

TEORIE ML

SĂPTĂMÂNA 1

- $P(A \cap \bar{B}) = P(A) - P(A \cap B)$; Bunătatea lui Bonferroni \Leftrightarrow
- $P(A \cap B) \geq P(A) + P(B) - 1$; A și B sunt abatorii \Leftrightarrow
- evenimentele A și B sunt abatorii \Leftrightarrow
- $P(A \cap B) = 0$; $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $\begin{cases} A \cap B \neq \emptyset \\ A \cap B = \emptyset \end{cases} \Rightarrow \begin{cases} P(A \cup B) = P(A) + P(B) \\ P(A \cup B) = P(A) + P(B) \end{cases}$; evenimente independente \Leftrightarrow condițională
- A și B sunt evenimente independente $\Leftrightarrow P(A \cap B) = P(A)P(B)$; (A este
- $P(\bar{A}) = 1 - P(A)$; A/B (A este
- evenimente condiționate $\Leftrightarrow A/B$ (A este
- condiționat de B); condiționat de B);
- $P(A|B) = \frac{P(A \cap B)}{P(B)}$; condiționat de B);
- $P(A|B,C) = \frac{P(A \cap B \cap C)}{P(B \cap C)}$; atunci $A+B=1$

- dacă atunci $A+B=1$
- $E[X] = 0 \cdot A + 1 \cdot B$ (media lui X); $E[X^2] - E[X]^2$ (varianța lui X);
- $V_{ar}[X] = E[X^2] - E[X]^2$ (varianța lui X);
- $P(X=x_1, Y=y_1) = P(X=x_1)P(Y=y_1)$; A și B sunt independenți
- $P(X=x_1, Y=y_1) \Leftrightarrow X=x_1 + Y=y_1$; + $E[X^2]$

- covarianta liniară $x_i + y_j \Leftrightarrow$
 ~~$\text{cov}[x, y] \text{ cov}(x, y) = E[xy] - E[x]E[y]$~~ ;
- dacă x, y sunt numai
 ~~$\text{cov}[x, y] = 0$~~ dacă și variabile
- $x_i + y_j$ sunt independente și numai
 dacă $\text{cov}(x, y) = 0$ (" \Rightarrow " întotdeauna)
- $x_i + y_j$ sunt binare dacă și numai
 dacă $\text{cov}(x, y) = 0$ (" \Rightarrow " întotdeauna,
 "⇒" dacă x, y binare);

SĂPTĂMÂNA 2

- entropia variabilă discrete +
- $H(x) = -\sum_i p(x=i) \log_2(p(x=i))$,
- $H(x)$ măs. $E[-\log_2(p(x))]$, considerăm
- dacă $p(x) = 0$, atunci
- $p(x) \log_2(p(x)) = 0$; (convenție);
- $\log_2 \rightarrow$ logă condițională specifică variabilei
- entropia condițională specifică variabilei
- $H(y/x=x_R) = -\sum_j p(y=y_j/x=x_R)$.
- $\cdot \log_2(p(y=y_j/x=x_R))$,
 $H(y/x=x_R) = E[y/x=x_R] [-\log_2(p(y/x=x_R))]$,
- entropia condițională medie a variabilei
- entropă variabilă + \Leftrightarrow
- entropă ca variabilă

$$H(x/t) = \sum_{x \in R} P(x=x_i) H(y/x) \quad ;$$

$$H(x/t) \stackrel{\text{def.}}{=} E_x[H(y/x)];$$

\rightarrow entropia comunica a variabili x, y

$$H(x, y) = -\sum_{i,j} P(x=x_i, y=y_j) \log P(x=x_i, y=y_j);$$

\rightarrow entropia = $E_{x,y}[-\log P(x, y)]$;

\rightarrow entropia de informație y de la x +

\rightarrow entropia de variabilă y de la y +

$$\text{gi. } H(y) = H(y) - H(y/x);$$

\rightarrow $H(y) \geq 0$; gi. măsură dezastru este

$$H(y) = 0 \quad \text{dacă gi. măsură dezastru este}$$

\rightarrow constante;

