Introducere in programarea probabilistă
Cursul 1

Programare şi modelare probabilistă - anul III

Facultatea de Informatică, UAIC

e-mail: adrian.zalinescu@uaic.ro

web: https://sites.google.com/view/fiicoursepmp/home

9 Octombrie, 2023

Despre modelare probabilistă

Motivaţie:

- volumele de date stocate acum sunt foarte mari (era „big data”);
- avem nevoie de instrumente capabile să:
  - să caute,
  - să vizualizeze,
  - să *modeleze* şi
  - să *înţeleagă* aceste date.

Modelare probabilistă

- Un model probabilist descrie informaţii pe care le-am putea observa dintr-un sistem.
- De regulă folosim teoria (matematică a) probabilităţilor pentru a exprima formele de incertitudine şi zgomot asociate.
- Probabilitatea inversă (adică regula lui Bayes) ne permite:
  - să deducem informaţii necunoscute,
  - să adaptăm modelele noastre,
  - să facem predicţii şi
  - să învăţăm din date.

Regula lui Bayes ca bază pentru raţionament probabilist

$$P(ipoteze|date) = \frac{P(date|ipoteze)P(ipoteze)}{P(date)}$$

- Regula ne învaţă cum să deducem (inferăm) ipoteze din date.
- Iar învăţarea şi predicţia sunt forme de raţionament deductiv (inferenţă).

Modalitatea de evaluare

Nota finală va fi alcătuită din:

- **Prezenţă laborator** 10 puncte;
  - 1 prezenţă = 1 punct, până la un maxim de 10p;
- **Activitate laborator** 30 puncte**: teme** ce vor fi
  - anunţate la laborator;
  - predate până la termenul desemnat de profesor.
- **Examen parţial:** 30 puncte
- **Examen final:** 30 puncte

Condiţii de promovare a disciplinei:

- **minim** 8 prezenţe;
- **minim** 12 puncte obţinute din activitatea la laborator (teme);
- **minim** 25 puncte obtinute din examene

Bibliografie (selectivă):

- Osvaldo Martin, *Bayesian Analysis with Python: Introduction to statistical modeling and probabilistic programming using PyMC3 and ArviZ*, 2nd ed., 2018
  - versiune online: `https://github.com/PacktPublishing/Bayesian-Analysis-with-Python-Second-Edition`
- Cameron Davidson-Pilon, *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*, 2016
  - versiune online: `https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers`
- Luis Enrique Sucar, *Probabilistic Graphical Models: Principles and Applications*, Springer, 2021

Ca limbaj de programare, vom folosi `Python` şi diverse librării ale sale:

- `NumPy`, `SciPy`, ...: oferă suport pentru distribuţii probabiliste;
- `Matplotlib`, `Seaborn`, ...: suport grafic, vizualizare date;
- `PyMc`, pgmpy, ...: inferenţă (programare probabilistă).

Plan

O posibilă motivaţie

„Ştiinţa reală a logicii este competentă în prezent numai cu lucruri certe, imposibile sau cu totul îndoielnice, pe care nu trebuie să le argumentăm (din fericire). Prin urmare, adevărata logică pentru această lume este calculul probabilităţilor, care ia în considerare magnitudinea probabilităţii care este sau ar trebui să fie în mintea unui om rezonabil."

James Clerk Maxwell (1850)

Exemplu motivator[1]

„Să presupunem că într-o noapte întunecată, un poliţist merge pe o stradă, aparent pustie; dar dintr-o dată aude o alarmă antiefracţie, se uită peste drum şi vede un magazin de bijuterii cu fereastra spartă. Apoi, un domn care poartă o mască iese târându-se prin fereastra spartă, purtând o geantă care se dovedeşte a fi plină de bijuterii scumpe. Poliţistul nu ezită deloc să decidă că acest domn este necinstit.”

[1] Probability Theory: The Logic of Science, by E. T. Jaynes

- Care este procesul de raţionament prin care poliţistul ajunge la această concluzie?
- Acest raţionament nu este unul deductiv logic, dar are un anumit grad de validitate.
- Există şi alte concluzii care se pot trage:
  - acest domn era proprietarul magazinului de bijuterii, venea acasă de la un bal mascat şi nu avea cheia cu el;
  - în timp ce trecea pe lângă magazinul său, dintr-un camion care trecea s-a aruncat o piatră în fereastră, iar el doar îşi proteja proprietatea.
- Evidenţele nu au făcut sigură necinstea domnului, dar a făcut-o extrem de plauzibilă.

