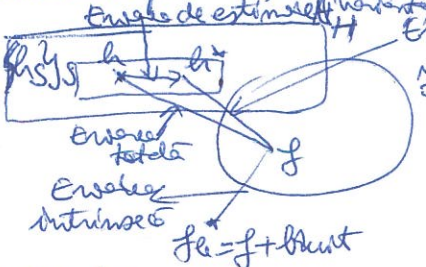
(CURS)

Învățare supervizată

Componentă deplasare-dispersie; f (din F = familia teoretică a tuturor funcțiilor posibile care generează datele) este o funcție de dependință a valorii introduse x și a etichetei y; h (din H = spațiul practic / model de funcții în care căutăm ipoteze) este o funcție care se diferențiază mai apropie de f; h^* = funcția optimă din H (cea mai apropiată de f). Eșantion de învățare supervizată: $S = \{z_1 = (x_1, y_1), \dots, z_m = (x_m, y_m)\}$; h_S = funcție optimă obținută în baza învățării supervizate S, w = ponderi; deplasare = diferență dintre F și H, din care H nu conține f. Varianță = dispersia ipotezelor h_S (estimărilor) în jurul soluției ideale h^* și crește odată cu flexibilitatea spațiului H. Varianță mare = predicțiile variază mult de la eșantion la eșantion. Overfitting: modelul învață și particularitățile întâmplătoare (noise-ul) ale eșantionului de antrenament, nu doar structura reală generată de f. Generalizare slabă: performanță bună pe datele de antrenament, dar slabă pe date noi. Zgomot / eroare intrinsecă = erori de măsurare, transmitere, uneori etc. f pierdută = f + noise; Compromisul deplasare-dispersie este alegerea unui H suficient de flexibil, încât să reducă deplasarea, dar nu atât de bogat încât să explodeze dispersia.



Eșantion: pentru a reduce deplasarea, H trebuie să fie cât mai apropiat de f. Cu cât H e mai mare, cu atât crește dispersia și eroarea totală implicit. În eșantion, eroare deplasare și dispersie nu pot fi reduse $\Rightarrow$ multe; Risc empiric ($R_{emp}(h)$) = performanța pe datele de antrenament,

$$R_{emp}(h) = \frac{1}{|S|}\sum_{i=1}^{|S|} l(u_i, h(x_i, w))$$

↓ funcție cost (loss) = măsoară costul pe care-l implică ieșirea dorită $h(x_i)$ în loc de u_i.

Principiul ERM

- găsește ipoteza h^* care minimizează riscul empiric (eroarea de antrenare)

$$h^*_{S,H} = \arg\min_{h \in H} R_{emp}(h); \quad R_{emp}(h) = \frac{1}{|S|}\sum_{i=1}^{|S|} l(u_i, h(x_i, w)).$$

- află parametrii w, care minimizează riscul empiric folosind funcție de cost

$$l(u_i, h(x_i, w)) = \sum_{i=1}^{|S|} (u_i - h(x_i, w))^2$$

Funcție de cost generică $l(y, a) = \begin{cases} 0, \text{ dacă } a = y \\ 1, \text{ dacă } a \neq y \end{cases}$.

Acă: $l(u_i, h(x_i, w))$.

$$l(u_i, h(x_i, w)) = \sum_{i=1}^{|S|} (u_i - h(x_i, w))^2$$

$$l(u_i, h(x_i, w)) = \sum_{i=1}^{|S|} |u_i - h(x_i, w)|$$

Risc real = performanța estimată pe orice date noi din același distribuție $\Rightarrow$ trebuie redus la minim, dar pentru că nu știm f, deci toate erorile au același pondere, atunci riscul real estimat = (suma falsurilor deci se calculează doar pe S (setul de antrenament), este t). Riscul empiric estimează un real. Fenomenul overfitting = adâncirea corectă găsite / adâncirea totale reale. $\rightarrow$

ERM = empirical risk minimization (selectează modelul care dă cea mai mică eroare pe setul de antrenament). Fenomenul Risk Minimization (SRM):

$$Risc_{real}(h) \leq Risc_{emp}(h) + ComplexityPenalty(H, m, \delta)$$ unde m = mărimea de exemple din setul de antrenament (teoreme de tip VC-dimensiunea Vapnik-Chervonenkis).