$$H(x/t) = -\sum_{i,j} P(x=x_i, y=y_j);$$

$$\cdot \log P(x=y_j, x=x_i);$$

$$H(x, y) = H(x) + H(y/x) = H(y) + H(x/y);$$

\rightarrow regulă de încreștere $\rightarrow H(x_1, \dots, x_m) =$

$$= H(x_1) + H(x_2/x_1) + \dots + H(x_m/x_1, \dots, x_{m-1});$$

$$H(x, t) = H(x) + H(t/x) = 1;$$

$$\rightarrow \sum_{y \in S} P(y/x=x_i) = 1;$$

- $\rightarrow P(x_1, \dots, x_n) = P(x_1) \cdot P(x_2 | x_1) \cdot \dots \cdot P(x_n | x_1, \dots, x_{n-1})$;
 → cantitatea de informatie obținută (surpriza) la observarea producării valoare x_i a unei variabile aleatoare X + sareacă \Leftrightarrow
 Informația $(P(X=x_i))$ = Surpriza $(P(X=x_i))$.
 $= \log_2 P(X=x_i) = -\log_2 P(X=x_i)$;
 → experiența (numărul) în baza de informație ;
- $\rightarrow H(X=x_i) = \sum_y P(x=x_i, Y=y)$;
 $H(X,Y) = \sum_{x,y} f_{X,Y}(x,y) \log \frac{f_{X,Y}(x,y)}{f_X(x) f_Y(y)}$;
 → $H(X,Y) = 0$ dacă și numai $\forall x, y$ sunt independenți ;
- $iG(X,Y) = H(X) - H(X|Y)$;
 → dacă X și Y sunt variabile aleatoare X și Y , atunci $iG(X,Y) = H(Y) - H(Y|X)$;
 → dacă X și Y sunt independenți , atunci $iG(X,Y) = 0$;
- $iG(X,Y) = H(X) + H(Y) - iG(X,Y)$;
- $iG(X,Y) \geq 0$;
- entropia relativă $H(X|Y)$ (diferența Kullback-Leibler) se raportează cu o altă distribuție ;

$\Leftrightarrow KL(p \parallel q) = - \sum_{x \in X} p(x) \log \frac{q(x)}{p(x)}$, unde
ambele distribuții sunt discrete;
 ↳ nr. de biti codificați care sunt
necesare în medie pentru a transmite
valoile variabilei X atunci când
poate sănușor să aceste valori sunt
distribuite conform distribuției q ,
deoarece este o următoare p

$$\begin{aligned} &\rightarrow KL(p \parallel q) \geq 0 \Leftrightarrow p = q; \\ &\rightarrow H(p, q) = KL(p_{+,+} \parallel (p_+ + p_+)) = \\ &= - \sum_{+} \sum_{+} p_{+,+}(+,+) \log \frac{p_{+,+}(+,+)}{p_{+,+}(+,+)} = \\ &= - \sum_{+} \sum_{+} p_{+,+}(+,+) \log \frac{p_{+,+}(+,+)}{p_{+,+}(+,+)}; \\ &\text{not.} \end{aligned}$$

$H(p, q)$ = entropia a 2 distribuții p și q \Leftrightarrow
 ↳ cross-entropie a 2 distribuții p și q ;
 $H(p, q) = \sum_{+} p(x) \log q(x);$
 nr. mediu de biti necesare pentru a
 codifica un eveniment dintr-o
 mulțime sau o varianță de posibilități atunci
 când schema de codificare folosește
 ca bază să se bazeze pe ceea
 ce datează în următoare (p, q sunt discrete);

- $\# g$ și g sunt continue \Leftrightarrow
 $C\#(f, g) = - \int_X f(x) \log g(x) dx;$
- $KL(f || g) = C\#(f, g) - \#(f);$
- $C\#(f \# g) \geq 0;$

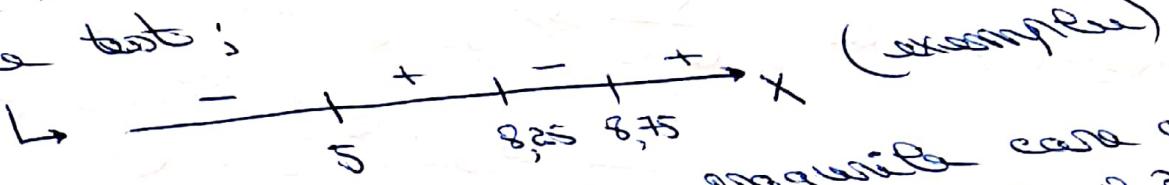
SĂPTAMANA 3

- arbore de decizie optim \Leftrightarrow este mai compact (m. de niveuri și de noduri)
- algoritmul ID3 \Leftrightarrow Greedy, divide-et-minimizam;
- algoritmul ID3 (nu produce cea mai bună soluție simplă);
 ↳ arboree produc uneori răspunsuri nesigure;
- ID3 pseudocod:
 - create the root node;
 - assign all examples to root;
 - START;
 - A \Leftrightarrow the "best" decision attribute for next node;
 - for each value of A, create a new descendant of node;
 - sort training examples to leaf nodes;
 - if training examples perfectly classified, STOP, else iterate over new leaf nodes;

