# Curs introductiv: Matematici Speciale

Conf.dr. Maria Jivulescu

Departamentul de Matematică
UPT

# Cuprins



- Partea I: Prezentare structura curs MS
- Partea II: Background matematic
- Partea III: Motivație

# Curs Matematici Speciale



- Curs pentru anul I, semestrul II (3h/ 14 săptămâni)
- Curs, Joi, 9-12
- Curs: Conf.dr. Maria Jivulescu
- Seminar: 2h/ 14 săptămâni
  - Conf.dr. Maria Jivulescu (grupa 1)
  - Asist. drd. Florinela Popa (grupele 2,3)

## Ce vom face la acest curs?

Vom studia Teoria Probabilitatilor si Statistica Matematică si aplicații in CS!

Conținutul cursului:

- **Capitolul I: Introducere în Teoria Probabilităților**: definiția axiomatică a probabilității, scenariul matematic pentru teoria probab., probabilități condiționate, teoria lui Bayes asupra probabilității
- **Cap II: Variabile (vectori) aleatoare discrete**: distribuții clasice de probabilitate
- **Cap III: Variabile (vectori) aleatoare continue**: distributia normala, clopotul lui Gauss
- **Cap IV: Procese stochastice**: Lanturi Markov, Procese Poisson
- **Statistica**: Estimatori, Teorema de limită centrală

# Desfășurarea cursului de MS



Desfășurare curs pe parcursul celor 14 saptămani:

- Orar: Joi 9:00-12:00
- pdf, video, materiale și informații utile pe Campusul Virtual-UPT( acolo se găsesc deja cursurile/seminariile)
- ne dorim ca la curs să clarificăm cursul postat pe Campus, deci este necesar o citire a acestuia, pentru ca voi să puteți formula întrebări și să răspundeți la ele

Sunteți rugați să consultați permament Campusul Virtual-UPT!
Acest mediu ne asigură legatura între noi!

# Desfășurarea seminarului de MS



Desfășurare seminarului pe parcursul celor 14 saptămani:

- Seminarul se va ține on-site, pe grupe.
- suportul pdf pentru aplicatii pe Campusul Virtual-UPT
- se vor da 2 teste
- notele de la teste vor fi parte integrantă al notei finale
- prezența este obligatorie

# Evaluarea materiei de MS



Forme de evaluare a materiei MS

- Final seminar $\rightarrow$ nota activitate pe parcurs: media aritmetică a notelor de la teste+ Bonus
- maxim 2 puncte se pot obtine din raspunsuri: 0.5 bonus pentru fiecare răspuns bun
- Seminarul se consideră încheiat dacă nota obţinută este $\geq 4.5$; în caz contrar el trebuie recontractat, în anul următor
- Final curs $\rightarrow$
  Nota examen=1/2 *Nota activitate pe parcurs+1/2 * [Nota examen-iunie ($\geq 4.5$)+Bonus curs (1 punct)]
- Bonus curs se obţine din realizarea unui referat pe echipe (temele se vor anunta pe CV si va veti putea alege)

# Cum să ne asigurăm succesul la materia MS



- citirea cursurilor de pe CV-UPT înainte/după curs
- participarea/prezenta activa la curs/seminar
- teme/teste predate la timp
- consultarea în permanența a CV-UPT pentru anunțurile legate de curs

# Bibliografie



Material bibliografic este disponibil pe CV-UPT, dacă doriți în plus, carți, culegeri de Matematici Speciale ( Probabilități si Statistică) sunt disponibile la Biblioteca UPT.


- Emilia Petrișor, Probabilități și Statistică-Curs si Aplicații in Inginerie (ediția tipărită Editura Politehnica 2001, online-capitole pe CV)
- Probability and Computing- Randomized algorithms and Probability Analysis, M. Mitzenmacher, Eli Upfal, Cambridge University Press, 2017
- Applied Statistics and Probability for Engineers, Douglas Montgomery, George Runger, John Wiley- Sons, Inc, 2003

- noțiuni din liceu despre metodele de numărare( permutări, aranjamente, combinări) si despre teoria probabilităților (determinarea probabilității ca un eveniment să aibă loc); Seminarul din sap 1 are scopul de a reaminti aceste lucruri.
- noțiuni din liceu din analiza matematica (de exemplu, calcul integral)
- noțiuni din semestrul I legate de serii numerice, valori/vectori proprii
- noțiuni introductive despre integralele generalizate

$$\int_2^\infty \frac{1}{x-1}dx = \lim_{t\to\infty}\int_2^t \frac{1}{x-1}dx = \ln|t-1| - \ln 1 = \infty$$

- integrale duble

$$\int\int_D f(x,y)dxdy$$

Vor exista întâlniri pentru a explica noțiunile noi și materiale suplimentare cursului.

