)/n \rightarrow 0$  när  $n \rightarrow \infty$ .

## Centrala gränsvärdessatsen

- ▶ Hur är  $\bar{X}_n$  fördelad ?
- ▶ **Centrala gränsvärdessatsen** (central limit theorem, på engelska). Låt  $X_1, X_2, \dots, X_n$  vara oberoende variabler med samma väntevärde  $\mu$  och standardavvikelse  $\sigma$ , och låt

$$S_n = X_1 + X_2 + \dots + X_n$$

När  $n \rightarrow \infty$  så kommer den standardiserade summan

$$Z_n = \frac{S_n - \mathbb{E}(S_n)}{\text{Std}(S_n)}$$

att **konvergera i fördelning** till en  $N(0, 1)$  variabel, dvs

$$F_{Z_n}(z) = \mathbf{P}\left(\frac{S_n - n\mu}{\sigma\sqrt{n}} \leq z\right) \rightarrow \Phi(z)$$

- ▶ Då  $S_n$  och  $\bar{X}_n$  konvergerar i fördelning till  $N(n\mu, \sigma\sqrt{n})$  och  $N(\mu, \sigma/\sqrt{n})$ . Vanlig tumregel:  $n > 30$ . Se Example 4.13 i Baron.
- ▶ Man kan approximera en binomialfördelning med  $N(np, \sqrt{npq})$  för  $n > 30$  pga CLT och faktumet att en binomial är en summa av  $n$  lika Bernoulli variabler. Samma gäller för negativa binomialfördelningen (summa av  $k$  geometriska variabler), och gamma fördelningen (summa av  $\alpha$  exponentiella variabler).

- ▶ **Pseudoslumptalsgenerator**: Datorer kan generera en lång sekvens tal som ser ut som  $U(0,1)$  slumptal. Good enough.
- ▶ R: `runif(1)`. Matlab: `rand`. Python: `numpy.random.uniform()`.
- ▶ Från  $U \sim U(0,1)$  kan vi skapa slumptal från andra fördelningar.
- ▶ Exempel: **Bernoulli** med sannolikhet  $p$  att lyckas:

$$X = \begin{cases} 1 & \text{om } U < p \\ 0 & \text{om } U \geq p \end{cases}$$

- ▶ R kod Bernoulli: `U=runif(1); X=(U<p)`
- ▶ Exempel: **Binomial**. Summan av Bernoullis
  - ▶ R-kod för Binomial( $n,p$ ): `U=runif(n); X=sum(U<p)`
- ▶ See Example 5.14 i Baron.

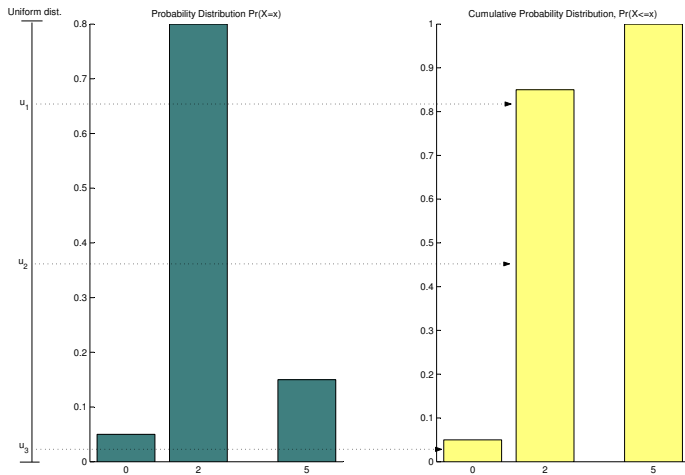
## Simulering från diskret fördelning

- ▶ Simulering från allmän diskret fördelning, dvs

$$p_i = P(X = x_i), \quad \sum_{i=1} p_i = 1$$

- ▶ Dela upp intervallet  $[0, 1]$  i delintervall:
  - ▶  $A_1 = [0, p_1)$
  - ▶  $A_2 = [p_1, p_2)$
  - ▶  $\vdots$
  - ▶  $A_n = [p_{n-1}, 1)$
- ▶ Slumpa  $U \sim U(0, 1)$ .
- ▶ Om  $U \in A_i$  låt  $X = x_i$ .
- ▶ Se Example 5.9 i Baron.