- multiplexarea / colectarea unor instante (instante din tabel) \Rightarrow schimbarea rezultatului algoritmului ID3 (daca nu este robust);
- atributul identificator (colectare din tabel) \Rightarrow atributul arborelui de decizie (este robust);
- atributul de instantă sunt categoriale (sunt un nr. finit) \Rightarrow aduncarea maximă a unei instante este nr. de atribut și de arbori de decizie;
- instanță \Leftrightarrow instanță care are o etichetă corectă; exemplu \Leftrightarrow instanță pozitivă (daca este pozitivă);
- $\Leftrightarrow \langle x, c(x) \rangle$, $x \in X$, $c \subseteq X$, $c: x \rightarrow \{0, 1\}$;
- instanță $\Leftrightarrow f: x \rightarrow \{0, 1\}$
- instanță consistentă $\Leftrightarrow f(x) = c(x)$, $\forall \langle x, c(x) \rangle$;
- instanță \Leftrightarrow nr. de instanțe pozitive și negativă clasificate corect;
- set de date pentru a verifica corectatea;
- set de date pentru a verifica stabilitatea;
- set de date pentru a verifica de către care este rezultatul ID3 \Leftrightarrow toti, arborii de decizie care pot fi generati cu același nr. de instanțe în meduriile de test și valori ale atributului să decidă în același moduri de decizie;

- specificitate de cearcare mare \Rightarrow ID3 nu poate să lucreze exhaustiv;
- \hookrightarrow și sugestarea specifică de căutare la fiecare pas;
- algoritmul ID3 \Rightarrow se obține varianta optimă pentru fiecare nod din funcție de contingență de informație (îl luăm pe cel mai mare la fiecare pas);
- expresivitatea arborilor de decizie (\Leftarrow unele funcții booleane de decizie exprimate ca jocuri de cel puțin la fel de mare ca cea a jocurilor binare);
- clasificarea ternară \Leftrightarrow ultima clasă din tabel are 3 clase (de exemplu $y \in \{1, 2, 3\}$);
- atribut continuu \Leftrightarrow un atribut care să împartă spațiul în trei secțiuni și să leasă ca ID3 poate să extindă (ms. scale) folosind atributul continuu (ms. scale) respectiv;
- discretizarea atributului numeric respectiv;
- testele se vor face cu acest atribut în raport cu ceva (de exemplu $T \in [30, 5)$);
- atributul continuu va fi apoi de statat și este să fie mărginit astfel;
- condiționalitatea specifică $\Leftrightarrow H(y|x=x_0)$;
- entropie condițională maximă \Leftrightarrow entropie maximă de informație;

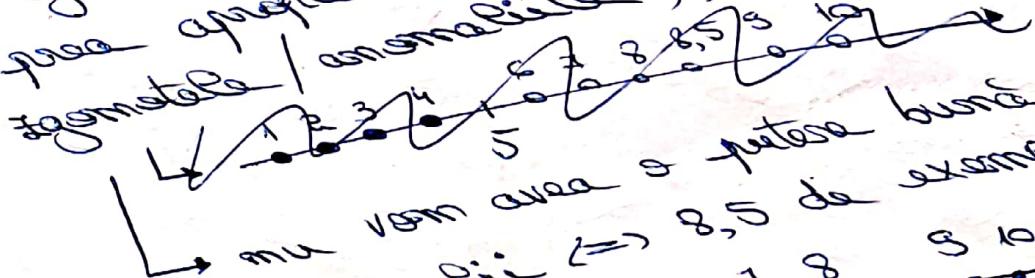
condiționale media minimă }
→ zone / suprafețe de decizie \Leftrightarrow o axă pe care
se vede clar că decizia cu cea mai mare
vici test:



esperanță de decizie \Leftrightarrow probabilitatea ca se operează
pe zonele de decizie (de exemplu 5; 8,25;
8,75);

menține leave one out \Leftrightarrow teste datele mai
fuzionare (de exemplu 5 și 8,75) și instanță în afara ei;
m. de instanță la care decizia este
superba (de exemplu 6) $\Leftrightarrow \frac{1}{6}$;

fenomenul de overfitting \Leftrightarrow modelul este
prea apropiat (mult) de date (explică

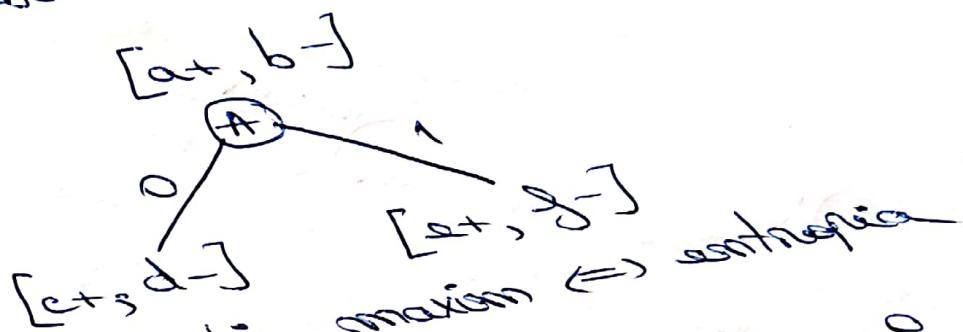


nu vom avea o poteră bună de generalizare;

zgomote / anomalii \Leftrightarrow 8,5 de exemplu;
leave one out \Leftrightarrow CV LOO ;
cross validate \Leftrightarrow LOOCV ;
leave one out cross validate \Leftrightarrow CV CV ;