Definiție: Problema modelului de selecție constă în alegerea, în baza unui eșantion de date de învățare S, de lungime m, o clasă de funcții ipoteză H_d și o ipoteză h_d din H_d, a.î. riscul real cu cost $R_{real}(h_d)$ e minim.

Problema estimării riscului real din riscul empiric se reduce la a-i adăuga o penalizare de complexitate cale:

1. Crește odată cu „mărimea" sau flexibilitatea lui H (relativ constant sau cu o accelerație). În același timp, riscul empiric scade la început mai rapid, apoi din ce în ce mai lent, astfel că suma ne formă un grafic în formă de U.
2. Scade odată cu numărul de exemple de antrenament m
3. Depinde și de nivelul de încredere δ (delta mic). În practică, pentru experimente statistice adesea se ia $\delta \in \{0,05; 0,01; 0,001\} \Leftrightarrow$ % de încredere $= 1 - \delta$

Metode propuse de rezolvare a calculării riscului real, pornind de la cel empiric (SRM):

- The regularization theory (adaugă un termen de penalizare direct în funcție obiectiv (loss) pe care o optimizează)
- Minimum Description Length Principle (MDLP) (orice model plus date poate fi și codificat

ce un șir de biți. Modelul bun e acela, care împreună cu datele are cea mai mică lungime totală).

- Akaike Information criterion (AIC) — (AIC oferă o aproximare a pierderii de informație, atunci când modelul tău este folosit pt a aproxima adevărata distribuție de generare; alegi modelul cu cea mai mică valoare AIC: modele cu mulți parametri sunt taxate, iar modelele care "prind" bine datele sunt răsplătite).
- Validare încrucișată (împarți setul de date în k subseturi de dimensiuni (aproxim.) egale. Antrenezi modelul pe toate subseturile în afară de cel selectat și evaluezi eroarea pe acesta. Calculezi apoi media erorilor obținute pe cele k subseturi ⇒ estimare a riscului real; Avantaj: folosește eficient toate datele pentru antrenament și test; Problema este costul computațional f mare; de obicei N (de câte ori luăm câte 1 subset) este între 5–10.

$$\hat{R}_{real}(h) = \frac{1}{N}\sum_{i=1}^{N} \hat{R}_{real,i}(h)$$

O variantă extremă a validării încrucișate este „lasă-una-afară" (leave-one-out), și este utilă când setul de date este sărac. Practic, se păstrează un exemplu pt test și se antrenează individual pt fiecare din toate celelalte exemple. Rezultă un H cu varietate mai mică, este mai simplu de implementat și mai rapid, dar estimarea ratei de eroare variază mai mult decât în metoda standard, care mai mare.

- Bootstrapping (din setul original de m exemple, generezi B noi seturi de antrenament S_b, extrăgând aleator m exemple din total (unele de mai multe ori, unele niciodată). Pt fiecare set eșantion: antrenezi modelul, evaluezi performanța lui pe exemplele care nu au fost selectate (în medie, aproxim. 36,8% din date) și calculezi media erorilor peste cele B re-eșantioane. ⇒ estimare a riscului real; Avantaj: funcționează bine și cu date puține, estimează bias-ul și varianța estimatorului; Problema este costul computațional mare și poate fi instabil dacă distribuția e dezechilibrată).
- Media Bayesiană (considerăm fiecare model ca o ipoteză cu o probabilitate inițială, apoi o ajustăm în sus sau în jos în funcție de cât de bine se potrivesc pe datele de antrenare. Această probabilitate actualizată devine greutatea cu care participă la predicția de ansamblu).
- Metode bag (generăm B seturi de antrenament cu m elemente prin extragere repetată, antrenăm B modele separat pe fiecare set și facem media aritmetică a rezultatelor modelelor);
- Metode boosting (folosim procese de antrenare pentru fiecare exemplu la o clasă, de obicei 1 sau −1; folosim T astfel de procese ("weak learners"), care corectează greșelile precedentului; le atribuim ponderi inițial egale; le antrenăm pe fiecare; calculăm eroarea = suma ponderată a tuturor proceselor; calculăm ponderea modelului $= \frac{1}{2} * \ln\left(\frac{1-e}{e}\right)$ și cortăm modelul cu ponderea cea mai mare; actualizăm ponderile exemplelor astfel setate ca egale și normalizăm distribuția pentru a menține suma ponderilor 1; Rezultă: bias mic, varianța controlată, dar mai complex);