Exemplu de raţionament
Un *silogism* este un tip de argument logic ce aplică raţionamentul deductiv pentru a ajunge la o concluzie bazată pe două propoziţii, care sunt afirmate sau presupuse a fi adevărate.
Silogisme tari (logica Aristotelică, cu două valori)

$$\frac{\begin{array}{l}\text{dacă } A \text{ e adevărată atunci } B \text{ e adevărată}\\ A \text{ e adevărată}\end{array}}{\text{deducem că } B \text{ e adevărată}}$$

şi inversa

$$\frac{\begin{array}{l}\text{dacă } A \text{ e adevărată atunci } B \text{ e adevărată}\\ B \text{ e falsă}\end{array}}{\text{deducem că } A \text{ e falsă}}$$

Silogisme slabe

$$\frac{\begin{array}{l}\text{dacă } A \text{ e adevărată atunci } B \text{ e adevărată}\\ B \text{ e adevărată}\end{array}}{\text{deducem că } A \text{ devine mai plauzibilă}}$$

$$\frac{\begin{array}{l}\text{dacă } A \text{ e adevărată atunci } B \text{ e adevărată}\\ A \text{ e falsă}\end{array}}{\text{deducem că } B \text{ devine mai puţin plauzibilă}}$$

Exemplu:
$A \equiv$ Va începe să plouă până cel târziu la 10 dimineaţa.
$B \equiv$ Cerul va deveni noros înainte de ora 10 dimineaţa.

Silogismul slab aplicat de poliţist

$$\frac{\begin{array}{l}\text{dacă } A \text{ e adevărată atunci } B \text{ devine mai plauzibilă}\\ B \text{ e adevărată}\end{array}}{\text{deducem că } A \text{ devine mai plauzibilă}}$$

Identificaţi A şi B.
Trebuie menţionat că gradul de plauzabilitate depinde de informaţia apriori (experienţa poliţiştilor).

Interpretări alternative ale probabilităţii

- Interpretarea clasică (Laplace): probabilitatea are de a face cu evenimente echiprobabile; dacă un anume experiment are N rezultate posibile, atunci probabilitatea de realizare a orcărui rezultat posibil este $1/N$.
- Interpretarea frecvenţială, mai utilizată în versiunea clasică a statisticii, afirmă că:
  probabilitatea este frecvenţa (de unde numele) *pe termen lung a producerii unui eveniment* (repetat la infinit/de un număr foarte mare de ori, în mod independent).
- Interpretarea Bayesiană(sau *logică, subiectivă*):
  probabilitatea este o măsură a plauzabilităţii (verosimilităţii) sau încrederii în realizarea unui eveniment.
- Cele două definiţii ar trebui să coincidă pe exemplele în care un anumit eveniment este reproductibil la infinit.

Cele două interpretări dau naştere la principalele două abordări în probabilităţi şi statistică:

- Interpretarea *obiectivă* (clasică, frecvenţială): probabilităţile există în lumea *reală* şi ele pot fi măsurate.
- Interpretarea *epistemologică* (Bayesiană): probabilitatea are de a face cu cunoaşterea umană - ea măsoară încrederea.
- În domeniul inteligenţei artificiale, se preferă în general ultima abordare, Bayesiană.

Formalizare 1/4

Cum măsurăm gradul de plauzabilitate?
Notaţii:

- $A, B, \ldots$ – *propoziţii*
- $\overline{A}$ sau $\neg A$ – *negaţia* (nu are loc A)
- AB sau $A \wedge B$ – *conjuncţia* (A şi B)
- $A + B$ sau $A \vee B$ – *disjuncţia* (A sau B)
- $A \Rightarrow B$ – *implicaţia* (A implică B)
  echivalent cu $\overline{A} + B$ sau cu $A = AB$
- $A \mid B$ – *plauzibilitate condiţională* (A ştiind B)

Formalizare 2/4

Prezumţii (postulate, deziderate):

1. *Gradele de plauzibilitate sunt reprezentate de numere reale.* (Divizibilitate şi comparabilitate)
2. *Plauzibilităţile ar trebui să varieze sensibil cu evaluarea plauzibilităţilor din model.* (Bun simţ)
3. *Dacă plauzibilitatea unei propoziţii poate fi derivată în mai multe moduri, toate rezultatele trebuie să fie egale.* (Consistenţă)

Cele trei presupuneri sunt cunoscute sub numele de "Postulatele (axiomele) lui Cox".

Formalizare 3/4

Teorema lui Cox:
Din cele trei postulate (şi funcţii adiţionale utilizate pentru a le materializa), se pot deduce următoare proprietăţi ale probabilităţilor (ca reprezentări ale gradelor de plauzabilitate):

- *adevărul* este reprezentat de $Pr(A \mid B) = 1$ şi
  falsitatea este reprezentată de $Pr(A \mid B) = 0$;
- $Pr(A \mid B) + Pr(\overline{A} \mid B) = 1$;
- $Pr(AB|C) = Pr(A|C)Pr(B|AC) = Pr(B|C)Pr(A|BC)$.