SĂPTĂMÂNA 4

- $H[a+, b-] = \frac{1}{a+b} \log_2 \frac{(a+b)ab}{a \cdot b^b}$ (a, b ≠ 0)
- $H_{node/attribute} = \frac{1}{a+b} \log_2 \frac{(c+d)c+d \cdot (e+f)e+f}{(c+d)^2 \cdot (e+f)^2}$
- $H_{node/attribute} = \frac{1}{a+b} \log_2 \frac{(a+b)ab \cdot (c+d)cd \cdot (e+f)ef}{a \cdot b \cdot (c+d)^2 \cdot (e+f)^2}$



- cantitate de informatie minima;
- condițională media minimă;
- este suficient să găsim minimul
- diție multușă
- diminuare se bucură ce e mai
- atunci ead se numește
- multă atribut contine, propunile trebuie
- granițele de decizie iterativă;
- călcătă la fiecare iterată;
- de coordonate;
- pentru a se obține zonele de
- decizie care nu sunt ferite
- decizie care nu sunt ferite
- coordonate (de decizie determinante de clasificare)
- fiecare de decizie determinante de clasificare
- diferite, nu sunt neapărat identice;
- index Gini $\Leftrightarrow Gini(g) = 1 - p_a^2 - (1-p_a)^2 = 2p_a(1-p_a)$

- impuritatea la clasificarea imprecise \Leftrightarrow
 $i(S) = 1 - \max_{i=1}^k P(c_i)$, unde c_i este k clase
 c_1, c_2, \dots, c_k ;
 ↓ indexul binii este $i(S) = 1 - \sum_{i=1}^k P(c_i)^2$;
- drop-off-impurity ($\Rightarrow \Delta i(S) = i(S) - i(S_A) - i(S_B)$)
 - $P(c_i) i(S_A)$, unde c_i și c_j sunt
 descendentele din stanga, respectiv dreapta
 ai lui c_i ;
 ↓ echivalent cu costul de informatie
 (drop-off-impurity \Leftrightarrow descendentele
 impurității);
- dacă suntem în linie, putem să să luăm
 tabel (pe linie), numai frecvența (de
 către valoare corespunzătoare, unde este valoarea
 pe care o să o săptămăriam), respectiv, numai frecvența
 lipsă) sau ca valoare să săptămăriam
 care apare în exemplu din cursă suntem
 clasă (cela care nu are rezultatul
 ca rezultatul unei attribute);
- attribute ca costuri $\Leftrightarrow \text{Gain}(S, A)$;
- attribute ca foarte multe valori \Leftrightarrow
 $\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInformation}(S, A)}$;
- $\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{-\sum_{i=1}^k \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}}$
 unde SplitInformation este o subvenție a lui S pentru
 care A este o venție și

- $\text{Gain}(S, A) = \text{IG}(S, A)$;
- set de date inconsistent \Leftrightarrow avem 2 linii egale in tabel care au variabile de tipul (eticheta) diferite (o inconsistentă); ID3 suferă de overfitting
- ~~partea stării~~ (explică rezultatul);
- $\text{error train}(h) < \text{error train}(h')$; $\left\{ \begin{array}{l} \text{DT1} \\ \text{DT2} \end{array} \right.$ ex. 10 (def. overfitting)
- $\text{error train}(h) > \text{error train}(h')$; $\left\{ \begin{array}{l} \text{DT1} \\ \text{DT2} \end{array} \right.$
- $\text{error}(h) * \text{error}(h')$; $\left\{ \begin{array}{l} \text{DT1} \\ \text{DT2} \end{array} \right.$ \Leftrightarrow mărește la conținut;
- $\text{error train} \Leftrightarrow$ mărește CVLOO (cross-validation)
- $\text{error} \Leftrightarrow$ mărește de valoare trebuie leave-one-out); mărește atât de date de valoare de valoare;
- mărește atât de date de valoare de valoare; se dă destinația factorului overfitting \Leftrightarrow acuratețea se mărește și se mărește la valoare
- reprezentarea grafică crește și se mărește la conținută se mărește;
- se mărește la ID3 (pentru a combate se mărește);
- prevenirea ID3 (pentru a combate se mărește);
- prevenirea ID3 (pentru a combate se mărește);

SĂPTĂMÂNA 5

- overfitting \leftrightarrow supraspecializare;
- punting la arborii ID3 \leftrightarrow top down sau bottom up (conform dict fizică);
 - ↳ DT slides, ex. 19, slide 60;
- punting top down (conform fizică) este în general mai eficient decât cel bottom up (baștă \rightarrow testul $\leq \epsilon$)
 - dezavantajul este că este mai mult, se elimină date ce eliminătoare sunt nec, se elimină tot subarborele respectiv dacă nu există măsură (atribut) care să poată de predictie (cu $\leq \epsilon$ margină);
 - ↳ puntingul bottom up nu are acest dezavantaj;
- dezavantajul puntingului bottom up este că la maturitatea case sunt sănătoase celor înainte în general (cantitativ, sunt mici (au putință exemplu de asociat)) \Rightarrow decizia respectivă relativă la $\leq \epsilon$ nu este neapărat corectă din punct de vedere statistic;
- ↳ adeseori în practică se aplică testul χ^2 ; astăzi, nu $\leq \epsilon$ (testul χ^2);
 - ↳ DT slides, ex. 20, slide 68;