Categorii de metode:

- selecția modelelor: definim o ierarhie de subspații cu complexitate (capacitate) crescătoare $H_0 \subseteq H_1 \subseteq H_2 \subseteq \ldots \subseteq H_d \subseteq \ldots$, apoi, având un eșantion de învățare, căutăm subspațiul optim din care ne alegem ipoteza finală; metode: penalizare a complexității (SRM, MDLP), validare prin învățare multiplă / re-eșantionare (validare încrucișată, bootstrapping). Spațiul funcțiilor polinomiale (curbe) de grad i:

$$H_i = \{h(x, w) = w_0 + w_1 x + w_2 x^2 + \ldots + w_i x^i\};$$

$$\underset{\text{funcție constantă}}{H_0} \subseteq \underset{\text{dreaptă paralelă}}{H_1} \subseteq H_2 \subseteq \ldots$$

- tehnici de regularizare: este folosit similar un criteriu de penalizare asociat fiecărei funcții din H care măsoară fie complexitatea structurii de parametri, fie cât e de „netedă" sau derivabilă (minimul local al unei funcții e: derivata de

gradul 1 = 0 și, cea de gradul 2 > 0 - eficiența maximă matematic pentru h optim; punctul de inflexiune e cel care face derivata de gr. 2 = 0, iar în vecinătatea ei are semne diferite).
- medii: funcția de predicție se formează din mai multe funcții din H, care primesc ponderi diferite. Astfel, se „netezește" varianța excesivă a ipotezelor (metoda bayesiană, metoda bag) sau crește capacitatea de reprezentare a unui spațiu H neconvex (metodele boosting).

Metode de regularizare

$$\ell(u_i; h(x_i; w)) = \sum_{i=1}^{|S|} (u_i - h(x_i, w))^2 + \underbrace{\lambda \|w\|^2}_{\text{penalitate}}$$

$$\|w\|^2 = w_0^2 + w_1^2 + \ldots + w_i^2$$

λ - controlează importanța termenului de regularizare/penalitate

Alegerea modelului: împărțim datele inițiale în 2 mulțimi: mulțimea de antrenare și mulțimea de validare și alegem i și λ pe baza erorii pe mulțimea de validare.

Matricea de confuzie ($M(i,j)$) a unei reguli de clasificare h este o matrice de dimensiune C×C al cărei element generic indică numărul de exemple din setul de teste T, care aparțin clasei i și care au fost clasificate ca făcând parte din clasa j. Pentru clasificare binară:

	'+'	'-'
'+'	true positives	false positives
'-'	false negatives	true negatives

TPR (true positive rate / sensibilitate / recall) = TP/P
FPR (false positive rate / fall-out) = FP/N
P = TP + FN (toate pozitivele corecte)
N = FP + TN (— " — negativele corecte)