Cele trei proprietăţi pot fi văzute ca deducţii în teoria Bayesiana a probabilităţilor (văzută ca o extensie a calculului propoziţional).

Formalizare 4/4

Axiomele lui Kolmogorov (cele pe care se bazează teoria standard a probabilităţilor):
$(\Omega, \mathcal{E}, P)$ – spaţiu de probabilitate ($\mathcal{E} \subseteq \mathcal{P}(\Omega)$, σ-algebră)

- $P(E) \in \mathbb{R}$, $P(E) \geq 0$, $\forall E \in \mathcal{E}$;
- $P(\Omega) = 1$;
- $P(\sum_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i)$ pentru evenimente E_i, $i \geq 1$, disjuncte două câte două .

Din axiomele de mai sus se poate deduce, de exemplu, că $A \subseteq B$ implică $P(A) \leq P(B)$ şi că $P(\emptyset) = 0$ ($A, B \in E$).

Variabilă aleatoare – Definiţie

- $(\Omega, \mathcal{E}, P)$ – spaţiu de probabilitate;
- V – spaţiu de măsură;
- O variabilă aleatoare (v.a.) este o funcţie $X : \Omega \to V$;
- Tradiţional (cazul standard) $V = \mathbb{R}$. Dacă $X(\Omega)$ este *finită* sau *numărabilă*, atunci X se numeşte *v.a. discretă*.
- Funcţia de masă de probabilitate: (pentru v.a. discrete)

$$f_X(v) = P(X = v) = P(\{\omega \in \Omega \mid X(\omega) = v\})$$

sau funcţia de repartiţie (distribuţie):

$$F_X(v) = P(X \leq v) = P(\{\omega \in \Omega \mid X(\omega) \leq v\}).$$

ne permit să facem abstracţie de Ω şi să lucram doar cu valorile v.a. şi probabilităţile acestora. Notăm $Val(X) \equiv X(\Omega)$.

Variabilă aleatoare – în informatică

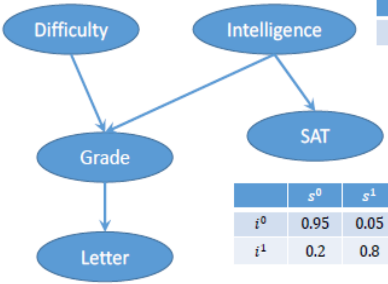
V – desemnează o structură de date

În acest caz $X : \Omega \to V$ se numeşte şi *element aleatoriu*.

Exemple:

- valori booleene aleatoare;
- cuvinte aleatoare;
- grafuri aleatoare;
- vectori aleatori;
- funcţii aleatoare;
- $\ldots$.

Modelul „Student”[2] - Reţea Bayesiană



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1
i^0,d^0	0.3	0.7
i^1,d^1	0.05	0.95
i^1,d^0	0.9	0.1
i^1,d^1	0.5	0.5

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.9	0.1
g^1	0.4	0.6

[2] Andrew D. Gordon, Thomas A. Henzinger, Aditya V. Nori, and Sriram K. Rajamani. 2014. Probabilistic programming. In Proceedings of the Future of Software Engineering (FOSE 2014)

Explicarea modelului

Difficulty (D), Intelligence (I), Grade (G), SAT (S), Letter (L) sunt variabile aleatoare:

$Val(D) = \{d^0, d^1\}, P(D = d^0) = 0.6, P(D = d^1) = 0.4;$

$Val(I) = \{i^0, i^1\}, P(I = i^0) = 0.7, P(D = i^1) = 0.3;$

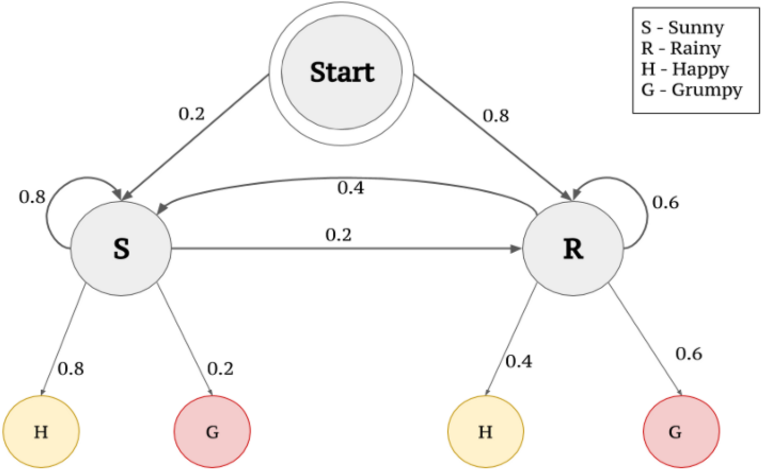
$Val(G) = \{g^0, g^1\}, P(G = g^0 \mid I = i^0, D = d^0) = 0.3,$