- $\rightarrow \chi^2 = \sum_{i=1}^n \sum_{j=1}^r \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$, unde O este matricea de observatii si E este matricea de asteptare;
- \rightarrow formula lui Bayes $\Leftrightarrow P(B|A) = \frac{P(A|B)P(B)}{P(A)}$;
- $\hookrightarrow P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B')P(B')}$
- $\rightarrow P(A|B) = \frac{P(AB)}{P(B)}$
- \rightarrow formula probabilitati totala $\Leftrightarrow P(A) = P(A|B)P(B) + P(A|B')P(B')$;
- $\hookrightarrow P(A) = \sum_i P(A|B_i)P(B_i)$
- \rightarrow evenimente independente $\Leftrightarrow P(AB) = P(A)P(B)$;
- $\hookrightarrow P(B) \neq 0 \Leftrightarrow P(A|B) = P(A)$; independent conditional $\Leftrightarrow P(AB|C) = P(A|C)P(B|C)$, $P(C) \neq 0$; independent $\Leftrightarrow P(ABC) = P(A|B,C)P(B|C)P(C)$
- $\hookrightarrow P(BC) \neq 0 \Leftrightarrow P(A|B,C) = P(A|C)$; independent comun $\Leftrightarrow P(x,y) = P(x=x, y=y)$ = distributie de probabilitate comună;
- $P(x,y) = P_{x,y}(x,y) = P(\omega \in \Omega | x(\omega) = x, y(\omega) = y)$; $P_x(x) = P(\omega \in \Omega | x(\omega) = x)$ = probabilitatea marginala;
- $P_x(x) = \sum_y P(x,y)$; $P_x(y) = \sum_x P(x,y)$; distributie de probabilitate continua $P_x(x) = \int p(x,y) dy$; variabila
- $P_x(y) = \int p(x,y) dx$;

- distribuție de probabilitate conditională
- ↔ $f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$;
- variabilă obiectivă independentă $\Leftrightarrow f_{X,Y}(x,y) = f_X(x) \cdot f_Y(y)$ (X, Y sunt marginale ale lui (X, Y));
- variabilă obiectivă conditională independentă $\Leftrightarrow f_{X|Y}(x|y) = f_{X|Y}(x|y/x)$;
- variabilă obiectivă totă valoarea $\cdot f_{Y|X}(y|x)$ (restulă a posteriori);
- possible x, y, z din X, Y, Z ; ~~probabilități~~ ~~a posteriori~~ Probabilități;
- MAP \Leftrightarrow Maximizare $f_{MAP} \Leftrightarrow f_{MAP} = \operatorname{argmax}_{f \in \mathcal{H}} P(f|D) =$
- ipoteza $\frac{P(D|f)P(f)}{P(D)}$ $= \operatorname{argmax}_{f \in \mathcal{H}} P(D|f)P(f)$;
- slides Bayes, ex. 3, slide 45;
- \Rightarrow setul de date;
- argmax \Leftrightarrow argumentul maximul expresiei;
- atât de Bayes nu se poate presupune, că este clasificatorul Bayes Naïf (naiv) independent de variabilele independente condiționale, iar cel de comunitate (Bayes Comunitar) nu folosește aceasta presupunere;
- slides Bayes, slide 45;

- Bayes Naïv \Leftrightarrow se poate formula într-o formă de independență condițională și condiția de independentă condițională este să o verificăm pentru fiecare variabilă.
- Bayes Comun \Leftrightarrow se poate formula probabilitatea unei multe date de comun;
- Bayes Naïv \Leftrightarrow are nevoie de mai multă date de antrenament (mai multă faza teoretică), etichetează variabilele de decizie v_1, v_2, \dots, v_n și a_1, \dots, a_m
- Bayes Naïv \Leftrightarrow $P(v_1, v_2, \dots, v_n | a_1, \dots, a_m) = P(v_1 | a_1) \cdot P(v_2 | a_2) \cdot \dots \cdot P(v_n | a_n)$
- ← este că valoarea deciziei v_i , atunci dacă este probabilă valoarea finală $f(+)$ este
- \checkmark MAP = $\underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, \dots, a_m) =$
- = $\underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, \dots, a_m | v_j) P(v_j)}{P(a_1, \dots, a_m)} =$
- = $\underset{v_j \in V}{\operatorname{argmax}} P(a_1, \dots, a_m | v_j) P(v_j) =$
- = $\underset{v_j \in V}{\operatorname{argmax}} P(a_1 | v_j) P(v_j) =$ \checkmark NB;
- = $\underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, \dots, a_m) =$
- ← regulă de decizie pentru Bayes Naïv:
- Bayes Comun \Leftrightarrow $v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, \dots, a_m) =$
- = $\underset{v_j \in V}{\operatorname{argmax}} P(a_1, \dots, a_m | v_j) P(v_j) =$

$$= \underset{v_j \in V}{\operatorname{argmax}} \varphi(v_1, \dots, v_m, v_j) = v_{jB}^*$$