Dacă avem nevoie de rezultate binare și avem mai multe funcții h (calculăm suma ponderată a lor) sau o funcție ce returnează valori într-un interval, folosim 1 prag pentru a genera un rezultat binar. Într-o sarcină de detecție binară (obiect prezent versus obiect absent), o formulă uzuală de predicție e: $\delta(x) = I(f(x) > \tau)$, unde:
- $\delta(x)$ e regula de decizie, care = 1, dacă obiectul există și, 0 dacă nu există,
- I(condiție) este o funcție indicator, care = 1, dacă condiția e adevărată și 0 dacă nu e;
- f(x) este scorul de încredere - poate fi și probabilitatea estimată $P(y=1|x)$, dar, în practică, f(x) vine dintr-un model de genul: - Regresie logistică: $f(x) = \sigma(w^T x + b)$, unde σ = funcție sigmoidală;
  - SVM: $f(x) = w^T x + b$ (o valoare reală, care se interpretează ca „distanță" față de hiperplan);
  - Rețea neuronală: ultimul strat cu activare softmax sau sigmoidă dă un scor $f(x) \in [0,1]$.
- τ e pragul de sensibilitate a detecției. Implicit, multe modele „calibrează" f(x) ca probabilitate și iau $\tau = 0,5$.

Variația valorii pragului / estimări corecte = curba ROC (caracteristica de operare a receptorului):
ROC = (FPR(τ), TPR(τ)), unde τ = pragul.

Când vrem să găsim un eveniment rar (precum identificarea unui element într-o imagine), fiindcă toată detecția se întâmplă când pragul τ e foarte mare (adică sunt puține $f(x) > \tau$, majoritatea lucrurilor sunt false), atunci FPR devine f. mic, așa că se preferă să se folosească pe axa Ox: FP în loc de FPR (FP = „câte alarme false dă detectorul").

Def. Precizia $P = TP/(TP+FN) = p(y=1 | \hat{y}=1)$; cât din ce detectăm pozitiv e corect

Def. Recall $R = TP/(TP+FP) = p(\hat{y}=1 | y=1)$; cât % dintre pozitive am detectat

Precizie-curba recall este un grafic de puncte cu coordonate Recall și Precizie, cu valori calculate pentru fiecare prag. Punctele trebuie să tindă spre dreapta sus (1 și 1).

Scorul F1 = media armonică dintre P (precizie) și R (recall) pentru un anumit prag = $= 2\left(\frac{1}{P}+\frac{1}{R}\right)$. Acesta este util când este un număr f mic de rezultate pozitive, dar precizie e mare ș.d. recall e f mic. Media aritmetică pr f mare, dar media armonică dr f mai mică și, ci evidnț mai corect eficiența strategiei. Estimarea $R_{real}(h)$ rezultă în funcție de intervalul de încredere, dacă numărul de exemple de test este mare (>100):

$$R_{real}(h) = \left[\frac{terr}{t} \pm \zeta(x) \sqrt{\frac{\frac{terr}{t}\left(1-\frac{terr}{t}\right)}{t}} \right].$$

Funcția $\zeta(x)$ = zeta(x) are următoarele valori, unde x = numărul de cazuri corecte din toate testele:

x	50%	68%	80%	90%	95%	98%	99%
$\zeta(x)$	0,67	1,00	1,28	1,64	1,96	2,33	2,58

$\rightarrow SE = \frac{\bar{\sigma}}{\sqrt{N}} = \sqrt{\frac{\bar{\sigma}^2}{N}}$, unde $\bar{\sigma}^2$ este estimarea varianței of loss

$\bar{\sigma}^2 = \frac{1}{N}\sum_{t=1}^{N}(L_t - \bar{L})^2$; $L_t = L(y_t, f_m^{ek(t)}(x_t))$, $\bar{L} = \frac{1}{N}\sum_{t=1}^{N} L_t$

σ este o măsură a variabilității intrinseci a L_t across samples și nu are nici o legătură cu mărimea lui $\bar{L}$.

Regula o Eroare Standard: Un mod comun de a alege un model după calcularea riscului și a erorii standard, este alegerea valorii care corespunde celui mai simplu model al cărui risc nu este mai mare decât riscul celui mai bun model + eroarea standard.