$P(G = g^1 \mid I = i^0, D = d^0) = 0.7, \ldots;$

$Val(S) = \{s^0, s^1\}, P(S = s^0 \mid I = i^0) = 0.95, P(S = s^1 \mid I = i^0) = 0.05, \ldots;$

$Val(L) = \{l^0, l^1\}, P(L = l^0 | G = g^0) = 0.9, P(L = l^1 | G = g^0) = 0.1, \ldots.$

Modelul „FSM”[3] - Hidden Markov Model



[3] Vivek Vinushanth Christopher. Markov and Hidden Markov Model. URL: https://towardsdatascience.com/markov-and-hidden-markov-model-3eec42298d75

Explicarea modelului

Vectori de variabile aleatoare: **State(t)**, **Obs(s)**, unde $t = 0, 1, 2, \ldots, s = S, R$,

$State(0) = Start$

$P(State(1) = S) = 0.2, P(State(1) = R) = 0.8$

$P(State(t+1) = S \mid State(t) = S) = 0.8$

$P(State(t+1) = R \mid State(t) = S) = 0.2$

$P(State(t+1) = S \mid State(t) = R) = 0.4$

$P(State(t+1) = R \mid State(t) = R) = 0.6$

$P(Obs(S) = H) = 0.8, P(Obs(S) = G) = 0.2$

$P(Obs(R) = H) = 0.4, P(Obs(R) = G) = 0.6$

Raţionament probabilist (exemple)

Fie (X_E, X_F) o partiţionare a variabilelor aleatoare din model.
Două exemple de inferenţe:

- probabilitatea marginală:

$$P(X_E = x_E) = \sum_{x_F} P(X_E = x_E, X_F = x_F);$$

- probabilitatea „maximum a posteriori” (MAP):

$$\begin{aligned} P^*(X_E &= x_E) = \max_{x_F} P(X_F = x_F \mid X_E = x_E) = \max_{x_F} \frac{P(X_E = x_E \mid X_F = x_F)P(X_F}{P(X_E = x_E)} \\ &= \max_{x_F} P(X_E = x_E \mid X_F = x_F)P(X_F = x_F) = \max_{x_F} P(X_E = x_E, X_F = x_F). \end{aligned}$$

- probabilitatea „verosimilitate maximă” (ML):

$$\max_{x_F} P(X_E = x_E \mid X_F = x_F).$$

Sistem de programare probabilistă

- modelul probabilist este descris de un program scris într-un limbaj de programare (DSL - limbaj specific domeniului);
- variabila aleatoare este descrisă de o variabilă program;
- sistemul pune la dispoziţie un set de algoritmi de inferenţă probabilistă peste modelele descrise ca programe;
- interogarea constă în calculul de valori ale variabilelor program.

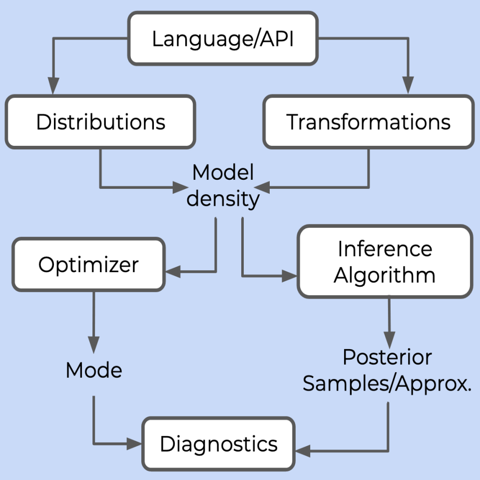


Figura: https://danmackinlay.name/notebook/probabilistic_programming.html

De ce programare probabilistă?

- expresivitate
  „raţionament probabilist + Turing-complete = programare probabilistă”
- raţionament probabilist mai bun
- simulare mai bună (execuţie + analiză)

Provocări

- ce limbaj gazdă utilizăm pentru DSL
- cum se reprezintă o variabilă aleatoare ca un element din limbaj
- cum se construiesc modele complexe
- ce algoritmi de inferenţă se implementează (eficienţă vs. acurateţe)
- scalabilitate: abilitatea unui sistem de a-şi creşte performanţa dacă acel sistem este dotat cu resurse suplimentare

Avantajele încorporării unui limbaj probabilist într-unul de programare general

- evidenţa poate fi descrisă în limbajul gazdă
- rezultatele date de sistem pot fi utilizate în programe
- se poate utiliza cod general în programe probabiliste
- se pot utiliza tehnici de programare generală (imperativă, funcţională, OO - object oriented) în scrierea de programe probabiliste

Listă (non-exhaustivă) de limbaje probabiliste:
`https://en.wikipedia.org/wiki/Probabilistic_programming`