# Problemă



Se presupune ca se aruncă o monedă, pâna cand se obtine "capul"(C=H=head); fata opusa este 'stema' (S=T=tail)
Sa se determine probabilitatea ca de a arunca moneda de 3 ori, inainte de a obține H?

- probabilitatea (de a obține la prima aruncare C)=1/2
- probabilitatea (de a obține prima data C la a doua aruncare) = probabilitatea (de a obține la prima S si la a doua aruncare C)$=1/2 \times 1/2 = 1/4$
- probabilitatea (de a obține prima data C la a treia aruncare) = probabilitatea (de a obține la a prima si la a doua aruncare S, iar la a treia C)$=1/2 \times 1/2 \times 1/2 = 1/8$
- in general, probabilitatea de a obține $N-1$ de S, inainte de a obține evenimentul cautat C este o serie de evenimente cu probabilitățile $1/2, 1/4, 1/8, \ldots (1/2)^{N-1} \times 1/2$

Cum putem să întelegem mai bine ce se întampla?

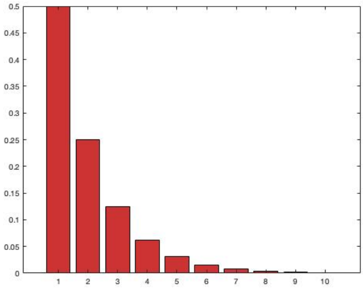



Figure: Graficul aruncarii unei monede pana cand se obtine Cap

- Histograma=graficul ce relaționează evenimente de probabilitatea lor;
- OX -numarul de aruncari, iar OY- probabilitatea acestui eveniment

# Alta problemă...mai grea



Care este numărul mediu de aruncări pentru a fi sigur (cu probabilitate de 90%) că se obține cap?

- P(succes cu o aruncare)=1/2
- P(succes cu doua aruncări)=1/2+1/4=0.75
- P(succes cu trei aruncări)=1/2+1/4+1/8=0.875
- P(succes cu patru aruncări)=1/2+1/4+1/8+1/16=0.9375

Raspuns: n=4

Se observă că ne apare seria geometrică $\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}, \ |r| < 1$

Se folosește la situații de genul : $N - 1$ insuccese, urmate de un succes

# O altă problemă...



Care este probabilitatea de a obține un Cap din 3 aruncări independente ale unei monede?
P( un Cap din 3 aruncări)
$=P(C,S,S)+P(S,C,S)+P(S,S,C)=3 \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{3}{8}$
Se folosesc combinările pentru a număra cazurile posibile, aici $C_3^1 = 3$
In general, $C_n^k = \frac{n!}{k!(n-k)!}$

# Aproximări. Polinomul Taylor

Polinoame Taylor ce aproximează diferite funții:
https://mathlets.org/mathlets/taylor-polynomials/
Rezultat folosite

- la distribuția Poisson: $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$
- $\ln(1+x) \approx x$;
- $\ln(1+x) \approx x - x^2$

Calcul Integral:

- integrarea prin părți
- metoda substituției

Ne poate furniza rezultate surprinzătoare legate de anumite situații/întrebări.

**Exemplu**: celebra problemă a zilei de naștere ("Birthday paradox")
**Formulare**: Intr-o cameră cu 50 de persoane, care este probabilitatea sa existe două persoane cu aceeași zi de naștere?
Obs.:

- aparent ne trebuie 366 de persoane pentru a asigura cel puțin un cuplu de persoane cu aceeași zi de naștere
- aparent, probabilitatea ca 2 persoane din 50 să aibă aceeași zi de naștere este mică

# Birthday paradox



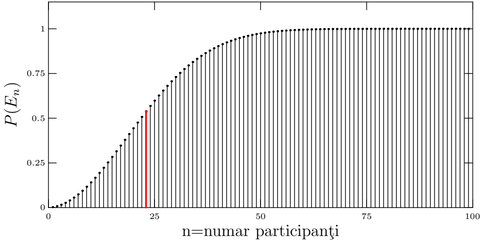



Figure: Probabilitatea ca 2 persoane să aibă aceeași zi de naștere ca funcție de numărul de persoane din grup

**Rezultat**: probabilitatea ca 2 persoane din 50 să aibă aceeași zi de naștere este approximativ 97%.
Pentru a avea probabilitate de 50%, sunt necesare doar 23 de persoane!

Cum se obține acest rezultat?

Vom studia probabilitatea ca alegand 50 persoane, ele sa aiba zile de nastere diferite.

- P(primele doua persoane alese să aibă zile de zile de nastere diferite)$=\frac{365}{365} \times \frac{364}{365}$
- P(primele 50 persoane alese să aibă zile de zile de nastere diferite)$=\frac{365}{365} \times \frac{364}{365} \ldots \times \frac{316}{365} \approx= 0.03$
- P(primele $k$ persoane alese să aibă zile de zile de nastere diferite)$=\frac{365}{365} \times \frac{364}{365} \ldots \times \frac{365-k+1}{365}$

Deci, probabilitatea ca sa existe cel puțin 2 persoane din 50 care să aibă aceeași zi de naștere este $1 - 0.03 \approx= 0.97$.