# Inversa cdf metoden: Diskreta fallet



## Inversa cdf metoden: Kontinuerliga fallet

- ▶ **Theorem.** Låt  $X$  vara en kontinuerlig variabel med cdf  $F_X(x)$  och låt  $U = F_X(X)$  vara en ny slumpvariabel. Då gäller att  $U \sim U(0,1)$ .
  - ▶ **Inversa transformationsmetoden:** Antag att  $X$  har cdf  $F(X)$ .  $X$  kan då simuleras med hjälp av en  $U \sim U(0,1)$  variabel, dvs

$$X = F^{-1}(U)$$

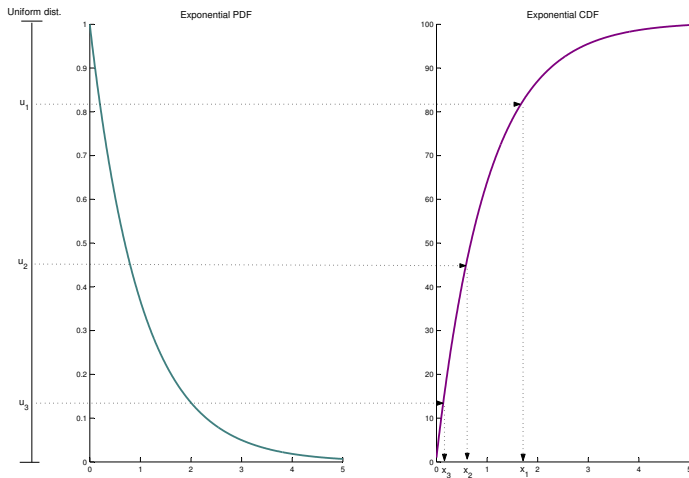
- ▶ Dvs, lös ut  $X$  från ekvationen  $U = F(X)$ .
- ▶ Exempel:  $X \sim \text{Exp}(\lambda)$ . Då

$$U = 1 - e^{-\lambda X}$$

$$X = -\frac{1}{\lambda} \ln(1 - U)$$



# Inversa cdf metoden: Kontinuerliga fallet



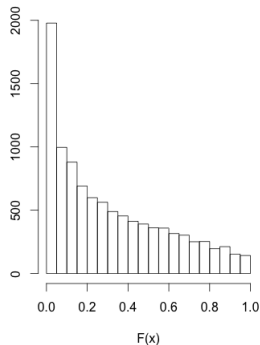
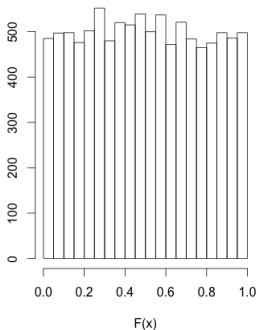
## Simulering i R

- ▶  $n$  **slumptal** från  $N(\mu = 2, \sigma^2 = 3^2)$  simuleras med  
`rnorm(n, mean = 2, sd = 3)`
- ▶  $n$  **slumptal** från  $\text{Gamma}(\alpha = 2, \lambda = 3)$  simuleras med  
`rgamma(n, shape = 2, rate = 3)`
- ▶ Beräkna **pdf:en** i punkten  $x = 1.5$  för  $N(\mu = 2, \sigma^2 = 3^2)$ :  
`dnorm(x=1.5, mean = 2, sd = 3)`
- ▶ Beräkna **cdf:en** i punkten  $x = 1.5$  för  $N(\mu = 2, \sigma^2 = 3^2)$ :  
`pnorm(x=1.5, mean = 2, sd = 3)`