Pt estimarea superioară a riscului se folosesc 3 seturi de date mutual exclusive: de antrenament, de testare și de validare. Metoda permite și identificarea convergenței dintre diverși algoritmi.

Comparare a 2 ipoteze obținute cu același algoritm: dacă avem 2 funcții din H care sunt obținute de același algoritm pentru seturi de date T_1 și T_2 cu $terr_1$ și $terr_2$ numărul de teste greșite din totalul t_1 și t_2, atunci riscul unei a diferite funcții față de cealaltă e:

$$\hat{\delta}_R(h_1, h_2) = \frac{terr.1}{t_1} - \frac{terr.2}{t_2}$$

Intervalul de confidență a valorii este dată de formula: $\left[\hat{\delta}_R \pm \zeta(x) \sqrt{\frac{\frac{terr.1}{t_1}\left(1-\frac{terr.1}{t_1}\right)}{t_1} + \frac{\frac{terr.2}{t_2}\left(1-\frac{terr.2}{t_2}\right)}{t_2}} \right]$

Comparare a 2 algoritmi (A_1 și A_2) aplicați pe seturi de testare diferite.

Folosim tehnici ce utilizează selectarea aleatorie a seturilor de antrenament și testare, precum validarea încrucișată. Exemplu de algoritm:

- împărțim setul de date în S (de antrenament) și T (de testare) în k părți egale: $T_1, T_2, \ldots, T_k$
- pentru fiecare k, antrenăm pe un subset S_i și testăm pe T_i fiecare algoritm obținând ratele de eroare ($terr/t$) pt fiecare R_1 și R_2. $\delta_i = R_1 - R_2$.
- calculăm valoarea medie a celor k valori δ ($sum\ \delta / k$)
- dacă fiecare pt k are cel puțin 30 de elem. în ea, putem calcula diferența dintre intervalele de încredere:

$$\left[\bar{\delta} \pm \zeta(x,k) \sqrt{\frac{1}{k(k-1)}\sum_{i=1}^{k}(\delta_i - \bar{\delta})^2} \right]$$

Comparare a 2 algoritmi pe același set de testare (testul pereche al lui McNemar sau Gillick)

Notăm: n_{00} = numărul de exemple clasificate greșit de h_1 și h_2

n_{01} = " " " " " " — doar de h_1, dar nu și de h_2

n_{10} = " " " " " " — h_2, dar nu și de h_1

n_{11} = " " " " " " corect de h_1 și de h_2

$$z = \frac{|n_{01} - n_{10}|}{\sqrt{(n_{01} + n_{10})}}$$

Modelul supervizat al învățării din date: - Ingredient: mediul (setul de date x), profesorul (etichetele u asociate x-urilor) și LM.

- Sarcina (LM caută în H cea mai bună funcție $h(x)$, care returnează cât mai des valorile u)

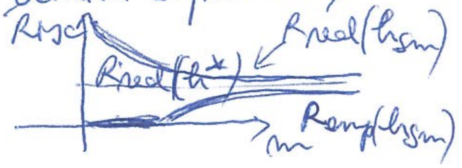
(curs) 3

— principiul inductiv (indică ce caută în unele alese, obiectivul ideal, ce influențează alegerea metodei de învățare)

Teoria diferenței inductoare. Principiul inductiv indică ce presupune să fie folosită pt a minimiza riscul real în funcție de setul de antrenament. Empirical Risk Minimization — ERM (pp. dacă scade riscul empiric — formula de mai sus, scade riscul real). Alegem folosind principiul de decizie (media) bayesian sau Alegem o singură ipoteză folosind principiul comprimării informației: eliminăm datele redundante pt a evidenția regularități din setul de antrenament. Pp că acestea sunt valabile și pt restul datelor.