Probabilitatea =aleatoriu (random)

Algoritm probabilist=**un algoritm care fac alegeri aleatoare în timpul execuției lor**, adica un program ce va folosi valori generate de un generator de numere aleatoare, pentru a decide pașii următori, la anumite ramuri de execuție.

De ce este nevoie de algoritmi probabiliști?
**Sunt mai eficienți, mai simpli și mai ușor de programat.**

**Prețul: Răspunsul (output) poate să fie incorect, cu o anumită probabilitate** sau eficienta acestui algoritm este garantata cu o anumita probabilitate

# Algoritm probabilist



De ce am face un program care să dea un rezultat posibil greșit (cu o anumita probabilitate)?

**Probabilitatea de a comite o eroare este suficient de mică ca să merite imbunătățiri in viteza sau in memorie**.

Observație: La algoritmii probabilisti este interesant si studiul complexitatii problemei!

# Exemplu de algoritm probabilist



Algoritm probabilist pentru verificarea echivalenței a două polinoame

**Scop**: folosirea notiunii de aleator pentru a verifica eficient corectitudinea unui rezultat.

**Situatie:** Presupunem ca cineva a programat un program de multiplicarea a monoamelor intr-un polinom dat ca produs.

Ne dorim sa verificam daca acest program ne da un rezultat corect.

**Exemplu:** dat imput $(x-1)(x-2)(x+3)(x-4)$,

programul ne da ca este egal cu $x^4 - 5x^3 + 2x^2 - 24$.

Cum se poate verifica ca este corect?

# Exemplu de algoritm probabilist



**Problema**: date doua polinoame $F(x) = \prod_{i=1}^{d}(x_i - a_i)$ și $G(x) = \sum_{i=1}^{d} c_i x^i$, cum putem verifica daca $F(x) = G(x)$?

**Metooda 1:**

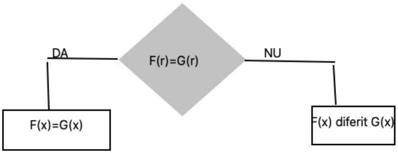
- Se efectuează înmulțirea celor $d$ monoame, pentru a-l aduce pe F la forma canonică.
- Sunt necesare un număr de operații de ordinul lui $d^2$, ceea ce implică timp și memorie!

**Concluzie:** aceasta metoda este ineficienta: dorim sa verificam eficient un rezultat, nu sa rescriem un nou program, care face acelasi lucru cu primul program!

# Exemplu de algoritm probabilist

**Metooda 2:**

- se fixeaza $d = \max\{F(x), G(x)\}$
- impunem ca algoritmul sa aleaga aleator uniform un numar $r \in S = \{1, 2, \ldots, 100d\}$
- algoritmul va calcula valorile $F(r)$ si $G(r)$ (asta se poate face intr-un numar de $d$ pasi)
- decizia se bazeaza pe schema



# Exemplu de algoritm probabilist

Apar doua intrebari:

- Q1: Care este timpul de rulare al acestui algoritm? Raspuns: de ordinul $d$
- Q2: In ce caz da algoritmul un raspuns gresit?
  Raspuns:
  - Daca $F(x) = G(x), \forall x \in S$, atunci algoritmul da raspuns corect (pt ca are loc si faptul ca $F(r) = G(r)$, iar in acest caz algoritmul ne spune ca $F(x) = G(x)$)
  - daca $F(x) \neq G(x)$ si $F(r) \neq G(r)$, atunci algoritmul da raspuns corect
  - Dar, daca a $F(x) \neq G(x)$ si $F(r) = G(r)$, atunci algoritmul da raspuns gresit.

# Exemplu de algoritm probabilist



Cazul in care algoritmul poate da un raspuns gresit:

$$F(x) \neq G(x), \text{ dar } F(r) = G(r) \rightarrow (F - G)(r) = 0,$$

deci $r$ este o radacina a polinomului $(F - G)$, polinom de grad cel mult $d$.
Teorema fundamentala a algebrei: polinomul $(F - G)$ poate sa aiba cel mult $d$ radacini.
Deci, sunt cel mult $d$ valori ale lui $r$ din multimea $\{1, \ldots, 100d\}$ pentru care $F(r) = G(r)$, chiar daca $F(x) \neq G(x)$.

$$\text{Probabilitatea (algoritmul da raspuns gresit )} \leq \frac{d}{100d} = 0.01$$

**Concluzie: In cazul $F(x) \neq G(x)$, algoritmul da razultat corect în cel putin 99% din cazuri.**
Cum se poate imbunatati aceasta probabilitate?
Rularea programului de mai multe ori sau marirea multimii $S$.

Vă mulțumesc pentru atenție și vă doresc mult succes în acest semestru!

Întrebări, vă rog?!