- dacă folosind clasificatorul Bayes Naïf obținem un alt rezultat față de celul în casă, acesta este Baye comun, atunci independența condițională nu este satisfăcută și probabilitatea (chiar și în cazul favorabil), căruia sunt egale);
- căruia favorabil \Leftrightarrow metoda estimării cauzale maxime;
- Bayes. Bayes. ex. (picătă);
- regula lui Laplace, add one \Leftrightarrow la numărator adăugăm probabilitatea de niciună parte, de exemplu;
- $\text{if } \varphi(\text{+1}) = \frac{\varphi(+)}{m+1}, \varphi(-)}$;
- $\varphi(+1) = \frac{\varphi(+)}{m+1}, \varphi(-)$;
- Bayes Comun \Leftrightarrow Bayes optimal;

SĂPTĂMÂNA 6

- Bayes comun lucruare cu datele reale și nu face presupozitii \Rightarrow se mai menține;
- Bayes Optimal;
 - ✓ NB se poate nota $J_{NB}(x_1=x, x_2=y)$;
 - ↳ sfidere Bayes, slide 29;
 - dacă independență condițională nu este satisfăcătoare (probabilitate / rezultatul lui Bayes Naïv diferență de cel al lui Bayes Optimal), atunci vom trebui să demonstrează că cele două variabile nu sunt independență condițională, deci suntem în stare să obținem o clasificare explicită;
 - Proiecționarea;
 - parametrii independenți \Leftrightarrow ceea ce sunt liberi;
 - stimație din $\hat{P}(X) = 1 - \hat{P}(\bar{X})$; unde, din punct de vedere probabilistică condițională, variabilă X este independentă de variabilele x_1, x_2, \dots, x_n și este constantă;
 - $\hat{P}(A|B) = 1 - \hat{P}(\bar{A}|B)$; unde, din punct de vedere probabilistică condițională, variabilă B este independentă de variabilele x_1, x_2, \dots, x_n și este constantă;
 - $\hat{P}(x_i|Y), i=1, m \Rightarrow m+1$ parametrii liberi / valori pentru Bayes Naïv (complexitatea modelului) (i);

- este atribut de intrare $\Rightarrow P(Y)$, $P(X_1, \dots, X_n | Y)$
 $\Rightarrow P(\tilde{x}_1, \dots, \tilde{x}_n | Y)$, $x_i \in \{x_i, \tilde{x}_i\}$, $i = 1, \dots, n$
 $\Rightarrow n+1$ valori / parametri liberi pentru
 Bayes Optimal (2);
- ↳ nr. de parametri necesari din date;
 nr. de puncte variabile / atribut
- (1) și (2) sunt puncte variabile / atribut
 binare (Bernoulli);
 $P(X_1, X_2, Y) = P(X_1|Y) \cdot P(X_2|Y) \cdot P(Y)$, dacă
 acele două sunt independență condițională;
 dacă și este binar (0 sau 1 , + sau - etc.),
 atunci $P_{NB}(Y|X_1, X_2) = \frac{q_0}{q_0 + q_1}$ sau $\frac{q_0 + q_1}{q_1}$
 (în funcție de o valoare obținută clasificatorul
 Bayes NB(X_1, X_2), să fie 0 , q_1 punctul);
 ↳ sfidă Bayes, sfida 50;
 nota medie a probabilității erorii $\Leftrightarrow \text{eror} = E_{\bar{x}}[1 \cdot Y_{NB} + 1 \cdot \bar{y}]$
 ↳ $\text{eror} = \sum 1 \cdot Y_{NB} + 1 \cdot \bar{y}$
 este o condiție (aceea sumă
 probabilităților calculate de
 clasificator Bayes unde $Y_{NB} \neq \bar{y}$);
 nota medie a probabilității corespunzătoare
 probabilității se prezintă (se calculează în
 clasificator de distribuție sau a datelor);
 funcție de

- distributie, distributie conditioanala
- Bayes Naiv \Rightarrow Bayes Naiv corespunde cu scris a datelor \Leftrightarrow independenta este respectata;
- Bayes Naiv \Rightarrow Bayes Naiv este (comuna) legitul or vorul majoritor de minoritate;
- este imposibil minoritate;
- Bayes Naiv mai este posibila independenta conditioanala \Rightarrow este verifica; $\pi_{\text{noi}} \leq 0,5$;
- nota media a π_{noi} se calculeaza din nota medie a π_{noi} chiar si atunci functie de distributie sunt chiar (si rezultatul cond probabilitatea degena (\Rightarrow rezultatul dupa probabilitatea stimulata \Rightarrow rezultatul dupa Bayes Naiv));
- ocazaticea la clasificare a obiectului Bayes Naiv scade atunci cand unul sau mai multe atributi sunt duplicate (\Rightarrow adaugati si unele conditii de independenta conditioanala);
- ↪ ID3 nu este sensibil la duplicitatea atributorilor;
- nota media a π_{noi} maxima ($= 0,5$) \Leftrightarrow dependenta conditioanala maxima;
- ↪ slides Bayes, slide 57;