Condiții de validitate pt principiul ERM: Diferența dintre riscul real pt ipoteza optimă $R_{real}(h^*)$ versus riscul empiric pt ipoteza găsită $R_{emp}(\hat{h}_S)$ din setul de antrenament. Considerăm ε o valoare maximă acceptată a diferenței dintre cele 2 riscuri. Diferența depinde de reprezentativitatea datelor de antrenament față de totalul datelor; se ve căuta funcția care este cea mai bună pt cele mai probabile situații, motiv pt care se folosesc setul de antrenament alese aleator.

Consistența principiului ERM: Principiul ERM este consistent dacă riscul real necunoscut $R_{real}(\hat{h}_S)$ și riscul empiric $R_{emp}(\hat{h}_S)$ converg către aceeași limită, $R_{real}(h^*)$, când dimensiunea m a setului de antrenare tinde la ∞. Riscul empiric este 0 la început datorită supraînvățării.



Generalizarea legii numerelor mari (Vapnik și Chervonenkis, 1971, 1989 - VC): dacă condițiile necesare și suficiente pt convergența uniformă nu sunt îndeplinite — adică dacă entropia VC raportată la numărul de observații nu tinde către 0 — atunci (∃) un subspațiu X^* al spațiului R^n cu măsura de probabilitate egală cu c, asemenea încât grupe oricare eșantion de dimensiune arbitrară k, $x_1^*, \ldots, x_k^* \in X^*$, poate fi separat în toate cele 2^k moduri posibile de către funcțiile din mulțimea admisibilă de funcții indicator $f(x, a), a \in A$.

Adică:

Fără convergența uniformă ($R_{real} - R_{emp} \to 0$, când $m \to \infty$), nici un algoritm determinist care alege o singură funcție nu poate generaliza în mod garantat. Excepție: dacă introduci informație a priori despre care funcții sunt "plauzibile" (de exemplu un prior bayesian) și apoi faci medierea / ponderare bayesiană, poți obține, totuși, generalizare. Consecința teoremei: fără nici o informație suplimentară (a priori), (∀) alegere unică de funcție ne "suprapretind" doar pe datele de antrenament și nu ne generaliza la date noi: poți găsi întotdeauna un eșantion advers în care acel model se potrivește perfect (eroare 0 pe antrenament), dar e complet eronat pe alte exemple. Entropia VC cuantifică complexitatea clasei de ipoteze; pt a obț. convergența uniformă, raportul entropiei VC la nr. de exemple trebuie să tindă către 0.

Protocolul învățării: stabilește interacțiunea dintre LM și mediu, inclusiv tipul de date de input, tipul de răspunsuri posibile și supervizorul. Învățarea batch — toate datele sunt furnizate la început, învățarea on-line — datele sunt furnizate în secvențe, iar LM trebuie să furnizeze un răspuns după fiecare (sarcini de identificare, a explicației și predicție). Metodele boosting — când LM poate influența probabilitatea distribuției de date din care învață. Învățarea prin întrebări de categorizare — LM întreabă despre clasele din care apare ce a observat.

Învățarea activă — LM organizează experimente asupra lumii și observă efectele

Problema legării riscului empiric de cel real

a) enunț: modelul de minimizare a riscului: $R(a) = \int Q(y, f(x, a))\, dP(y, x), a \in A$, unde probabilitatea nu este cunoscută, dar perechile $(y_1, x_1) \ldots$ sunt date.

b. soluții:

b1. proceduri bazate pe gradient: $a_n = a_{n-1} + \gamma_n \nabla_\alpha Q(y_n, f(x_n, a_{n-1}))$, unde a_n = vectorul de greutăți la pasul n, γ_n = rata de învățare, ∇_α = gradientul (în ce direcție și cu ce intensitate trebuie modificați parametrii), $Q(y_n, f(x_n, \alpha_{n-1}))$ - o funcție care măsoară eroarea (sau pierderea) între valoarea estimată y_n și predicția modelului $f(x_n, \alpha_{n-1})$.

b.2 ERM cu condiții convergență uniformă;

c. Demonstrații;

c1. teoria aproximării stochastice - pt. procedurile bazate pe gradient

c2 teoria convergenței uniforme pt. principiul ERM.