## Testar inversa CDF metoden

- ▶ Följande funkar (dvs  $F_x$  blir likformigt fördelad):

- ▶ `x = rgamma(10000, shape = 2, rate = 3)`
- ▶ `Fx = pgamma(x, shape = 2, rate = 3)`
- ▶ `hist(Fx, 30)`



- ▶ Följande funkar inte (dvs  $F_x$  blir **inte** likformigt fördelad):

- ▶ `x = rgamma(10000, shape = 2, rate = 3)`
- ▶ `Fx = pgamma(x, shape = 1, rate = 3)`
- ▶ `hist(Fx, 30)`

## Monte Carlo metoder

- ▶ Simulering från fördelningar kan användas för att approximera t ex olika sannolikheter.
- ▶ Låt  $X_1, X_2, \dots, X_N$  vara oberoende dragningar från en sannolikhetsfördelning. Vi kan t ex approximera sannolikheten  $p = P\{X < 2\}$  med

$$\hat{p} = \hat{P}\{X < 2\} = \frac{\text{antal av } X_1, X_2, \dots, X_N \text{ som är mindre än } 2}{N}$$

- ▶  $\hat{\theta}$  (t ex  $\hat{p}$ ) är en **estimator** (uppskattning) av kvantiteten  $\theta$  (t ex  $p$ ).
- ▶ 

```
x = rnorm(10000, mean = 1, sd = 2)
pHat = sum(x<2)/10000
```

## Monte Carlo metoder

- ▶ Men  $\hat{p}$  är bara en **skattning** av  $p$ . Varierar från stickprov till stickprov.
- ▶ Om vi upprepar hela receptet flera gånger, varje gång med ett nytt stickprov av storleken  $N$ , kommer vi då att ha rätt i genomsnitt? Dvs, är  $\mathbb{E}(\hat{p}) = p$ ?
- ▶ Hur mycket kommer  $\hat{p}$  att variera från stickprov till stickprov? Hur stor är  $\text{Var}(\hat{p})$ ?
- ▶  $Y = \text{Antal } X_1, \dots, X_N \text{ som är mindre än } 2$ . Då  $Y \sim \text{Bin}(N, p)$ . Så

$$\mathbb{E}(\hat{p}) = \mathbb{E}\left(\frac{Y}{N}\right) = \frac{1}{N}N \cdot p = p$$

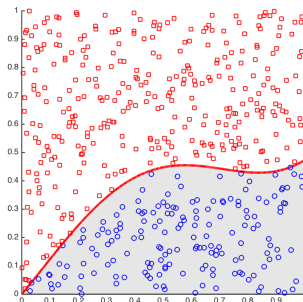
så  $\hat{p}$  är en **väntevärdesriktig** (unbiased på engelska) estimator av  $p$ .

$$\text{Var}(\hat{p}) = \text{Var}\left(\frac{Y}{N}\right) = \frac{1}{N^2}Np(1-p) = \frac{p(1-p)}{N}.$$

- ▶ Se Baron sidor 115-116 om hur man kan välja  $N$  för att nå given exakthet  $P\{|\hat{p} - p| > \varepsilon\} \leq \alpha$ .

## Monte Carlo integration

- ▶ Mål:  $\mathcal{I} = \int_0^1 g(x)dx$  där  $0 \leq x \leq 1$  och  $0 \leq g(x) \leq 1$ .



- ▶ Simulera likformigt fördelade tal  $U_1, \dots, U_N$  och  $V_1, \dots, V_N$ .
- ▶ Monte Carlo skattning

$$\hat{\mathcal{I}} = \frac{\text{Antal dragningar där } V_i < g(U_i)}{N}$$

- ▶ 

```
u = runif(10000)
v = runif(10000)
IHat = mean(v < g(u))
```

# Översikt

- ▶ **Stora talen lag**
- ▶ **Centrala gränsvärdessatsen**
- ▶ **Simulering**
- ▶ **Monte Carlo metoder**