- XOR cu Bayes Naïf produce rate media a eroarei maxime, adică 0,5 (nu sugerează XOR perfect; învăță), în cînd ID3 învăță XOR perfect;
- în practică, aplicarea algoritmului Bayes Naïf este precedată de o procedură de feature selection (selecție atributelor) și carea scop este să elimene / reducă dependențele conditionale dintre atributi în raport cu atributul de ieșire;
 - ↳ sfidă Bayes, slide 62;
- în cînd învățării unei funcții binecunoscute (carecace), rate media a eroarei produce la cîndrența de către Bayes Naïf descrierea de Bayes Naïf este 0;
- ↳ sfidă Bayes, slide 66;

SĂPTĂMÂNA 7

- complexitatea de extindere \Leftrightarrow de ordin logarithmic pentru Bayes Naïf și de ordin exponential pentru Bayes Optimal (doar Bayes);
- ↳ m. de exemplu de extindere (doar Bayes);
inventă, modelabil (complexity);
- ↳ sfidă Bayes, slide 73;
- suggerie \Leftrightarrow aproximarea de funcție de probabilitate;

→ formula probabilității totale \Leftrightarrow
 $P(X, Y) = \sum_{Y \in Y} P(X=x | Y=y) P(Y=y)$;

→ $e^{\ln x} = x$ ($\exp(\ln x) = x$);

↪ funcția exponentială și ea logarithmică sunt inverse unele pentru celelalte;

→ $x = x \Leftrightarrow x_1 = x_1, x_2 = x_2, \dots, x_d = x_d$ (explicătoare);

→ $P(Y=1 | X=x) = \frac{1 + \exp(w_0 + \sum_{i=1}^d w_i x_i)}{1 + \exp(w_0 + \sum_{i=1}^d w_i x_i)}$, unde

$$w_0 = \ln \frac{1-p}{p} + \sum_{i=1}^d \ln \frac{1-\theta_{i0}}{1-\theta_{i1}}$$

$$w_i = \ln \frac{\theta_{i0}}{\theta_{i1}} - \ln \frac{1-\theta_{i0}}{1-\theta_{i1}}, \forall i=1, d;$$

↪ se aplică Bayes, să se rezolve ecuația

→ dreptatea $w_0 + w_1 x_1 + w_2 x_2 = 0$;

→ se scrie în spatiul euclidian ecuația

$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 = 0$;

→ se scrie hiperplan în R⁴ din ecuația

$w_0 + w_1 x_1 + \dots + w_{m+1} x_{m+1} = 0$;

→ se aplică decizional determinat de Bayes

Nau este de tip liniar ($w_0 + \sum_{i=1}^d w_i x_i = 0$);

deosema $w_0 + \sum_{i=1}^d w_i x_i = 0$;

↪ instance-based learning \Leftrightarrow convoluție bazată pe memoria;

- k -NN \Leftrightarrow algoritmul celor mai apropiate \leftarrow vecini; \leftarrow antrenare \Leftrightarrow monozade (sotschadire)
- lista de datele de antrenament;
 - \hookrightarrow algoritmul k -NN \Leftrightarrow geometrie / stabilitate și mai apropiate \leftarrow vecini (sau multe k -NN vecinătate), și stabilitate eticheta majoritară dintr-o vecinătate de test;
 - $\hookrightarrow \hat{y}(x) = \arg \max \sum_{i=1}^k y_i(x_i) = \frac{\sum_{i=1}^k y_i}{k}$ (pentru variabilă discretă); $\hat{y}(x) = \frac{\sum_{i=1}^k y_i}{k}$ (pentru variabilă continuă); k este de obicei impar pentru a evita cazul în care avem m. egal de vecini;
- rețeaua lui R nu se schimbă;
- outputul algoritmului k -NN pentru o instanță concreta de test se definește de valoarea lui k ;
- ID3 este binecunoscut ca puternică;
- este că ID3 este inductiv, folosește atributul de intrare ca să considerăm că atributul de ieșire este independentă condițională față de atributul de ieșire;

- bisecul inductiv pentru algoritmul k-NN este luptul ca o instanță de test primește ca etichetă eticheta majoritară a celor mai apropiate k vecini ai săi (k-NN vecinătate);
- ↳ "căci nu sună departe de trunchi";
- "cine se ascundem, se adună";
 "spune-mi ce cine te împrietenești ca să stiu cine este"; "like father, like son";
- înseamnă la antrenare pentru k-NN \Leftrightarrow luăm fiecare instanță din setul de date și o clasificăm bine, atunci avem eroare (eroră la antrenare este $\frac{e}{n}$), unde e este nr. de căsi și n nr. total de instanțe de antrenament);
- ↳ sună $\frac{e}{n} = \frac{1}{k}$, unde k este nr. m. total de vecini proprii vecini pentru k-NN este
- al mai apropiat vecin respectiv funcție care numărul vecinilor dăce avem eroare;
- nu al mai apropiat vecin (distanță = 0);
- de datele de antrenament este proprietatea că toate sunt consistentă, înseamnă că antrenarea produce de k-NN este o funcție care măsoară de distanță;

- $d: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \Rightarrow d(x, y) \geq 0, \forall x, y \in \mathbb{R}^n$ (non-negativitate),
 $d(x, y) = 0 \Leftrightarrow x = y$ (identitatea indiscutabilă),
 $d(x, y) = d(y, x)$ (simetria) și
 $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in \mathbb{R}^n$
 (inegalitatea triunghiului);
- distanța euclidiană $\Leftrightarrow d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \forall (x_1, y_1), (x_2, y_2) \in \mathbb{R}^n$;
- featură de date puternic mixte, k-NN da
- featură mai la CMLD; media și medie a variației \Leftrightarrow când avem distribuții probabilitățile;
- zone de decizie pentru k-NN \Leftrightarrow diagramă Vennuri \Leftrightarrow unim funcția punctelor cu etichete dințite și tragem mediatoarele;
- subiectele de decizie și separatoare decizionale supraduale de distanță Manhattan $\Leftrightarrow |x_1 - x_2| + |y_1 - y_2|$, depend de măsura de distanță Manhattan $\Leftrightarrow (x_1, y_1), (x_2, y_2)$ (indusă de punctul L_1);
- fizica, noțiile scorante: iBL. ex. 2021f
 (1-NN este de 21-NN);

SĂPTĂMÂNA 8

- "baza multiori dimensiunii" (the curse of dimensionality) \Leftrightarrow în anumite condiții, nr. de instanțe de antrenament necesare pentru a avea un algoritm practic de distanță, crește exponențial cu numărul de teste și crește exponential funcția de nr. de atributuri folosite (în contextul algoritmului k-NN);
 - ↳ slides iPSL, slide 45;
- cunoașterea practică a "bazelor multiori dimensiunii" \Leftrightarrow înainte de a optița algoritm R-NN se recomandă să se selecție o procedură de feature selection care să reducă nr. de atributuri folosite;
 - ↳ la Bayes Naïv, feature selection dependă de distribuția atributelor;
 - ↳ este posibil să se obțină o selecție practică a modelului de clasificare;
- cross validație;
 - ↳ AdaBoost \Leftrightarrow adaptive boosting;
 - în date de antrenament $s = \{(x_1, y_1), \dots, (x_m, y_m)\}$, $x_i \in \mathbb{R}^d$ și $y_i \in \{-1, +1\}$, la fiecare iterație avem o distribuție de probabilitate, pentru fiecare i avem $\pi_i(i) = \frac{1}{m}$, $i = 1, m$, și iterată, $t = \overline{1, T}$, la fiecare iterație facem

$$\text{update la distributie } D_{t+1}(i) = \frac{1}{\Delta t} D_t(i) \cdot e^{-\Delta t f_t(x_i)}, \quad i=1, m$$

$f: X \rightarrow \{-1, +1\}$ ipoteza, $\Delta t = \frac{1}{n} \text{ sau } \frac{1 - E_t}{E_t}$

E_t normalizator, $E_t = \sum_{i=1}^m D_t(i) = \frac{1}{n} - \Delta t$

f_t la expert $H_T = \operatorname{sign}\left(\sum_{t=1}^T \Delta t f_t\right)$ ca sistem de vot (vot majoritar)

algoritmul probabilist, iterativ si asamblat (la fiecare iteratie, se identifica cate o ipoteza f_t care sunt state);

E_t este o masura ponderata la contributie a ipotezei f_t ($E_t = \frac{1}{n} + \Delta t [y + f_t(x)]$), $E_t < \frac{1}{n}$;

$\Delta t \Leftrightarrow$ importanta ipotezei f_t (un vot);

$\sum_{t=1}^T \Delta t f_t \Leftrightarrow$ comitet de experti;

algoritmul creaza asamblarea consumului de clasificatori si-i aggregati;

"Adaboost" vine de la futura;

putem asambla clasificatorii slabii si sa obtinem unul bun (cu futura surgingtime);

generalizare \Rightarrow reduce surgingtime;

slides DT, slide S1;

- AdaBoost produce rezultate diverse atunci când oca posibilitate să aleagă între 2 sau mai multe (cele mai bune) ipoteze "slabe";
- $\pi_{t+1}(h_{t+1}) = \frac{1}{2}$ (ht nu poate fi selectată în iterația t+1, ea poate fi selectată în o iterație ulterioară);
- efectul de normalizare se obține ca suma probabilităților din distribuție să fie 1;